

**PENGEMBANGAN SISTEM PELAPORAN DAN KEARSIPAN
FENOMENA PADA LINGKUNGAN PRODUKSI
BERBASIS *LEAN MANUFACTURING***

TUGAS SARJANA

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

Oleh

Gian Gervase Geoffrey

13116020



**PROGRAM STUDI TEKNIK MESIN
FAKULTAS TEKNIK MESIN DAN DIRGANTARA
INSTITUT TEKNOLOGI BANDUNG**

2020

“

The biggest opportunity for big companies has come by far
in the digitization of internal process

“

Jack Welsh - Business Executive & Writer
Chairman & CEO of General Electrics (1981-2001)

“

Digitalization has created opportunities for everybody
to accumulate information in a way they were never able to,
and analyze it with a speed that just wasn't there.

“

Ken Moelis - Investment Banker
Founder of Moelis & Company

Lembaran Pengesahan

Tugas Sarjana

Pengembangan Sistem Pelaporan dan Kearsipan
Fenomena pada Lingkungan Produksi
Berbasis *Lean Manufacturing*

Oleh

Gian Gervase Geoffrey

13116020

Program Studi Teknik Mesin
Fakultas Teknik Mesin Dan Dirgantara
Institut Teknologi Bandung

Disetujui pada Tanggal: 23 Desember 2020

Pembimbing Utama



Prof. Dr.Ir. Yatna Yuwana Martawirya

NIP 195603231981031003

Tugas Sarjana

Judul	Pengembangan Sistem Pelaporan dan Kearsipan Fenomena pada Lingkungan Produksi Berdasarkan <i>Lean Manufacturing</i>	Gian Gervase Geoffrey
Program Studi	Teknik Mesin	13116020
Fakultas Teknik Mesin dan Dirgantara Institut Teknologi Bandung		
<p style="text-align: center;">Abstrak</p> <p>Persaingan pada industri manufaktur menjadi semakin kompetitif, sehingga perusahaan manufaktur harus menekan harga jual. Adapun untuk mempertahankan margin keuntungan, biaya produksi juga harus ditekan. Salah satu cara menekan biaya produksi adalah dengan mengurangi pemborosan (<i>muda</i>) dan melakukan perbaikan berkelanjutan (<i>kaizen</i>).</p> <p>Maka dari itu dikembangkan aplikasi web sistem pelaporan yang dapat menangani pemborosan dan kesempatan <i>kaizen</i>. Fenomena berkaitan dengan pemborosan dan kesempatan <i>kaizen</i> dapat diarsipkan pada sistem untuk ditangani. Pada saat proses pelaporan, suatu laporan akan diberi <i>tag</i> mengenai suatu topik agar mudah dikelompokkan berdasarkan topiknya. Kemudian laporan tersebut akan disampaikan kepada pihak-pihak yang sesuai untuk diselesaikan.</p> <p>Dari sisi pihak yang akan menangani laporan, pihak tersebut dapat mengatur laporan apa yang akan diterima berdasarkan <i>tag</i> yang ia langgan, serupa dengan media sosial. Fitur mengenai manajemen tugas juga memastikan laporan yang sudah diambil untuk ditangani tidak dapat diambil lagi sehingga tidak ada redundansi dalam penyelesaian laporan. Terdapat juga papan kanban untuk memudahkan pihak tersebut dalam manajemen tugas.</p> <p>Sistem kearsipan dan manajemen tugas diatas memang baik untuk perusahaan, namun untuk memastikan urgensi pekerja untuk menggunakan sistem tersebut, ditambahkan sistem gamifikasi untuk memberi insentif kepada pihak yang berpartisipasi. Dihubungkan juga aplikasi web ke jaringan WLAN agar mudah diakses oleh perangkat selain hos.</p> <p>Kata kunci: <i>Lean manufacturing</i>, <i>kaizen</i>, kearsipan, manajemen tugas, gamifikasi, aplikasi web</p>		

Final Project

Title	Development of Phenomena Reporting and Record Management System on Lean Manufacturing Based Shop Floor	Gian Gervase Geoffrey
Major	Mechanical Engineering	13116020
Faculty of Mechanical and Aerospace Engineering Bandung Institute of Technology		
<p style="text-align: center;">Abstract</p> <p>The manufacturing industry has gotten more competitive than ever, so the manufacturer should press their product selling price. Therefore, to maintain its profit margins, production costs should have been pressed too. One way to overcome this problem is by reducing wastes (muda) and making continuous improvements (kaizen).</p> <p>Hence, the author has developed a web application that accommodates the reporting of wastes and kaizen opportunities. The phenomenon regarding waste and kaizen opportunity could be recorded for further resolution. During the reporting process, several tags will be given to a report, so that report can be grouped easily based on its topics. Then those reports will be delivered to the relevant parties to be resolved.</p> <p>From the perspective of the party that resolves reports, those parties could customize kinds of report that they receive based on those tags that they subscribed to, like on social media. The task management feature also ensures those reports that have been taken, could not be taken again. So, there are no redundant works. There is also a kanban board to help those parties to manage those reports they have taken.</p> <p>Lastly, to raise the urgency of participation in this system through web application among workers, a gamification scheme has been added to incentivized those participating parties. The web application is also connected to a WLAN, so the web application could be accessed easily.</p> <p>Key words: Lean manufacturing, kaizen, records management, task management, gamification, web application</p>		

Kata Pengantar

Segala puji dan syukur kepada Tuha Yang Maha Esa karena atas kehendak dan berkat dari-Nya, penulis dapat menyelesaikan tugas sarjana dengan judul “Pengembangan Sistem Pelaporan dan Kearsipan Fenomena pada Lingkungan Produksi Berbasis *Lean Manufacturing*”. Dengan adanya tugas sarjana ini diharapkan dapat berguna dalam bidang manajemen produksi serta menjadi langkah awal dalam proses mendigitalisasi proses produksi.

Adapun penulis seorang diri tidak dapat menyelesaikan tugas akhir ini. terselesaikannya tugas akhir ini tidak terlepas dari pihak-pihak terkait. Maka dari itu penulis ingin berterimakasih kepada:

1. Orang tua dan keluarga penulis yang memenuhi segala kebutuhan baik kebutuhan materiil maupun kebutuhan non-materiil dari penulis sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Prof. Dr.Ir. Yatna Yuwana Martawirya selaku dosen pembimbing yang selalu memberi arahan, bimbingan, dan memastikan kualitas tugas akhir ini.
3. Dr. Sri Raharno ST., MT. dan Kak Fahmi yang membantu dalam memberi masukan dan bimbingan dalam menyusun proposal tugas sarjana.
4. Steven, Christian, Nicho, Andy, Suparjo, Marcello, dan Davin selaku rekan seperlombaan dan rekan sekosan yang memberi masukan dan mendukung penulis dalam penyelesaian tugas akhir.
5. Teman-teman Teknik Mesin 2016, terutama yang bertugas akhir di Laboratorium Teknik Produksi yang menjadi rekan berdiskusi dan bertukar pemikiran.
6. *Content creator* dan *developer* yang menunggah materi video pembelajaran dan solusi pemrograman mengenai Django dan Ajax di Youtube dan Stack Overflow yang sangat membantu penulis dalam mengembangkan aplikasi web.

Adapun penulis juga ingin berterima kasih kepada para pembaca yang telah memilih tugas akhir ini sebagai salah satu referensi. Selain itu mohon maaf terdapat banyak keterbatasan dalam tugas akhir ini. Bila ada kritik dan masukan, pembaca dapat mengirim surel kepada penulis ke alamat surel giangervase@gmail.com. Penulis sangat menanti masukan tersebut. Semoga tugas akhir ini dapat bermanfaat bagi kita semua.

Tangerang Selatan, 17 Desember 2020

Gian Gervase Geoffrey

Daftar Isi

Kata Pengantar	i
Daftar Isi	ii
Daftar Gambar	vi
Daftar Tabel	xv
Bab 1 Pendahuluan.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan.....	2
1.4. Batasan Masalah.....	3
1.5. Manfaat.....	3
1.6. Sistematika Penulisan.....	3
Bab 2 Tinjauan Pustaka	5
2.1. <i>Lean Manufacturing System</i>	5
2.1.1. <i>MUDA</i>	5
2.1.2. <i>Jishu Hozen (Autonomous Maintenance)</i>	6
2.1.3. <i>Kaizen (Continuous Improvement)</i>	6
2.2. Arsip dan Kearsipan	9
2.3. Gamifikasi.....	11
2.4. Python.....	13
2.4.1. Fungsi	14
2.4.2. Pemrograman Berbasis Objek.....	14
2.4.3. Kelas	14
2.5. Aplikasi Web	15
2.6. Django	16

2.6.1. Lapisan Abstraksi	18
2.6.2. Lapisan Penyajian.....	19
2.6.3. Lapisan Logika Bisnis	19
2.7. Manajemen Tugas	20
2.7.1. Siklus Hidup Tugas	21
2.7.2. Metode Kanban	25
2.8. WLAN	28
2.9. ERD	28
Bab 3 Pengembangan Perangkat Lunak	30
3.1. Metodologi.....	30
3.2. Analisis Kebutuhan	31
3.3. Perancangan Sistem.....	33
3.3.1. Perancangan Aktivitas	34
3.3.2. Perancangan Model	52
3.3.3. Basis Data Hasil Perancangan.....	64
3.4. Pengembangan Perangkat Lunak	67
3.4.1. Pengembangan Laman <i>'login'</i>	68
3.4.2. Pengembangan Laman <i>'logout'</i>	68
3.4.3. Pengembangan Laman <i>'password_reset'</i>	69
3.4.4. Pengembangan Laman <i>'register'</i>	70
3.4.5. Pengembangan Laman <i>'profile'</i>	71
3.4.6. Pengembangan Laman <i>'tag-list'</i>	72
3.4.7. Pengembangan Laman <i>'report-create'</i>	74
3.4.8. Pengembangan Laman <i>'all-reports'</i>	75
3.4.9. Pengembangan Laman <i>'report-detail'</i>	77
3.4.10. Pengembangan Laman <i>'progress-taken'</i>	83

3.4.11. Pengembangan Laman ' <i>unit-detail</i> '	86
3.4.12. Pengembangan Laman ' <i>auth-detail</i> '	89
3.4.13. Pengembangan Laman ' <i>user-reports</i> '	92
3.5. Perancangan dan pengaturan jaringan	94
Bab 4 Pengujian dan Analisis	96
4.1. Skema Pengujian	96
4.2. Pengujian Fitur pada Aplikasi Web	96
4.3. Pengujian Kinerja Aplikasi Web saat Terhubung dengan Jaringan WLAN	97
Bab 5 Kesimpulan dan Saran	99
5.1. Kesimpulan	99
5.2. Saran	99
Daftar Pustaka	101
Lampiran A Pengembangan Perangkat Lunak (Lanjutan)	105
A.1. Pengembangan Laman ' <i>password_reset_done</i> '	105
A.2. Pengembangan Laman ' <i>password_reset_confirm</i> '	105
A.3. Pengembangan Laman ' <i>password_reset_complete</i> '	106
A.4. Pengembangan Laman ' <i>point-list</i> '	107
A.5. Pengembangan Laman ' <i>user-list</i> '	108
A.6. Pengembangan Laman ' <i>tag-create</i> '	110
A.7. Pengembangan Laman ' <i>unit-create</i> '	111
A.8. Pengembangan Laman ' <i>point-create</i> '	112
A.9. Pengembangan Laman ' <i>tag-reports</i> '	114
A.10. Pengembangan Laman ' <i>home</i> '	116
A.11. Pengembangan Laman ' <i>report-update</i> '	117
A.12. Pengembangan Laman ' <i>report-delete</i> '	119
A.13. Pengembangan Laman ' <i>progress-subscribed</i> '	120

A.14. Pengembangan Laman ‘ <i>unit-list</i> ’	122
A.15. Pengembangan Laman ‘ <i>user-taken</i> ’	123
A.16. Pengembangan Laman ‘ <i>user-collab</i> ’	125
A.17. Pengembangan Laman ‘ <i>user-career</i> ’	127
A.18. Pengembangan Laman ‘ <i>user-point</i> ’	128
A.19. Pengembangan Laman ‘ <i>about</i> ’	130
A.20. Pengembangan Laman ‘ <i>dashboard</i> ’	130
A.21. Pengembangan <i>Template</i> Dasar	131
Lampiran B Pengujian Fitur pada Aplikasi Web.....	136
B.1. Pengujian Fitur 1 “Membuat Laporan” (<i>Create a Report</i>)	136
B.2. Pengujian Fitur 2 “Melihat Laporan” (<i>View Reports</i>)	146
B.3. Pengujian Fitur 3 “Membuat <i>Tag</i> ” (<i>Create a Tag</i>)	156
B.4. Pengujian Fitur 4 “Mengambil Laporan” (<i>Take a Report</i>)	158
B.5. Pengujian Fitur 5 “Melihat Aktivitas Pengguna” (<i>View User Activities</i>).....	167
B.6. Pengujian Fitur 6 “Mengelola Organisasi” (<i>Manage Organization</i>)	171
B.7. Pengujian Fitur 7 “Mengelola Otorisasi” (<i>Manage Authorization</i>)	178
B.8. Pengujian Fitur 8 “Mengelola Point” (<i>Manage Point</i>).....	181
B.9. Pengujian Fitur 9 “Berkolaborasi pada Laporan” (<i>Collaborate on a Report</i>)..	185
B.10. Pengujian Fitur “Akun Pengguna”	188
B.11. Pengujian Fitur “Tambahan”	194
Lampiran C Pengujian Kinerja Aplikasi Web pada Jaringan WLAN	197
C.1. Pengujian Kinerja Aplikasi Web pada Ponsel Pintar	197
C.2. Pengujian Kinerja Aplikasi Web pada Komputer (Selain Hos).....	201

Daftar Gambar

Gambar 2.1 Pembagian peran departemen operasi dan departemen perawatan [2].....	7
Gambar 2.2 Contoh lembar prosedur kerja dan standar sementara <i>Jishu Hozen</i> [2].....	8
Gambar 2.3 Contoh <i>tag</i> pada pelaksanaan <i>Jishu Hozen</i> [5].....	8
Gambar 2.4 Skema model <i>Holistic Information Assets Management (HIAM)</i> [7]	11
Gambar 2.5 Kerangka kerja MDE pada skema gamifikasi [8]	13
Gambar 2.6 Konsep pola perancangan MTV oleh Django [19].....	17
Gambar 2.7 Teori siklus hidup tugas (<i>task life cycle</i>) menurut IBM [30]	22
Gambar 2.8 Teori siklus hidup tugas (<i>task life cycle</i>) menurut BMC [31]	23
Gambar 2.9 Teori siklus hidup tugas (<i>task life cycle</i>) menurut Camunda [32].....	25
Gambar 2.10 Contoh papan kanban pada Microsoft Azure [35].....	27
Gambar 2.11 Hubungan antar-entitas pada basis data	29
Gambar 2.12 Notasi Kaki Gagak.....	29
Gambar 3.1 Diagram alir metodologi penelitian tugas akhir	30
Gambar 3.2 Diagram alir <i>predefined processes</i> pada metodologi tugas akhir.....	31
Gambar 3.3 Alur kerja utama pada sistem	35
Gambar 3.4 Aktivitas pada fitur 1, “Membuat Laporan” (“ <i>Create a Report</i> ”).....	37
Gambar 3.5 Aktivitas pada fitur 2, “Melihat Laporan” (“ <i>View Reports</i> ”).....	39
Gambar 3.6 Aktivitas pada fitur 3, “Membuat <i>Tag</i> ” (“ <i>Create a Tag</i> ”)	40
Gambar 3.7 Aktivitas pada fitur 4, “Mengambil Laporan” (“ <i>Take a Report</i> ”).....	42
Gambar 3.8 Aktivitas pada fitur 5, “Melihat Aktivitas Pengguna” (“ <i>View User Activities</i> ”).	44
Gambar 3.9 Aktivitas pada fitur 6, “Mengelola Organisasi” (“ <i>Manage Organization</i> ”)	45
Gambar 3.10 Aktivitas pada fitur 7, “Mengelola Otorisasi” (“ <i>Manage Authorization</i> ”)	46
Gambar 3.11 Aktivitas pada fitur 8, “Mengelola Point” (“ <i>Manage Point</i> ”).....	47
Gambar 3.12 Aktivitas pada fitur 9, “Berkolaborasi pada Laporan” (“ <i>Collaborate on a Report</i> ”)	49
Gambar 3.13 Aktivitas pada fitur “Akun Pengguna”	50
Gambar 3.14 Aktivitas pada fitur “Tambahan”	51
Gambar 3.15 Penyederhanaan penggambaran hubungan banyak-dengan-banyak.....	65
Gambar 3.16 ERD kluster sistem pelaporan	65
Gambar 3.17 ERD kluster sistem keorganisasian.....	66

Gambar 3.18 ERD kluster sistem poin.....	66
Gambar 3.19 Cuplikan layar laman ' <i>login</i> ' dengan petunjuk	68
Gambar 3.20 Cuplikan layar laman ' <i>logout</i> ' dengan petunjuk	69
Gambar 3.21 Cuplikan layar laman ' <i>password_reset</i> ' dengan petunjuk.....	69
Gambar 3.22 Cuplikan layar laman ' <i>register</i> ' dengan petunjuk	70
Gambar 3.23 Cuplikan layar laman ' <i>profile</i> ' dengan petunjuk.....	72
Gambar 3.24 Cuplikan layar laman ' <i>tag-list</i> ' dengan petunjuk	73
Gambar 3.25 Cuplikan layar laman ' <i>report-create</i> ' dengan petunjuk.....	74
Gambar 3.26 Cuplikan layar laman ' <i>all-reports</i> ' dengan petunjuk.....	76
Gambar 3.27 Cuplikan layar laman ' <i>report-detail</i> ' pada kasus 1 dengan petunjuk.....	78
Gambar 3.28 Cuplikan layar laman ' <i>report-detail</i> ' pada kasus 2 dengan petunjuk.....	79
Gambar 3.29 Cuplikan layar laman ' <i>report-detail</i> ' pada kasus 3 dengan petunjuk.....	80
Gambar 3.30 Cuplikan layar laman ' <i>report-detail</i> ' pada kasus 4 dengan petunjuk	81
Gambar 3.31 Cuplikan layar laman ' <i>report-detail</i> ' pada kasus 5 dengan petunjuk.....	82
Gambar 3.32 Cuplikan layar laman ' <i>progress-taken</i> ' dengan petunjuk	84
Gambar 3.33 Cuplikan layar laman ' <i>unit-detail</i> ' dengan petunjuk	88
Gambar 3.34 Cuplikan layar laman ' <i>auth-detail</i> ' dengan petunjuk	91
Gambar 3.35 Cuplikan layar laman ' <i>user-reports</i> ' dengan petunjuk	93
Gambar 3.36 Cuplikan layar bagian WLAN <i>adapter</i> Wi-Fi pada <i>terminal</i>	95
Gambar B.1 Cuplikan layar laman ' <i>tag-list</i> ' pada pengujian pada peramban web pertama..	136
Gambar B.2 Cuplikan layar laman ' <i>tag-list</i> ' pada pengujian pada peramban web kedua.....	137
Gambar B.3 Cuplikan layar laman ' <i>tag-list</i> ' pada pengujian pada peramban web ketiga.....	137
Gambar B.4 Cuplikan layar laman ' <i>report-create</i> ' pada pengujian pada peramban web pertama	138
Gambar B.5 Cuplikan layar laman ' <i>report-create</i> ' pada pengujian pada peramban web kedua	139
.Gambar B.6 Cuplikan layar laman ' <i>report-create</i> ' pada pengujian pada peramban web ketiga	139
Gambar B.7 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian pada peramban web pertama	140

Gambar B.8 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian pada peramban web kedua	141
Gambar B.9 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian pada peramban web ketiga	141
Gambar B.10 Cuplikan layar laman ' <i>report-update</i> ' pada pengujian pada peramban web pertama.....	142
Gambar B.11 Cuplikan layar basis data untuk model ' <i>Report</i> ' setelah pengujian laman ' <i>report-update</i> '	143
Gambar B.12 Cuplikan layar laman ' <i>report-update</i> ' pada pengujian pada peramban web kedua	143
Gambar B.13 Cuplikan layar laman ' <i>report-update</i> ' pada pengujian pada peramban web ketiga	144
Gambar B.14 Cuplikan layar laman ' <i>report-delete</i> ' pada pengujian pada peramban web kedua	145
Gambar B.15 Cuplikan layar laman ' <i>report-delete</i> ' pada pengujian pada peramban web ketiga	145
Gambar B.16 Cuplikan layar laman ' <i>report-delete</i> ' pada pengujian pada peramban web pertama	146
Gambar B.17 Cuplikan layar basis data untuk model ' <i>Report</i> ' setelah pengujian laman ' <i>report-delete</i> '	146
Gambar B.18 Pesan galat 404 untuk objek ' <i>Report</i> ' yang tidak ada.....	146
Gambar B.19 Cuplikan layar basis data untuk hubungan model ' <i>Profile</i> ' dengan model ' <i>Tag</i> '	147
Gambar B.20 Cuplikan layar basis data untuk model ' <i>Report</i> '.....	147
Gambar B.21 Cuplikan layar laman ' <i>home</i> ' pada pengujian pada peramban web pertama ..	148
Gambar B.22 Cuplikan layar laman ' <i>home</i> ' pada pengujian pada peramban web kedua	149
Gambar B.23 Cuplikan layar laman ' <i>home</i> ' pada pengujian pada peramban web ketiga	149
Gambar B.24 Cuplikan layar laman ' <i>tag-reports</i> ' pada pengujian pada peramban web pertama	150
Gambar B.25 Cuplikan layar basis data untuk hubungan model ' <i>Profile</i> ' dengan model ' <i>Tag</i> ' setelah pengujian laman ' <i>tag-reports</i> '	150

Gambar B.26 Cuplikan layar laman ' <i>tag-reports</i> ' pada pengujian pada peramban web kedua	151
Gambar B.27 Cuplikan layar laman ' <i>tag-reports</i> ' pada pengujian pada peramban web ketiga	151
Gambar B.28 Cuplikan layar laman ' <i>user-reports</i> ' pada pengujian pada peramban web pertama	152
Gambar B.29 Cuplikan layar laman ' <i>user-reports</i> ' pada pengujian pada peramban web kedua	153
Gambar B.30 Cuplikan layar laman ' <i>user-reports</i> ' pada pengujian pada peramban web ketiga	153
Gambar B.31 Cuplikan layar laman ' <i>user-collab</i> ' pada pengujian pada peramban web pertama	154
Gambar B.32 Cuplikan layar laman ' <i>user-collab</i> ' pada pengujian pada peramban web kedua	155
Gambar B.33 Cuplikan layar laman ' <i>user-collab</i> ' pada pengujian pada peramban web ketiga	155
Gambar B.34 Cuplikan layar laman ' <i>tag-create</i> ' pada pengujian pada peramban web pertama	156
Gambar B.35 Cuplikan layar basis data untuk model ' <i>Tag</i> ' setelah pengujian laman ' <i>tag-create</i> '	157
Gambar B.36 Cuplikan layar laman ' <i>tag-create</i> ' pada pengujian pada peramban web kedua	157
Gambar B.37 Cuplikan layar laman ' <i>tag-create</i> ' pada pengujian pada peramban web ketiga	157
Gambar B.38 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian kedua pada peramban web pertama	159
Gambar B.39 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian kedua pada peramban web kedua	159
Gambar B.40 Cuplikan layar basis data untuk model ' <i>Report</i> ' setelah pengujian ' <i>report-detail</i> ' kedua	160
Gambar B.41 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian ketiga pada peramban web pertama	160

Gambar B.42 Cuplikan layar basis data untuk model ' <i>Report</i> ' setelah pengujian ' <i>report-detail</i> ' ketiga.....	161
Gambar B.43 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian ketiga pada peramban web kedua	161
Gambar B.44 Cuplikan layar laman ' <i>progress-taken</i> ' pada pengujian pada peramban web pertama.....	162
Gambar B.45 Cuplikan layar laman ' <i>progress-taken</i> ' pada pengujian pada peramban web pertama (2)	162
Gambar B.46 Cuplikan layar laman ' <i>progress-taken</i> ' pada pengujian pada peramban web kedua	163
Gambar B.47 Cuplikan layar laman ' <i>progress-taken</i> ' pada pengujian pada peramban web ketiga	163
Gambar B.48 Cuplikan layar laman ' <i>progress-subscribed</i> ' pada pengujian pada peramban web pertama.....	164
Gambar B.49 Cuplikan layar laman ' <i>progress-subscribed</i> ' pada pengujian pada peramban web kedua	164
Gambar B.50 Cuplikan layar laman ' <i>progress-subscribed</i> ' pada pengujian pada peramban web ketiga.....	165
Gambar B.51 Cuplikan layar laman ' <i>user-taken</i> ' pada pengujian pada peramban web pertama	165
Gambar B.52 Cuplikan layar laman ' <i>user-taken</i> ' pada pengujian pada peramban web kedua	166
Gambar B.53 Cuplikan layar laman ' <i>user-taken</i> ' pada pengujian pada peramban web ketiga	167
Gambar B.54 Cuplikan layar laman ' <i>user-career</i> ' pada pengujian pada peramban web pertama	168
Gambar B.55 Cuplikan layar laman ' <i>user-career</i> ' pada pengujian pada peramban web kedua	168
Gambar B.56 Cuplikan layar laman ' <i>user-career</i> ' pada pengujian pada peramban web ketiga	169
Gambar B.57 Cuplikan layar laman ' <i>user-point</i> ' pada pengujian pada peramban web pertama	170

Gambar B.58 Cuplikan layar laman ' <i>user-point</i> ' pada pengujian pada peramban web kedua	170
Gambar B.59 Cuplikan layar laman ' <i>user-point</i> ' pada pengujian pada peramban web ketiga	171
Gambar B.60 Cuplikan layar laman ' <i>unit-create</i> ' pada pengujian pada peramban web pertama	172
Gambar B.61 Cuplikan layar basis data untuk model ' <i>Unit</i> ' setelah pengujian laman ' <i>unit-create</i> '	172
Gambar B.62 Cuplikan layar laman ' <i>unit-create</i> ' pada pengujian pada peramban web kedua	172
Gambar B.63 Cuplikan layar laman ' <i>unit-create</i> ' pada pengujian pada peramban web ketiga	173
Gambar B.64 Cuplikan layar laman ' <i>unit-detail</i> ' pada pengujian pada peramban web pertama	173
Gambar B.65 Cuplikan layar basis data untuk model ' <i>Unit</i> ' setelah pengujian laman ' <i>unit-detail</i> '	174
Gambar B.66 Cuplikan layar basis data untuk model ' <i>CareerHistory</i> ' setelah pengujian laman ' <i>unit-detail</i> '	174
Gambar B.67 Cuplikan layar laman ' <i>unit-detail</i> ' pada pengujian pada peramban web kedua	174
Gambar B.68 Cuplikan layar laman ' <i>unit-detail</i> ' pada pengujian pada peramban web ketiga	175
Gambar B.69 Cuplikan layar laman ' <i>unit-list</i> ' pada pengujian pada peramban web pertama	175
Gambar B.70 Cuplikan layar laman ' <i>unit-list</i> ' pada pengujian pada peramban web kedua..	176
Gambar B.71 Cuplikan layar laman ' <i>unit-list</i> ' pada pengujian pada peramban web ketiga..	176
Gambar B.72 Cuplikan layar laman ' <i>user-list</i> ' pada pengujian pada peramban web pertama	177
Gambar B.73 Cuplikan layar laman ' <i>user-list</i> ' pada pengujian pada peramban web kedua .	177
Gambar B.74 Cuplikan layar laman ' <i>user-list</i> ' pada pengujian pada peramban web ketiga .	178
Gambar B.75 Cuplikan layar laman ' <i>auth-detail</i> ' pada pengujian pada peramban web pertama	179

Gambar B.76 Cuplikan layar basis data untuk model ' <i>Auth</i> ' setelah pengujian laman ' <i>auth-detail</i> '	180
Gambar B.77 Cuplikan layar laman ' <i>auth-detail</i> ' pada pengujian pada peramban web kedua	180
Gambar B.78 Cuplikan layar laman ' <i>auth-detail</i> ' pada pengujian pada peramban web ketiga	181
Gambar B.79 Cuplikan layar laman ' <i>point-create</i> ' pada pengujian pada peramban web pertama	182
Gambar B.80 Cuplikan layar basis data untuk model ' <i>PointHistory</i> ' setelah pengujian laman ' <i>point-create</i> '	182
Gambar B.81 Cuplikan layar laman ' <i>point-create</i> ' pada pengujian pada peramban web kedua	182
Gambar B.82 Cuplikan layar laman ' <i>point-create</i> ' pada pengujian pada peramban web ketiga	183
Gambar B.83 Cuplikan layar laman ' <i>point-list</i> ' pada pengujian pada peramban web pertama	183
Gambar B.84 Cuplikan layar laman ' <i>point-list</i> ' pada pengujian pada peramban web kedua	184
Gambar B.85 Cuplikan layar laman ' <i>point-list</i> ' pada pengujian pada peramban web ketiga	184
Gambar B.86 Cuplikan layar basis data untuk model ' <i>Collaboration</i> ' setelah pengujian laman ' <i>report-detail</i> ' keempat	185
Gambar B.87 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian keempat pada peramban web pertama.....	186
Gambar B.88 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian keempat pada peramban web kedua.....	187
Gambar B.89 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian kelima pada peramban web pertama.....	188
Gambar B.90 Cuplikan layar basis data untuk model ' <i>Collaboration</i> ' setelah pengujian laman ' <i>report-detail</i> ' kelima.....	188
Gambar B.91 Cuplikan layar laman ' <i>profile</i> ' pada pengujian pada pengguna yang telah masuk	189
Gambar B.92 Cuplikan layar basis data untuk hubungan model ' <i>Profile</i> ' dengan model ' <i>Tag</i> ' setelah pengujian laman ' <i>profile</i> '	189

Gambar B.93 Cuplikan layar laman ' <i>profile</i> ' pada pengujian pada pengguna yang belum masuk	190
Gambar B.94 Cuplikan layar laman ' <i>logout</i> ' pada pengujian fitur “Akun Pengguna”	190
Gambar B.95 Cuplikan layar basis data untuk model ' <i>User</i> ' setelah pengujian fitur “Akun Pengguna”	190
Gambar B.96 Cuplikan layar basis data untuk model ' <i>Profile</i> ' setelah pengujian fitur “Akun Pengguna”	191
Gambar B.97 Cuplikan layar laman ' <i>register</i> ' pada pengujian fitur “Akun Pengguna”	191
Gambar B.98 Cuplikan layar laman ' <i>password_reset</i> ' pada pengujian fitur “Akun Pengguna”	192
Gambar B.99 Cuplikan layar laman ' <i>password_reset_done</i> ' pada pengujian fitur “Akun Pengguna”	192
Gambar B.100 Cuplikan layar laman surel konfirmasi yang dikirimkan pada pengujian fitur “Akun Pengguna”	192
Gambar B.101 Cuplikan layar laman ' <i>password_reset_confirm</i> ' pada pengujian fitur “Akun Pengguna”	193
Gambar B.102 Cuplikan layar laman ' <i>password_reset_complete</i> ' pada pengujian fitur “Akun Pengguna”	193
Gambar B.103 Cuplikan layar basis data untuk model ' <i>User</i> ' setelah pengujian fitur “Akun Pengguna”	194
Gambar B.104 Cuplikan layar laman ' <i>login</i> ' pada pengujian fitur “Akun Pengguna”	194
Gambar B.105 Cuplikan layar laman ' <i>about</i> ' pada pengujian pada pengguna yang telah masuk	195
Gambar B.106 Cuplikan layar laman ' <i>about</i> ' pada pengujian pada pengguna yang belum masuk	195
Gambar B.107 Cuplikan layar laman ' <i>dashboard</i> ' pada pengujian pada pengguna yang telah masuk	195
Gambar B.108 Cuplikan layar laman ' <i>dashboard</i> ' pada pengujian pada pengguna yang belum masuk	196
Gambar C.1 Cuplikan layar laman ' <i>login</i> ' pada pengujian masuk menggunakan ponsel	197
Gambar C.2 Cuplikan layar laman ' <i>home</i> ' setelah pengujian masuk menggunakan ponsel	197

Gambar C.3 Cuplikan layar laman ' <i>report-create</i> ' pada pengujian pembuatan laporan menggunakan ponsel.....	198
Gambar C.4 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian pengambilan laporan menggunakan ponsel.....	199
Gambar C.5 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian pengubahan status laporan menggunakan ponsel.....	199
Gambar C.6 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian pembuatan kolaborasi menggunakan ponsel.....	200
Gambar C.7 Cuplikan layar laman ' <i>report-detail</i> ' setelah pengujian pembuatan kolaborasi menggunakan komputer lain	200
Gambar C.8 Cuplikan layar laman ' <i>logout</i> ' setelah pengujian keluar menggunakan ponsel	203
Gambar C.9 Cuplikan layar laman ' <i>login</i> ' pada pengujian masuk menggunakan komputer lain	201
Gambar C.10 Cuplikan layar laman ' <i>home</i> ' setelah pengujian masuk menggunakan komputer lain.....	202
Gambar C.11 Cuplikan layar laman ' <i>report-create</i> ' pada pengujian pembuatan laporan menggunakan komputer lain	202
Gambar C.12 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian pengambilan laporan menggunakan komputer lain	203
Gambar C.13 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian pengubahan status laporan menggunakan komputer lain	203
Gambar C.14 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian pembuatan kolaborasi menggunakan komputer lain	204
Gambar C.15 Cuplikan layar laman ' <i>report-detail</i> ' pada pengujian pembuatan kolaborasi menggunakan komputer lain	205
Gambar C.16 Cuplikan layar laman ' <i>logout</i> ' setelah pengujian keluar menggunakan komputer lain.....	205

Daftar Tabel

Tabel 3.1 Bidang yang digunakan pada <i>model "User"</i>	53
Tabel 3.2 Bidang pada <i>model "Tag"</i>	53
Tabel 3.3 Bidang pada <i>model "Report"</i>	54
Tabel 3.4 Bidang pada <i>model "Report"</i> (lanjutan).....	55
Tabel 3.5 Pilihan pada kelas Level(model.IntegerChoices)	56
Tabel 3.6 Pilihan pada kelas TaskLevel(model.IntegerChoices)	56
Tabel 3.7 Bidang pada <i>model "Collaboration"</i>	57
Tabel 3.8 Bidang pada <i>model "Unit"</i>	58
Tabel 3.9 Pilihan pada kelas Status(model.IntegerChoices)	59
Tabel 3.10 Bidang pada <i>model "Profile"</i>	60
Tabel 3.11 Bidang pada <i>model "CareerHistory"</i>	61
Tabel 3.12 Bidang pada <i>model "Auth"</i>	62
Tabel 3.13 Pilihan pada kelas Feature(model.IntegerChoices)	63
Tabel 3.14 Bidang pada <i>model "PointHistory"</i>	63

Bab 1

Pendahuluan

1.1. Latar Belakang

Semakin hari, persaingan dalam dunia manufaktur menjadi semakin ketat [1]. Di satu sisi perusahaan ingin memaksimalkan keuntungan dengan menebalkan margin keuntungan. Namun di lain sisi harga produk juga harus kompetitif agar produk dapat tetap eksis di pasar. Maka dari itu harus dilakukan pengefisienan seperti menerapkan konsep *Lean Manufacturing* dan mengurangi *MUDA* (segala sesuatu yang tidak memberikan nilai tambah pada produk) [2].

Dalam pengaplikasian *Lean Manufacturing*, sistem produksi didukung dengan konsep pengoperasian dan perawatan yang dikenal dengan *Total Productive Maintenance* (TPM). TPM merupakan konsep perawatan menyeluruh yang mengikutsertakan partisipasi seluruh karyawan, mulai dari divisi operator hingga divisi perawatan. Divisi operator menjalankan *Jishu Hozen* (*Autonomous Maintenance*), divisi perawatan menjalankan perawatan lainnya yang lebih membutuhkan keahlian perawatan. Adapun semua pihak juga berkolaborasi dalam menerapkan konsep *Kaizen* (*Continuous Improvement*) agar efisiensi dan efektivitas produksi menjadi meningkat.

Adapun selama ini kearsipan pada *Jishu Hozen* menggunakan lembar pemeriksaan fisik (kertas), yang mana pengarsipannya seringkali masih diarsipkan secara tradisional sehingga membutuhkan waktu yang cukup lama bila ingin memeriksa arsip yang lampau. Bila fenomena yang butuh penanganan, pada umumnya ditandai dengan *tag*. *Tag* putih untuk ketidaksesuaian yang dapat ditangani operator, sedangkan *tag* merah dipasang oleh operator bila membutuhkan bantuan dan keahlian dari petugas perawatan. Sistem pelaporan menggunakan *tag* ini juga memiliki resiko pengarsipan yang buruk mengingat ukuran *tag* yang kecil, serta susah untuk mencari arsip yang lampau. Dan yang terakhir dalam pengimplementasian *Kaizen*, bila ada ide perbaikan (*improvement*), ide yang telah disetujui tersebut harus diarsipkan dengan baik. Agar ide tersebut dapat dengan mudah diakses bila dibutuhkan.

Mengingat dengan semakin maju dan murahnya teknologi informasi [3], maka didapati peluang untuk mengefisienkan metode TPM yang masih tradisional tersebut. Dibutuhkan platform komunikasi yang dapat mengakomodasi seluruh pihak untuk menerima dan merespons laporan fenomena dalam lingkungan produksi. Dan tentunya platform tersebut juga harus memiliki arsitektur kearsipan data yang baik. Selain itu untuk meningkatkan semangat

partisipasi seluruh pihak yang terlibat untuk melaporkan fenomena, juga dibutuhkan skema insentif [4].

1.2. Rumusan Masalah

Berdasarkan latar belakang pada sub bab 1.1, maka didapati beberapa masalah yang dipaparkan sebagai berikut:

1. Bagaimana sistem pelaporan dan kearsipan harus dikembangkan agar tidak bergantung pada media fisik (kertas) sehingga lebih efisien dan awet?
2. Bagaimana sistem pelaporan dan kearsipan harus dikembangkan agar arsip pelaporan tertentu dapat didapatkan kembali (*retrieve*) dengan mudah di lain waktu kedepannya?
3. Bagaimana sistem pelaporan harus dikembangkan agar dapat memberi kesempatan pada seluruh pihak untuk berpartisipasi dalam perbaikan perusahaan?
4. Bagaimana sistem pelaporan harus dikembangkan agar dapat meningkatkan rasa urgensi pada seluruh pihak untuk berpartisipasi dalam perbaikan perusahaan?

1.3. Tujuan

Dari rumusan masalah pada sub bab 1.2, maka dapat diturunkan beberapa tujuan pengerjaan tugas akhir ini yang dipaparkan sebagai berikut:

1. Merancang dan membuat platform berbasis web yang dapat mengakomodasi kegiatan pelaporan, pengarsipan, dan manajemen tugas oleh pihak-pihak yang terlibat pada lingkungan produksi.
2. Merancang dan membuat basis data dan laman pada platform berbasis web yang mengakomodasi pengambilan kembali (*retrieving*) arsip dan data statistik terkait dengan terstruktur dan mudah.
3. Merancang dan mengimplementasi jaringan untuk mengakses platform berbasis web yang memungkinkan pihak-pihak yang terlibat pada lingkungan produksi pada suatu rantai produksi untuk mengakses platform tersebut.
4. Merancang skema dan membuat laman antarmuka pada platform berbasis web yang dapat memberi insentif pada pihak-pihak yang terlibat pada lingkungan produksi yang telah berpartisipasi pada perbaikan perusahaan melalui platform berbasis web.
5. Merancang skema dan membuat laman antarmuka penunjang pada platform berbasis web yang menunjang fungsi dan skema utama seperti pelaporan, kegiatan terkait pengarsipan, serta skema insentif.

1.4. Batasan Masalah

Karena dibatasi oleh sumber daya, data, serta waktu, maka pengerjaan tugas akhir ini dibatasi oleh hal-hal yang dipaparkan sebagai berikut:

1. Sistem pelaporan dan dokumentasi diimplementasikan pada lingkungan produksi berbasis *lean manufacturing*.
2. Yang dimaksud dengan pihak-pihak yang terlibat pada lingkungan produksi adalah pihak yang secara langsung dan rutin turun ke lantai produksi dan pemangku kepentingan yang membutuhkan informasi dan arsip terkait.
3. Platform pelaporan berbasis web hanya bisa diakses melalui jaringan WLAN dimana komputer peladen (*host*) terhubung.

1.5. Manfaat

Pengerjaan tugas akhir ini diharapkan membawa manfaat sebagai berikut:

1. Sistem memberi lebih banyak data kualitatif kepada perusahaan terkait.
2. Sistem pelaporan menjadi lebih mudah dan lebih murah karena tidak membutuhkan media fisik.
3. Sistem kearsipan menjadi lebih awet dan lebih aman karena arsip dapat dicadangkan (*back up*) kepada penyimpanan awan (*cloud storage*).
4. Sistem kearsipan menjadi lebih mudah untuk diakses kembali (*retrieve*) dengan lebih mudah dan efisien karena disimpan secara terstruktur.
5. Sistem yang dapat diakses melalui komputer dan ponsel pintar memungkinkan lebih banyak pihak berpartisipasi pada usaha *kaizen* perusahaan.
6. Sistem dengan skema insentif ini meningkatkan urgensi dan semangat pihak-pihak di perusahaan untuk berpartisipasi lebih pada usaha *kaizen* perusahaan.
7. Sistem membantu pemangku kepentingan (*stakeholder*) untuk mengambil keputusan yang lebih baik dengan menyediakan lebih banyak data.
8. Sistem membantu mempersingkat penanganan masalah yang sama bila terulang dengan menyediakan data historis.

1.6. Sistematika Penulisan

Selanjutnya konten tugas sarjana ini akan dibagi dalam lima bab yang dipaparkan sebagai berikut:

1. Bab 1 Pendahuluan

Bab 1 menjelaskan masalah dan kondisi yang melatarbelakangi penelitian tugas sarjana. Dari masalah dan kondisi tersebut kemudian diperas menjadi pertanyaan kunci. Kemudian ditentukan tujuan tugas sarjana ini agar pertanyaan tersebut dapat terjawab. Adapun dipaparkan juga batasan masalah yang membatasi cakupan penelitian. Diperkirakan juga manfaat yang ditelurkan dari pengerjaan tugas akhir ini. Dan yang terakhir dituliskan sistematika penulisan agar pembaca dapat mengerti alur penulisan tugas akhir ini.

2. Bab 2 Tinjauan Pustaka

Bab 2 memuat studi-studi terkait yang akan memberi gambaran umum dan menjadi dasar pengetahuan dalam merancang dan mengembangkan sistem terkait. Pengetahuan yang menjadi dasar dari perancangan sistem akan berupa beberapa topik multidisiplin terapan mulai dari *lean manufacturing*, kearsipan digital bisnis, gamifikasi, dan manajemen tugas. Adapun pengetahuan yang menjadi dasar dari pengembangan sistem akan berupa beberapa topik teknis mengenai perangkat lunak seperti python, aplikasi web, Django, dan jaringan WLAN.

3. Bab 3 Pengembangan Perangkat Lunak

Bab 3 menjabarkan proses yang dilaksanakan penulis mulai dari merencanakan pengembangan, menganalisis kebutuhan, merancang hingga mengembangkan aplikasi web. Pada awalnya akan dijabarkan metodologi penelitian tugas akhir. Metodologi tersebut dimulai dari pencarian kebutuhan untuk menyelesaikan permasalahan yang dibahas di Bab 1 dengan bantuan dasar pengetahuan dari Bab 2. Kemudian dirancang interaksi pengguna pada aplikasi web yang dapat menjawab seluruh kebutuhan yang telah terkumpul. Diteruskan pada perancangan konsep-konsep abstraksi pada aplikasi web yang menopang struktur basis data aplikasi web. Dan yang terakhir adalah mengembangkan aplikasi web yang dapat mengakomodasi seluruh interaksi yang telah dirancang.

4. Bab 4 Pengujian dan Analisis

Bab 4 menjabarkan pengujian yang dilakukan penulis pada fitur aplikasi web dan kinerja aplikasi web pada jaringan WLAN. Pada pengujian aplikasi web akan diuji setiap laman dari suatu fitur. Dari laman tersebut akan diuji aksesibilitas, interaksi elemen, dan pengunggahan datanya bila memungkinkan. Sedangkan untuk pengujian kinerja aplikasi web pada jaringan WLAN, pengguna menggunakan dua perangkat lain yang terhubung dengan jaringan WLAN.

5. Bab 5 Kesimpulan dan Saran

Bab 5 akan memaparkan kesimpulan dan saran dari penelitian ini. Pada bab ini juga terdapat tautan alamat dari repositori kode dari aplikasi web yang telah dikembangkan penulis.

Bab 2

Tinjauan Pustaka

2.1. *Lean Manufacturing System*

Lean Manufacturing System adalah sistem manufaktur yang bersifat ramping dan lincah. Tujuan diterapkannya *Lean Manufacturing System* adalah untuk mengurangi biaya produksi dengan menghilangkan *MUDA* (kegiatan tanpa nilai tambah), *MURA* (ketimpangan proses produksi), dan *MURI* (pembebanan berlebihan) [2].

Untuk menerapkan *Lean Manufacturing System*, terdapat persyaratan dasar yang harus dipenuhi. Syarat tersebut adalah :

1. Jaminan keselamatan kerja
2. 5S (*Seiri, Seiton, Seisou, Seiketsu, Shitsuke*)
3. Jaminan kualitas
4. Jaminan stabilitas produksi dan ketepatan pengiriman
5. Stabilitas tingkat kehadiran pegawai

Dalam pengimplementasiannya *Lean Manufacturing System* didukung oleh 2 pilar utama. Pilar pertama adalah prinsip *Just in Time* (JIT) dan prinsip kedua adalah prinsip *Jidoka*. Prinsip JIT adalah prinsip untuk memproduksi atau menyediakan hanya barang yang diperlukan, saat diperlukan, dan dalam jumlah yang diperlukan. Pelaksanaan JIT didasari oleh 3 prinsip yaitu pengaliran proses produksi, produksi dengan *tack time*, dan menerapkan *pull system* (sistem tarik). Prinsip *Jidoka* adalah mekanisme untuk mencegah terjadinya produksi barang NG (*not good*). Pada prinsip ini, kegiatan produksi akan berhenti bila terdeteksi ketidaknormalan pada proses kerja. Dengan begitu ketidaknormalan dapat diselesaikan secepatnya sehingga kualitas produk terkait dan produk lain yang kan dikerjakan akan terjamin.

2.1.1. *MUDA*

MUDA adalah aktivitas pada kegiatan produksi yang tidak memberikan nilai tambah pada produk [2]. Untuk meningkatkan efisiensi produksi, maka *MUDA* harus ditekan agar biaya produksi produk dapat menjadi kompetitif. Namun terdapat juga kegiatan *MUDA* lainnya yang perlu dilakukan walaupun tidak memberi nilai tambah pada produk, aktivitas tersebut seperti

penggantian perkakas potong, *loading-unloading* benda kerja pada alat produksi, maupun transportasi benda kerja. Kita harus berusaha menekan frekuensi dan lama kegiatan tersebut.

Adapun terdapat 7 *MUDA* pada tempat kerja (*genba*), yaitu:

1. *MUDA* overproduksi
2. *MUDA* menunggu
3. *MUDA* transportasi
4. *MUDA* proses
5. *MUDA* stok
6. *MUDA* gerakan
7. *MUDA* produk NG dan perbaikan

2.1.2. *Jishu Hozen (Autonomous Maintenance)*

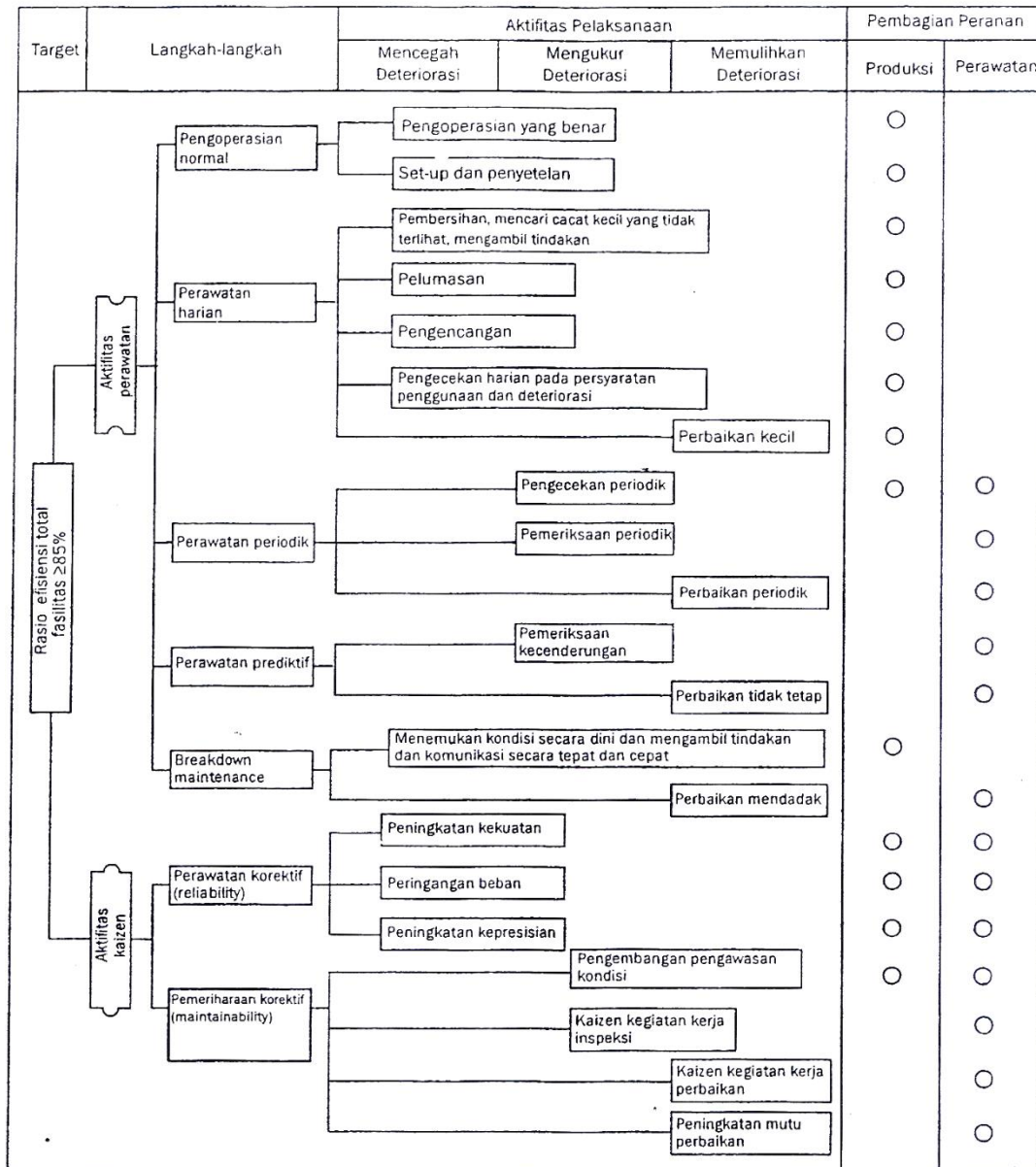
Lean Manufacturing System menggunakan konsep perawatan yang dinamakan dengan *Total Productive Maintenance* (TPM) [2]. Pada perawatan TPM, terdapat konsep *Jishu Hozen (Autonomous Maintenance)*, dimana operator juga turut serta dalam menjaga fasilitas produksi. Operator diharuskan melakukan pencegahan deteriorasi seperti pelumasan, pengukuran deteriorasi seperti pengecekan harian, serta pemulihan deteriorasi ringan seperti perbaikan kecil.

Pada *Jishu Hozen* ini terdapat dokumen-dokumen kontrol seperti lembar standar dan *tag*. Operator yang menjalankan *Jishu Hozen* akan mengisi lembar standar tersebut untuk setiap perawatan dan pemeriksaan ringan. Bila terdapat ketidaknormalan, operator akan berefleksi apakah dia bisa menangani itu? Bila bisa operator akan menanganinya dan memasang *tag* putih pada bagian yang bermasalah. Sedangkan bila operator tidak dapat menanganinya, operator dapat meminta bantuan petugas perawatan dengan memasang *tag* merah pada bagian yang bermasalah.

2.1.3. *Kaizen (Continuous Improvement)*

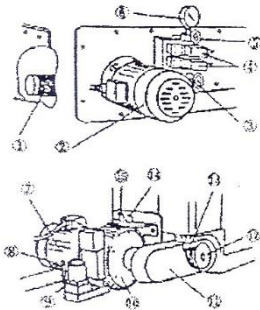
Kaizen merupakan sebuah prinsip dimana perubahan atau perbaikan berkelanjutan [2]. Dengan terus melakukan perbaikan kecil, sistem menjadi tidak “terkejut”. Secara konsep prinsip *kaizen* baik untuk diterapkan, namun pada pelaksanaannya seringkali ide *kaizen* tidak terdokumentasi dengan baik sehingga budaya ini menjadi tidak terlalu efektif. Selain itu walaupun manajemen telah mengajak seluruh pihak terkait untuk mengimplementasikan

prinsip *kaizen*, namun pada pelaksanaannya tidak semua pihak memiliki ruang untuk memberi saran serta banyak juga pihak yang merasa kurangnya urgensi dari prinsip ini.



Gambar 2.1 Pembagian peran departemen operasi dan departemen perawatan [2]

Lembar Prosedur Kerja		Lembar Standar Sementara Jishu Hozen		Masa Berlaku	Penerbitan	Tanggal:		
Divisi		Nama Fasilitas	(ke-1)			Kepala Seksi	Group Leader	Team Leader



No.	Nama	Fungsi	Keterangan
1	Suction strainer	Alat untuk menghilangkan sampah dan benda asing	
2	Motor hidrolik	Tenaga untuk mengaktifkan pompa	0,75kW
3	Katup pengontrol tekanan	Katup yang mengontrol tekanan	
4	Solenoid	Alat yang mengubah arah aliran minyak	
5	Gauge cock	Alat yang mencegah kejutan tekanan pulsasi pada manometer	
6	Manometer	Alat ukur yang menunjukkan tekanan	
7	Main power motor	Tenaga yang menggerakkan mesin	3,7kW
8	Oil level meter	Alat yang menunjukkan kuantitas minyak pelumas transmisi	
9	Pilot motor	Tenaga yang menggerakkan planet gear	
10	Gearless drive mechanism	Alat yang mengubah rasio kecepatan tanpa roda gigi	
11	Belt cover	Penutup yang melindungi belt	
12	Pulley	Alat yang mengirimkan tenaga	
13	Belt	"	V belt A - 49
14	Lubang pengisian minyak	Lubang untuk mengisi minyak pelumas	
15	Lubang keluarnya angin	Lubang tempat keluarnya angin yang ada di dalam motor saat pengisian minyak	

	No	Bagian yang Dibersihkan	Standar	Metode	Tindakan	Waktu	Siklus			Pelaksana
							Hari	Minggu	Bulan	
Pengecekan	1	Suction strainer	Tidak boleh kotor	Penglihatan	Membersihkan saat pengecekan	5'			3 bulan	OP
	2	Motor hidrolik	Tidak boleh ada keabnormalan (bunyi, panas, bau)	Pendengaran, perabaan, penciuman	Menghentikannya (meminta perawatan)	30"		○		"
	3	Katup pengontrol tekanan	Kerja tekanan yang telah ditetapkan harus dijaga	Penglihatan	Menghentikannya (meminta perawatan)	20"		○		"
	4	Solenoid	Kepala roda gerinda harus maju-mundur dengan lancar	Penglihatan dan perabaan	Menghentikannya (meminta perawatan)	30"		○		"
	5	Gauge cock	Diaphragm harus bekerja	Penglihatan	Menggantinya	5'			○	"

Gambar 2.2 Contoh lembar prosedur kerja dan standar sementara *Jishu Hozen* [2]

TPM
Autonomous Maintenance

Step: 1 2 3 4 5

WHITE TAG (Operator) ABNORMALITY SITE
ป้ายค้นหาความผิดปกติของเครื่องจักร

TAG. NO. _____ เลขที่ป้าย

Equip. Code: _____ รหัสเครื่องจักร
Equip. Name: _____ ชื่อเครื่องจักร
Date found: ____/____/____
Found by: _____ ผู้พบ
Description: _____ รายละเอียดความผิดปกติ

Attach this tag to the relevant equipment
ติดป้ายบนชิ้นส่วนของเครื่องจักรที่มีอาการผิดปกติ

TPM
Autonomous Maintenance

Step: 1 2 3 4 5

RED TAG (Maintenance) ABNORMALITY SITE
ป้ายค้นหาความผิดปกติของเครื่องจักร

TAG. NO. _____ เลขที่ป้าย

Equip. Code: _____ รหัสเครื่องจักร
Equip. Name: _____ ชื่อเครื่องจักร
Date found: ____/____/____
Found by: _____ ผู้พบ
Description: _____ รายละเอียดความผิดปกติ

Attach this tag to the relevant equipment
ติดป้ายบนชิ้นส่วนของเครื่องจักรที่มีอาการผิดปกติ

TPM
Autonomous Maintenance

Step: 1 2 3 4 5

YELLOW TAG (Safety) ABNORMALITY SITE
ป้ายค้นหาความผิดปกติของเครื่องจักร

TAG. NO. _____ เลขที่ป้าย

Equip. Code: _____ รหัสเครื่องจักร
Equip. Name: _____ ชื่อเครื่องจักร
Date found: ____/____/____
Found by: _____ ผู้พบ
Description: _____ รายละเอียดความผิดปกติ

Attach this tag to the relevant equipment
ติดป้ายบนชิ้นส่วนของเครื่องจักรที่มีอาการผิดปกติ

Gambar 2.3 Contoh tag pada pelaksanaan *Jishu Hozen* [5]

2.2. Arsip dan Kearsipan

Arsip (*Record*) yang dibuat sebagai bagian dari proses bisnis yang terstruktur adalah sumber dokumen yang memenuhi kebutuhan komunikasi pada institusi dan digunakan secara luas untuk tujuan proses komunikasi dan autentikasi [4]. Tidak seperti dokumen lain, arsip mengandung informasi yang autentik mengenai proses bisnis.

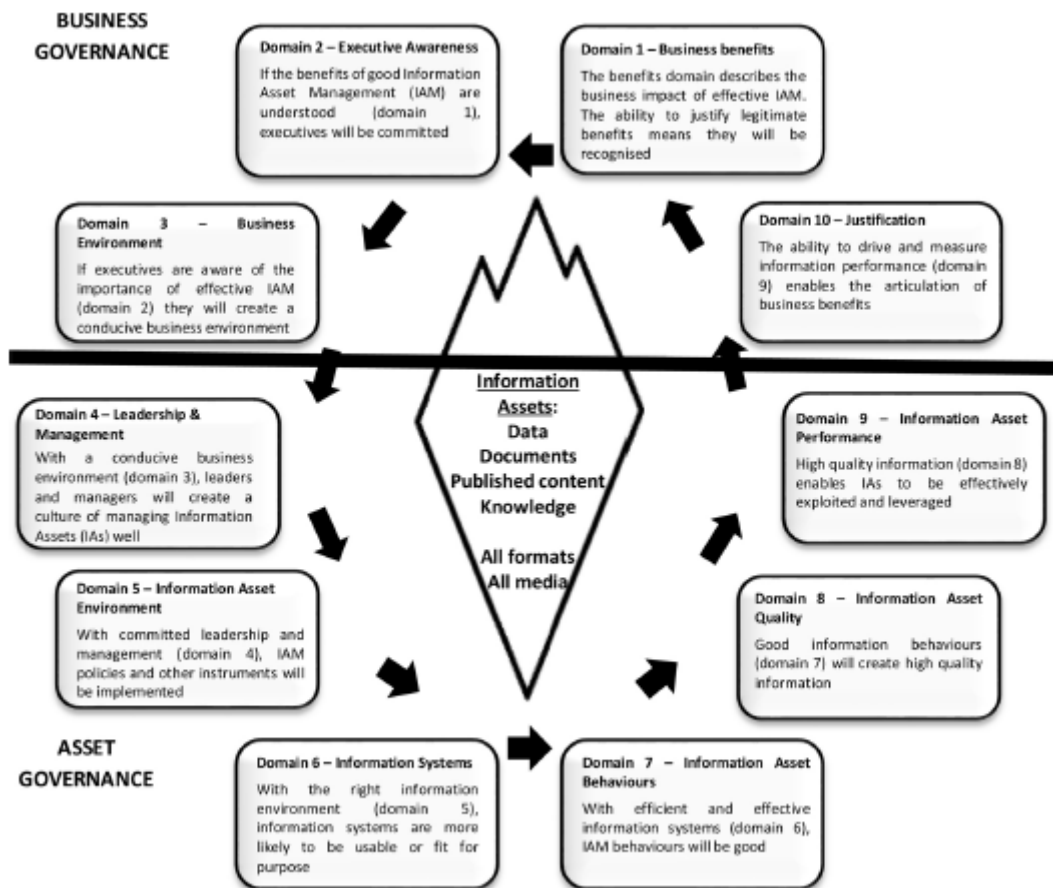
Adapun kearsipan (*records management*) atau yang biasa disebut kearsipan dan manajemen informasi (*records and information management / RIM*), menurut ISO 15489-1:2001, adalah bidang manajemen yang bertanggung jawab dalam efisiensi dan pengendalian dari penciptaan (*creation*), memberi pengakuan atau pengesahan (*receipt*), pemeliharaan (*maintenance*), penggunaan (*use*), dan pelenyapan (*disposition*) dari suatu arsip (*records*), meliputi proses mengumpulkan (*capturing*) dan memelihara (*maintaining*) bukti (*evidence*) dari informasi mengenai aktivitas bisnis dan transaksi dalam bentuk suatu arsip (*records*) [6].

Bagi institusi bisnis, arsip (*records*) juga merupakan aset institusi yang bisa disebut juga dengan aset informasi (*information assets / IA*). Selanjutnya dalam aplikasi pada dunia bisnis peran manajemen aset informasi (*information assets management / IAM*) dapat digambarkan dalam sebuah skema yang komprehensif yang bernama *Holistic IAM (HIAM)* [7]. Dalam skema HIAM, aset informasi akan mempengaruhi 10 domain bisnis yang akan dijabarkan sebagai berikut:

1. Domain 1 : Manfaat bisnis (*business benefits*), adalah dampak bisnis yang dirasakan dari manajemen aset informasi yang efektif. Sebagai contoh, berkurangnya biaya produksi dan rasio profitabilitas yang lebih tinggi.
2. Domain 2 : Kesadaran eksekutif (*executive awareness*), adalah kesadarannya pihak manajemen puncak (*top level management*) akan pentingnya sistem manajemen aset informasi pada performa perusahaan akibat manfaat yang dihasilkan. Sebagai contoh, memberikan anggaran untuk pengembangan sistem manajemen aset informasi dan mempertimbangkan pengimplementasian sistem manajemen informasi pada strategi perusahaan.
3. Domain 3 : Lingkungan kerja (*business environment*), adalah lingkungan bisnis tempat sistem diimplementasikan sehingga terdapat perubahan proses bisnis akibat sadarnya manajemen puncak. Sebagai contoh, penyediaan informasi dari suatu departemen kepada departemen lain dan pengimplementasian standar operasional prosedur (SOP) kerja baru.

4. Domain 4 : Kepemimpinan dan manajemen (*leadership and management*), adalah perubahan gaya dan budaya kepemimpinan dan manajemen akibat berubahnya lingkungan kerja. Sebagai contoh, munculnya budaya dan kebiasaan kerja yang baru setelah diimplementasikannya sistem manajemen aset informasi.
5. Domain 5 : Lingkungan informasi (*information environment*), adalah perubahan rancangan dan aturan pada lingkungan informasi akibat perubahan gaya kepemimpinan dan manajemen. Sebagai contoh, dibuatkan satu akun sistem manajemen aset informasi untuk setiap karyawan agar administrator sistem tahu siapa yang berbuat apa.
6. Domain 6 : Sistem informasi (*information system*), adalah perangkat keras maupun perangkat lunak serta instrumen lainnya yang menunjang fungsionalitas sistem manajemen aset informasi. Sebagai contoh, dengan ditambahkan fitur baru akibat perubahan lingkungan informasi, dibutuhkan perubahan kapabilitas seperti menyewa peladen (*server*) baru yang memiliki kapasitas lebih.
7. Domain 7 : Perilaku informasi (*information behavior*), adalah bagaimana institusi dan pihak-pihak di dalamnya mengelola dan bekerja dengan informasi. Sebagai contoh, dengan sistem informasi yang buruk, dimana kapasitas peladen (*server*) kurang, pekerja menjadi tidak produktif saat ingin mengakses informasi.
8. Domain 8 : Kualitas informasi (*information quality*), adalah ukuran dari kualitas dari informasi yang disampaikan seperti ketersediaan (*availability*), kelengkapan (*completeness*), kebenaran (*correctness*), maupun relevansi (*relevance*). Sebagai contoh, dengan perilaku informasi yang baik, seperti memperbaharui (*update*) informasi dengan teratur, kualitas informasi menjadi meningkat karena informasi menjadi relevan.
9. Domain 9 : Performa informasi (*information performance*), adalah potensi yang ada dan yang telah dihasilkan dari suatu informasi. Sebagai contoh, dengan kualitas informasi yang baik, informasi yang terbaharui (*updated information*), departemen logistik dapat melakukan kegiatan operasional yang lebih efisien.

10. Domain 10 : Justifikasi (*justification*), adalah kegiatan untuk menilai dan mengukur performa informasi untuk mengetahui apakah pengimplementasian suatu sistem informasi dibenarkan (*justified*) untuk kedepannya. Sebagai contoh, untuk mengukur performa informasi pada departemen logistik, dilakukan audit biaya operasional seperti bahan bakar dan pemborosan waktu.



Gambar 2.4 Skema model *Holistic Information Assets Management (HIAM)* [7]

2.3. Gamifikasi

Gamifikasi (*gamification*) adalah pengaplikasian dari prinsip-prinsip dari domain permainan (*game*) pada situasi non-permainan (*non-game*) [8]. Kegiatan gamifikasi biasanya berfokus pada suatu proses bisnis (seperti akuisisi pelanggan) maupun berfokus untuk mencapai suatu capaian (seperti jumlah penjualan). Kegiatan gamifikasi dapat melibatkan pihak-pihak baik di dalam suatu institusi (seperti meningkatkan kepuasan karyawan), maupun pihak-pihak di luar institusi (seperti untuk mengajak pengguna layanan atau pelanggan untuk berkolaborasi dalam mengembangkan produk layanan).

Gamifikasi dapat mengubah perilaku pihak terkait karena kegiatan ini mempengaruhi faktor penggerak dari perilaku manusia dalam 2 cara yang berkaitan yaitu penguatan (*reinforcements*) dan emosi. Pertama, penguatan baik positif (*positive reinforcements*) maupun negative (*negative reinforcements*) mendorong keterulangan perilaku sebagai pengondisian operan (*operant conditioning*) [9] dan hukum akibat (*law of effect*) [10]. Pendekatan ini telah lama digunakan dalam bidang psikologi untuk menjelaskan rentang dari perilaku manusia dan perubahan perilaku. Adapun akibat dari penguatan dapat membawa dampak internal yang utamanya adalah emosi yang merupakan motivator yang sangat kuat dalam perubahan perilaku [11].

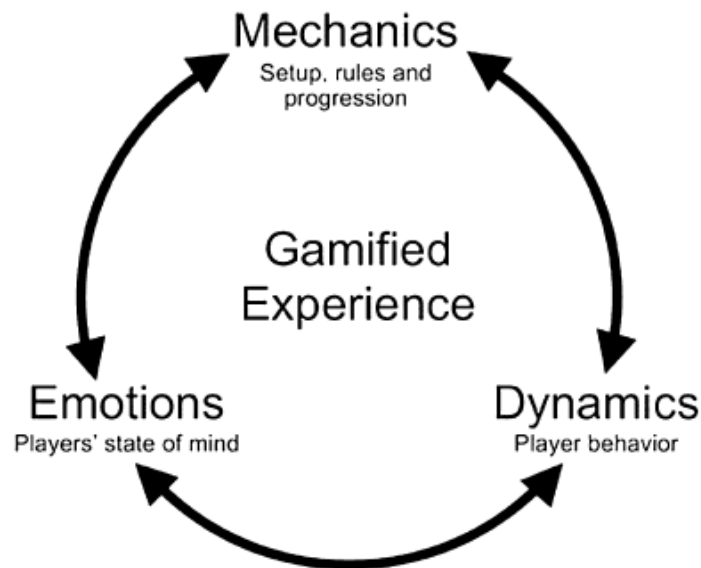
Dalam mengembangkan skema gamifikasi digunakan kerangka (*framework*) kerja yang melibatkan pihak-pihak yang terkait dengan skema gamifikasi dan kerangka kerja MDE yang diadaptasi dari literatur desain permainan (*game*) [12]. Pihak yang terkait yang dimaksud adalah:

1. Pemain (*player*), adalah pihak yang terlibat secara aktif dalam kegiatan permainan.
2. Desainer (*designer*), adalah pengambil keputusan yang mengembangkan dan merancang skema. Sering pula menjadi pihak yang bertanggung jawab dalam mengatur keberjalanan gamifikasi serta merawat sistem dan skema.
3. Penonton (*spectator*), adalah pihak yang secara tidak langsung terlibat dalam kegiatan permainan, namun kehadirannya dapat memberi pengaruh dalam skema gamifikasi.
4. Pengamat (*observer*), adalah pihak eksternal yang tidak terlibat dan tidak memberi dampak langsung pada skema gamifikasi.

Adapun prinsip yang diadopsi dari kerangka kerja MDE adalah:

1. Mekanisme (*mechanism*), adalah keputusan yang diambil desainer untuk mencapai tujuan (*goals*) tertentu, peraturan (*rules*), pengaturan (*settings*), konteks (*context*), dan batasan (*boundaries*) yang menjadi dasar skema gamifikasi. Mekanisme sudah diketahui dan ditentukan sebelum skema dijalankan dan mekanisme akan terus kontan (kecuali dengan perubahan oleh desainer) seiring berjalannya skema.
2. Dinamika (*dynamics*), adalah fenomena dan perilaku dari pihak terkait (seperti pemain, penonton, dan pengamat) yang muncul saat dan setelah berpartisipasi dalam skema gamifikasi. Berbeda dengan mekanisme yang ditentukan oleh desainer, dinamika tercipta dari bagaimana interaksi pihak terkait dengan mekanisme dan interaksi antar-pihak baik yang sama jenisnya maupun yang berlainan jenis.

3. Emosi (*emotion*), adalah keadaan mental dan reaksi yang muncul dari pihak terkait (seperti pemain, penonton, dan pengamat) saat dan setelah berpartisipasi dalam skema gamifikasi. Emosi merupakan hasil dari interaksi pihak terkait dengan mekanisme, begitu juga mekanisme dapat diatur ulang untuk mencapai keadaan emosi tertentu. Emosi juga sangat berkaitan dengan dinamika, dimana perubahan emosi menghasilkan perubahan dinamika dan perubahan dinamika menghasilkan perubahan emosi. Skema gamifikasi yang baik seharusnya menarik (*appealing*) dan berorientasi pada kesenangan (*fun-oriented*), tidak hanya berfokus pada tujuan pragmatis [13]. Skema sebaiknya diatur dalam mekanisme yang membuat pihak terkait menikmati skema agar keterlibatan pihak menjadi lebih tinggi [14].



Gambar 2.5 Kerangka kerja MDE pada skema gamifikasi [8]

2.4. Python

Python adalah bahasa komputer yang bertujuan umum (*general-purpose*), berorientasi objek (*object-oriented*), dan bersumber terbuka (*open-sourced*) [15]. Biasanya Python digunakan untuk program mandiri (*standalone programs*) dan menyekrip (*scripting*) aplikasi dalam domain yang sangat luasan beragam.

Python didesain untuk mengoptimasi produktivitas dari pengembang perangkat lunak, kualitas perangkat lunak, portabilitas program, dan komponen integrasi [15]. Program berbahasa Python dapat bekerja pada hampir semua platform yang umum digunakan, termasuk komputer bingkai utama (*mainframe computers*) dan komputer super (*supercomputers*), Unix dan Linux, Windows dan Macintosh, Java dan .NET, dan berbagai platform lainnya.

2.4.1. Fungsi

Fungsi (*functions*) adalah perangkat (*device*) yang mengelompokkan sejumlah pernyataan (*statements*) sehingga pernyataan-pernyataan tersebut dapat dijalankan (*run*) lebih dari sekali pada suatu program [16]. Fungsi juga dapat menghitung nilai dari suatu hasil dan membuat kita bisa mengatur parameter yang menjadi masukan (*input*) dari fungsi tersebut, yang mana nilainya dapat berubah setiap kali fungsi tersebut dijalankan. Fungsi adalah suatu jalan alternatif dalam pemrograman dengan tidak memprogram kode dengan operasi yang sama berulang kali. Dan bilamana kita ingin mengubah detail dari suatu fungsi, kita tidak perlu mengubah seluruh baris yang bersangkutan, kita hanya perlu mengubah satu saja.

Secara garis besar, fungsi dapat dibagi menjadi fungsi bawaan (*built-in function*) dan fungsi yang didefinisikan atau diciptakan oleh pengembang. Fungsi dapat diciptakan dengan membuat pernyataan *def* (*def statement*). Secara umum pernyataan *def* dibagi menjadi fungsi yang berisikan prosedur dan fungsi yang memberikan keluaran (*output*). Fungsi yang memberikan keluaran memiliki pernyataan *return* (*return statement*) yang biasanya terletak diakhir fungsi.

2.4.2. Pemrograman Berbasis Objek

Pemrograman berbasis objek adalah paradigma pemrograman yang berfokus pada suatu konsep “objek” yang memiliki sifat hierarki pewarisan (*inheritance hierarchy*) [16]. Pemrograman berbasis objek menawarkan cara pandang yang berbeda pada pemrograman seperti dengan meminimalkan pengulangan (*redundancy*) maupun menulis program baru dengan mengubahsuaikan (*customizing*) program yang telah ada daripada mengubah sepenuhnya (*changing*) program yang telah ada.

Penggunaan pemrograman berbasis objek pada Python bersifat opsional. Pada pemrograman yang lebih sederhana dapat hanya menggunakan fungsi, sedangkan pemrograman berbasis objek membutuhkan perencanaan yang lebih di awal. Karena itu, pemrograman berbasis objek lebih sering digunakan di pekerjaan yang bersifat strategik (pengembangan jangka panjang) daripada pada pekerjaan yang bersifat taktis (hanya terdapat sedikit waktu untuk mengembangkan).

2.4.3. Kelas

Kelas (*Class*) adalah perangkat yang digunakan untuk mengimplementasikan jenis objek yang baru pada Python yang mendukung sifat pewarisan (*inheritance*) [16]. Kelas adalah alat

utama dalam pemrograman berbasis objek pada bahasa Python. Kelas dapat dibuat dengan pernyataan kelas (*class statement*). Pada saat mendefinisikan kelas, kita dapat menambahkan bidang-bidang (*fields*) yang selanjutnya akan diisi oleh data yang disebut dengan atribut (*attributes*). Kita juga dapat memberi suatu kelas kemampuan untuk melakukan fungsi (*function*) dengan mendefinisikan metode (*method*) pada kelas tersebut.

Bila dibandingkan dengan unit pemrograman lain pada Python, kelas memiliki tiga perbedaan utama yang membuatnya sangat berguna dalam membangun objek baru:

1. Banyak objek (*multiple instances*), pada dasarnya kelas adalah pabrik untuk membuat satu atau lebih banyak objek. Setiap objek dari kelas yang sama dapat memiliki atribut yang serupa, namun dapat mengandung data yang berbeda.
2. Pengubahsuaian melalui pewarisan (*customization via inheritance*), kelas juga mendukung konsep pewarisan (*inheritance*) dari pemrograman berbasis objek, sehingga kita dapat mengembangkan (*extend*) suatu kelas baru (untuk tujuan spesifik) dengan mendefinisikan ulang (*redefining*) atribut yang spesifik di luar kelas awalnya (yang memiliki sifat umum).
3. Pelembihan operator (*operator overloading*), dengan menyediakan protokol metode (*method*) khusus, kelas dapat mendefinisikan objek yang merespon pada operasi dari fungsi bawaan (*built-in function*).

2.5. Aplikasi Web

Aplikasi web (*web application / web app*) adalah perangkat lunak yang bekerja pada peladen web (*web server*) [17]. Tidak seperti aplikasi desktop (*desktop application*) tradisional yang dijalankan oleh sistem operasi, aplikasi web harus diakses melalui peramban web (*web browser*).

Aplikasi web memiliki beberapa kelebihan bila dibandingkan dengan aplikasi *desktop* (*desktop application*) seperti pengembang perangkat lunak (*software developer*) tidak perlu mengembangkan aplikasi web pada berbagai platform, karena bila aplikasi dapat bekerja melalui peramban web, aplikasi tersebut dapat digunakan melalui komputer berbasis Windows, macOS, maupun ponsel pintar berbasis Android [17]. Adapun bila terdapat pembaruan (*update*) aplikasi, pengembang perangkat lunak tidak perlu mendistribusikan berkas pembaruan tersebut. Dengan memperbarui aplikasi pada peladen, semua pengguna dapat mengakses aplikasi yang telah diperbarui.

Selain itu, aplikasi web juga berbeda dengan situs web (*website*). Walaupun keduanya harus diakses melalui peramban web, namun terdapat beberapa hal yang membedakan keduanya [18]:

1. Tujuan utama situs web adalah membuat pengguna dapat dengan mudah untuk berselancar (*web surfing*) dan mendapatkan informasi relevan yang dibutuhkan. Sedangkan tujuan utama aplikasi web adalah untuk menjadi responsif pada aktivitas pengguna, interaktif, dan dalam memberi pengguna kemampuan untuk memanipulasi data dan membuat permintaan untuk keluaran yang berbeda.
2. Fitur utama dan keuntungan dari situs web adalah mudah untuk diakses, relatif lebih cepat untuk dikembangkan, dan lebih murah untuk dikembangkan. Sedangkan fitur utama dan keuntungan dari aplikasi web adalah pengalaman pengguna yang terpersonalisasi (*personalized user experience*), kemampuan untuk dikembangkan (*expandability*), dan mengungkit kapabilitas perangkat (*leverage device capabilities*).
3. Elemen utama dari situs web adalah Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), dan JavaScript. Sedangkan elemen utama dari aplikasi web juga HTML, CSS, serta JavaScript, namun aplikasi web juga menggunakan bahasa pemrograman tambahan seperti Ruby, PHP, dan kerangka kerja (*frameworks*) seperti Ruby on Rails, Scriptcase, Django, dan basis data.

Adapun contoh dari situs web adalah itb.ac.id, kompas.com, dan metrotvnews.com. Namun contoh dari aplikasi web adalah akademik.itb.ac.id, twitter.com, dan youtube.com.

2.6. Django

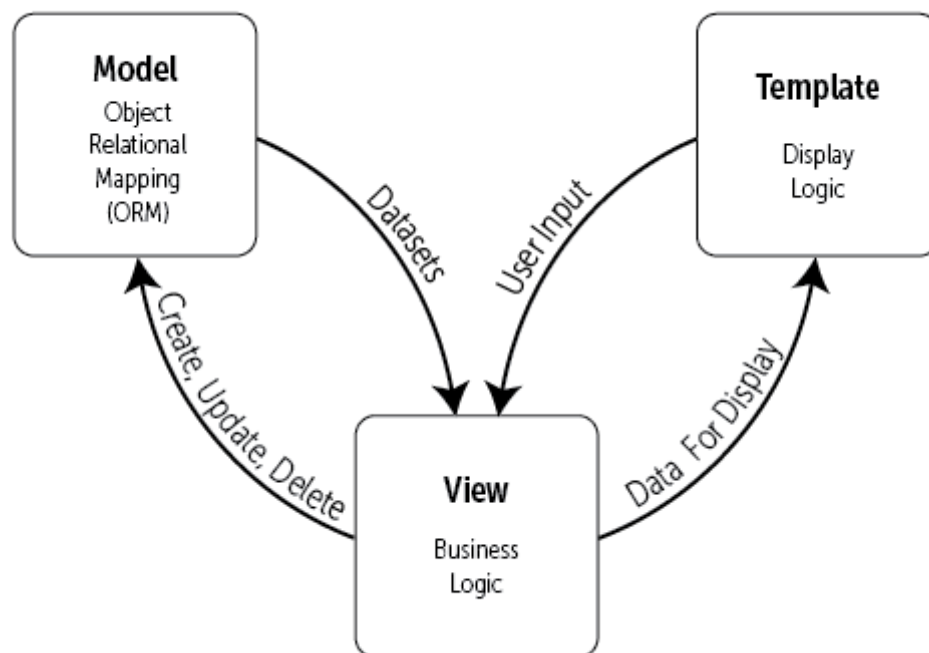
Pada dasarnya Django adalah sekumpulan *libraries* yang tertulis dalam bahasa pemrograman Python [19]. Django ditujukan untuk pengembangan aplikasi web, maka dari itu dapat dikatakan bahwa Django adalah kerangka kerja aplikasi web (*web application framework* / *WAF*).

Django menggunakan pola rancangan perangkat lunak (*software design pattern*) MTV. Pada pola rancangan MTV terdapat beberapa layar sebagai berikut:

1. *Model*, adalah komponen utama pada lapisan abstraksi (*abstraction layer*). Lapisan ini mengandung abstraksi dari data pada aplikasi web sehingga membuat kita bisa menstrukturkan dan memanipulasi data. *Model* pada lapisan ini akan menyediakan data kepada *view* pada lapisan logika bisnis.

2. *Template*, adalah komponen utama pada lapisan penyajian (*presentation layer*). Lapisan ini mengandung keputusan mengenai penyajian seperti bagaimana penyajian harus ditampilkan pada laman web atau tipe dokumen lain. *Template* pada lapisan ini akan memberikan data masukan (*input data*) kepada *view* pada lapisan logika bisnis.
3. *View*, adalah komponen utama pada lapisan logika bisnis (*business logic layer*). Lapisan ini mengandung logika yang memberi perintah pada *model* beserta data terkait. Lapisan ini juga mengatur tampilan dengan mengakses *template* yang sesuai serta menyediakan data untuk ditampilkan oleh *template*. Dapat diibaratkan bahwa lapisan ini yang menghubungkan *model* dan *template*.

Namun bagi pengembang perangkat lunak yang sudah terbiasa dengan pola perancangan MVC (*Model-View-Control*), kita dapat menganalogikan “*view*” pada MVC sebagai “*template*” pada MTV dan “*control*” pada MVC sebagai “*view*” pada MTV karena pada dasarnya pola perancangan MVC dan MTV serupa [19] [20].



Gambar 2.6 Konsep pola perancangan MTV oleh Django [19]

Sebagai kerangka kerja pengembangan yang berfokus untuk menyelesaikan pekerjaan (*getting things done*), Django akan menangani membuat dan mengatur berkas pengaturan (*settings*) dasar sehingga kita dapat fokus memprogram fungsi utama dari aplikasi web [19] [21]. Maka dari itu saat ingin memulai suatu proyek pengembangan perangkat lunak, akan

dibuat proyek (*project*) dengan perintah (*command*) “*django-admin startproject [nama proyek]*” dan mengisi nama proyek sesuai dengan keinginan dan kebutuhan pengembang.

Agar kumpulan kode yang berisi fungsi utama dari suatu proyek (*project*) Django menjadi teratur, terdapat konsep aplikasi (*apps*). Aplikasi adalah struktur direktori yang berisi kumpulan kode pada suatu proyek Django [19] [21]. Aplikasi dapat dibuat dengan perintah (*command*) “*manage.py startapp [nama aplikasi]*” dan mengisi nama aplikasi sesuai dengan keinginan dan kebutuhan pengembang. Bila aplikasi telah dibuat, akan terlihat muncul berkas-berkas (*files*) dasar yang berbasis MTV yang menunjang kerangka kerja Django.

2.6.1. Lapisan Abstraksi

Model adalah sumber tunggal dari sumber informasi mengenai data [22]. *Model* mengandung bidang esensial (*essential fields*) dan perilaku data yang disimpan. Secara umum setiap *model* terhubung dengan satu basis data. Pada dasarnya setiap *model* adalah kelas (*class*) Python yang merupakan subkelas (*subclasses*) dari “*django.db.models.Model*” dan setiap atribut (*attributes*) dari *model* mewakili bidang (*fields*) pada basis data.

Bagian terpenting dari *model* adalah larik (*list*) dari bidang basis data (*database fields*), sehingga bidang (*fields*) harus ada dalam suatu *model* [22]. Walaupun menjadi atribut (*attributes*) dari kelas (*class*) *model*, bidang (*fields*) juga merupakan objek (*instance*) dari kelas bidang (*field class*). Karena bidang menggambarkan kolom pada suatu basis data [23], jenis dari kelas bidang (*field types*) yang dipilih menentukan jenis data yang disimpan pada basis data. Sebagai contoh “*IntegerField*” untuk menyimpan data berbentuk bilangan bulat, “*Charfield*” untuk menyimpan data berbentuk string dengan panjang yang ditentukan, dan “*ImageField*” untuk memfasilitasi penyimpanan data berbentuk gambar.

Adapun data yang disimpan dalam suatu basis data juga dapat berhubungan dengan basis data lainnya. Adapun hubungan (*relationship*) antara basis data dapat dibagi menjadi 3:

1. Hubungan banyak-dengan-satu (*many-to-one relationships*), adalah hubungan yang menghubungkan suatu bidang pada satu model dengan suatu bidang pada banyak model lainnya. Contoh sederhananya setiap jurusan di Institut Teknologi Bandung (ITB) dapat meluluskan banyak mahasiswa, namun setiap mahasiswa hanya bisa lulus dari satu jurusan. Pada kerangka kerja Django hubungan ini dapat diciptakan dengan jenis kelas bidang (*field type*) “*ForeignKey*”.
2. Hubungan banyak-dengan-banyak (*many-to-many relationships*), adalah hubungan yang menghubungkan suatu bidang pada banyak model dengan suatu bidang pada

- banyak model lainnya. Contoh sederhananya setiap mahasiswa di ITB dapat menghadiri beberapa kelas dan setiap kelas dapat dihadiri banyak mahasiswa. Pada kerangka kerja Django hubungan ini dapat diciptakan dengan jenis kelas bidang “*ManyToManyField*”.
3. Hubungan satu-dengan-satu (*one-to-one relationships*), adalah hubungan yang menghubungkan suatu bidang pada satu model dengan suatu bidang pada satu model lainnya. Contoh sederhananya setiap mahasiswa di ITB diberikan satu akun akademik.itb.ac.id dan 1 akun akademik.itb.ac.id hanya diberikan kepada satu mahasiswa. Pada kerangka kerja Django hubungan ini dapat diciptakan dengan jenis kelas bidang “*OneToOneField*”.

2.6.2. Lapisan Penyajian

Template adalah berkas teks (*text documents*) yang berisi kode dalam Django Template Language (DTL) [23]. DTL didesain sehingga pengembang yang terbiasa menggunakan HTML dapat menggunakan DTL dengan mudah [25]. DTL merupakan satu-satunya pilihan bawaan (*built-in*) yang disediakan. Namun bila ada suatu kebutuhan tertentu, pengembang dapat menggunakan bahasa *template* berbasis teks (*text-based template language*) lainnya seperti Smarty maupun Jinja2.

Pada *template* terdapat dua komponen utama yaitu *variables* dan *tags* [23]. *Variable* mengeluarkan nilai dari suatu konteks (*context*) menggunakan *dictionary*, sebuah fitur pada bahasa Python yang memetakan *keys* dengan *values*. Sedangkan *tag* menyediakan proses yang dibutuhkan pada saat menjalankan (*rendering*) proses. Contoh proses yang dapat dibuat dengan *tag* adalah logika jika-maka (*if-else*) atau proses iterasi (*for loops*).

2.6.3. Lapisan Logika Bisnis

Sebuah fungsi *view* (*view function*), atau disingkat *view*, adalah fungsi dalam bahasa Python yang menerima permintaan web (*web request*) dalam bentuk *HttpRequest* dan menghasilkan respons web (*web response*) dalam bentuk *HttpResponse* [19][26]. Respons web dapat berupa konten HTML berbentuk lama web, sebuah *redirect*, maupun sebuah pesan galat 404 (*404 error*). *View* mengandung logika fitur yang dibutuhkan untuk mengeluarkan suatu respons web yang diinginkan. *View* sebenarnya dapat berada dimanapun di dalam direktori proyek (*project directory*). Maka dari itu sudah kesepakatan bersama (*convention*) untuk meletakkan *view* pada berkas (*file*) yang bernama “*views.py*”, yang diletakkan pada direktori proyek maupun aplikasi.

Selain berbentuk fungsi Python, terdapat juga *views* berbasis kelas (*class-based views*) yang berbentuk kelas Python [27]. *Views* berbasis kelas dapat membuat kita bisa menyusun *views* dan menggunakan ulang kode dengan menggunakan fitur pewarisan (*inheritance*) serta menggunakan *mixins*. Django menyediakan *views* berbasis kelas umum bawaan (*built-in class-based generic views*) untuk kegunaan yang umum sehingga pengembang tidak perlu menghabiskan waktu untuk membuat *views* yang umum. Namun untuk kebutuhan khusus, akan lebih mudah bila menggunakan *views* berbasis fungsi.

Suatu *template* akan muncul di peramban web bila dipanggil oleh suatu *view* dan suatu *view* akan bekerja bila dipanggil oleh suatu *Uniform Resource Locator* (URL) (Lokator Sumber Seragam) atau yang kita biasa kenal sebagai alamat web [26][28]. Untuk menghubungkan *view* dengan URL, Django menggunakan *URLconf* [19]. *URLconf* seperti sebuah tabel isi dari aplikasi web berbasis Django yang memetakan satu URL dengan satu *view*.

2.7. Manajemen Tugas

Manajemen tugas (*task management*) adalah proses memungkinkan (*enable*) para individu pada suatu organisasi maupun pada organisasi gabungan (*joint organization*) untuk mengambil tindakan (*action*) [29]. Tindakan yang diambil ini dipengaruhi oleh pengetahuan bersama yang berasal dari para individu atau organisasi itu sendiri. Pengambilan tindakan ini didorong oleh tujuan untuk mencapai suatu capaian. Capaian tersebut biasanya dapat dicapai dengan beberapa cara. Dimana cara-cara ini ditentukan berdasarkan situasi dan kondisi yang sedang berlangsung. Maka dari itu pengetahuan bersama yang mendasari penentuan tindakan adalah sumber daya yang dinamis, sehingga pengetahuan tersebut dapat membantu individu menentukan tindakan yang efisien serta tidak mubazir (*redundant*). Sebagai contoh dalam sistem kerja tanpa manajemen tugas atau dengan manajemen tugas yang buruk, ibu A dan bapak B yang berasal dari tim yang berbeda dapat mengerjakan tugas yang sama sehingga kinerja mereka menjadi mubazir (*MUDA* overproduksi). Ataupun karena manajemen tugas yang buruk, beban kerja ibu A menjadi jauh berat dari bapak B dan bapak B menjadi menganggur sehingga terdapat *MURA* (ketimpangan pekerjaan), *MURI* (pembebanan berlebihan) pada ibu A, dan *MUDA* menunggu pada bapak B. Contoh di atas dapat terjadi karena tidak adanya pengetahuan bersama atau karena pengetahuan bersama mereka tidak terbaharui (*up to date*).

2.7.1. Siklus Hidup Tugas

Tugas (*task*) yang harus dikerjakan dapat dibagi-bagi berdasarkan siklus hidupnya. Berdasarkan penelitian penulis, setiap perusahaan penyedia jasa perangkat lunak manajemen tugas memiliki teori siklus hidup tugas (*task life cycle*) yang berbeda, namun secara garis besar memiliki kemiripan. Berikut adalah teori siklus hidup tugas menurut perusahaan penyedia perangkat lunak :

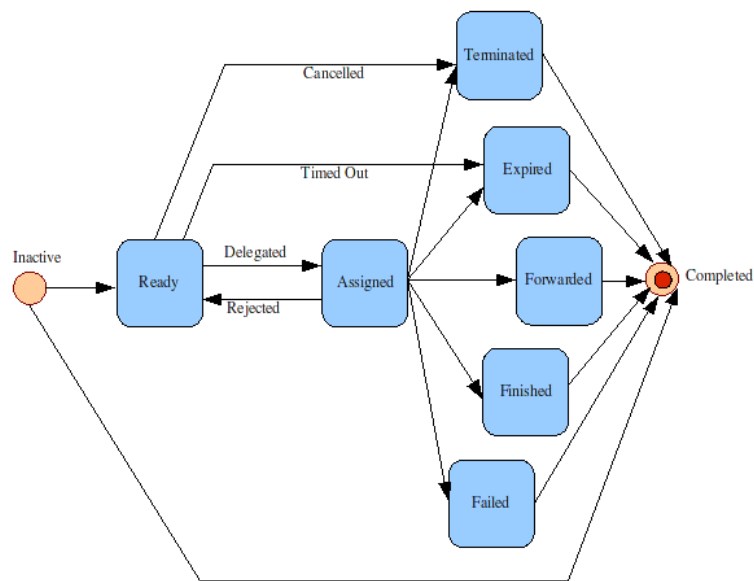
1. Teori siklus hidup tugas menurut IBM

Menurut IBM, terdapat beberapa keadaan (*state*) tugas dari tidak aktif (*inactive*) hingga tugas tersebut terselesaikan (*completed*). Pada mulanya tugas berada dalam keadaan tidak aktif hingga terjadi masalah yang membuat tugas tersebut menjadi siap dikerjakan (*ready*). Kasus tugas tersebut dapat ditutup (*terminated*) karena dibatalkan (*cancelled*) atau tugas tersebut habis tenggat waktunya (*expired*) karena kehabisan waktu (*timed out*) sebelum ditugaskan sehingga menyebabkan kasus tugas terselesaikan (*completed*) tanpa ditugaskan kepada entitas manapun.

Pada kasus yang lebih baik, tugas tersebut ditugaskan (*assigned*) kepada suatu entitas yang didelegasikan (*delegated*), namun entitas tersebut juga dapat menolak (*reject*) tugas yang telah didelegasikan.

Tugas yang telah ditugaskan dan diterima kepada suatu entitas dapat berakhir (*completed*) dari entitas tersebut karena empat kemungkinan. Kemungkinan pertama adalah karena kehabisan waktu (*expired*). Kemungkinan kedua adalah dengan diteruskan kepada entitas lain (*forwarded*). Kemungkinan ketiga adalah karena tugas tersebut gagal dijalankan (*failed*). Dan kemungkinan terakhir adalah karena tugas tersebut telah terselesaikan (*finished*) dengan baik.

Tugas yang selesai (*completed*) akan menjadi tidak aktif (*inactive*) hingga pada suatu waktu tugas tersebut perlu untuk dilakukan kembali.



Gambar 2.7 Teori siklus hidup tugas (*task life cycle*) menurut IBM [30]

2. Teori siklus hidup tugas menurut BMC

Menurut BMC, setiap entitas bisnis dapat memiliki siklus hidup tugas yang berbeda. Namun menurut pandangan BMC, berdasarkan siklus hidup tugas manual asali (*default life cycle of a manual task*) terdapat beberapa status kasus tugas, mulai dari diadakan (*staged*) hingga kasus tugas ditutup (*closed*) atau dibatalkan (*cancelled*).

Pada pengaturan asali (*default settings*), status kasus tugas adalah diadakan (*staged*) ketika akan dieksekusi. Setelah diadakan (*staged*), tugas dapat dibatalkan (*cancelled*) atau dapat ditugaskan (*assigned*) kepada entitas tertentu.

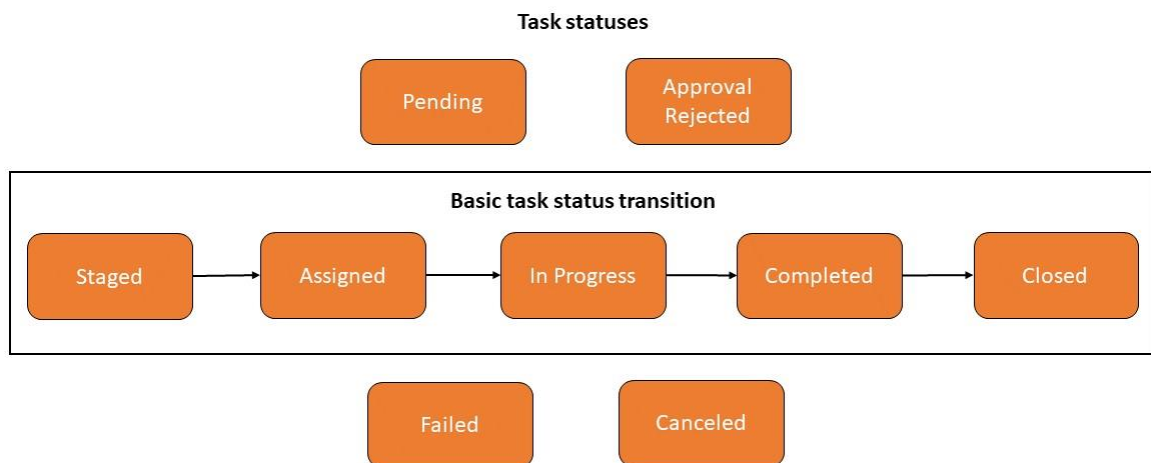
Tugas pada status ditugaskan (*staged*) dapat berlanjut pada tiga kondisi. Kondisi pertama saat entitas yang ditugaskan mengerjakan tugas sehingga tugas dalam pengerjaan (*in progress*). Kondisi kedua dimana entitas yang bertugas membutuhkan respons atau data tertentu pada pemberi tugas sehingga tugas ditunda (*pending*). Dan kondisi terakhir adalah pemberi tugas membatalkan tugas (*cancelled*).

Bila tugas berhasil mencapai status dalam pengerjaan (*in progress*), tugas dapat berlanjut pada empat kondisi. Kondisi pertama bila pemberi tugas memindahtugaskan tugas tersebut dari suatu entitas ke entitas lainnya sehingga tugas tersebut ditugaskan ulang (*assigned*). Kondisi kedua ketika tugas terselesaikan sehingga tugas telah selesai (*completed*). Kondisi ketiga ketika tugas gagal diselesaikan sehingga tugas telah gagal (*failed*). Kondisi terakhir adalah ketika pemberi tugas membatalkan tugas sehingga tugas dibatalkan (*cancelled*).

Apabila tugas dalam status ditunda (*pending*), terdapat empat kemungkinan dari kelanjutan tugas. Kemungkinan pertama adalah respons atau data yang dibutuhkan telah diberikan dan entitas yang ditugaskan mulai mengerjakan tugas sehingga status tugas menjadi dalam pengerjaan (*in progress*). Kemungkinan kedua adalah ketika pengesah tugas (*approver*) menolak tugas sehingga pengesahan tugas ditolak (*approval rejected*). Kemungkinan ketiga adalah ketika pemberi tugas menugaskan tugas tersebut kepada entitas lainnya sehingga status tugas menjadi ditugaskan (*assigned*). Dan kemungkinan terakhir adalah ketika pemberi tugas membatalkan tugas sehingga tugas dibatalkan (*cancelled*).

Ketika pengesahan tugas ditolak (*approval rejected*), terdapat tiga kemungkinan dari kelanjutan tugas. Kemungkinan pertama adalah ketika pemberi tugas ingin mengerjakan lagi tugas tersebut dan mengirimkan permintaan pengesahan lagi, bila tugas tersebut disahkan kembali dan tugas mulai dikerjakan oleh entitas yang sebelumnya mengambil tugas tersebut, status tugas tersebut akan menjadi dalam pengerjaan (*in progress*). Kemungkinan kedua adalah ketika pemberi tugas ingin melakukan perubahan sehingga tugas tersebut disahkan ulang, setelah itu tugas tersebut ditugaskan kepada entitas yang berbeda, sehingga status tugas tersebut adalah ditugaskan (*assigned*). Dan Kemungkinan terakhir adalah pemberi tugas membatalkan tugas tersebut sehingga tugas tersebut dalam status dibatalkan (*cancelled*).

Bila tugas tersebut gagal (*failed*), tugas tersebut dapat ditugaskan ulang pada entitas lain (*assigned*) atau tugas tersebut dapat diselesaikan (*completed*). Dan suatu tugas telah diselesaikan (*completed*), tugas tersebut dapat dibatalkan (*cancelled*) oleh pemberi tugas atau kasus tugas tersebut ditutup (*closed*) sehingga tidak perlu dikerjakan lagi.



Gambar 2.8 Teori siklus hidup tugas (*task life cycle*) menurut BMC [31]

3. Teori siklus hidup tugas menurut Camunda

Menurut Camunda siklus hidup dari tugas bersifat umum (*generic and common*) terhadap segala proses atau tugas. Berikut adalah teori siklus hidup yang dianggap praktik terbaik (*best practices*) sehingga digunakan menjadi penggerak belakang layar (*"behind the scenes"*) untuk setiap tugas dalam produk perangkat lunak Camunda.

Siklus hidup tugas dimulai dengan operasi penciptaan tugas (*create operation*) sehingga terbentuk tugas dengan status baru (*new state*). Tugas dengan status baru (*new state*) dapat ditawarkan kepada beberapa kandidat dengan operasi mengatur kandidat (*set candidate operation*) sehingga tugas dalam status belum ditugaskan (*unassigned status*). Tugas dengan status baru (*new status*) juga dapat ditugaskan spesifik kepada entitas tertentu dengan operasi menugaskan (*assign operation*) sehingga status tugas tersebut telah ditugaskan (*assigned status*).

Tugas dengan status belum ditugaskan (*unassigned status*) dapat diambil oleh suatu entitas dengan operasi mengklaim (*claim operation*) sehingga tugas tersebut dalam status telah ditugaskan (*assigned status*) kepada entitas yang mengklaim. Namun bila sekelompok kandidat yang ditawarkan dirasa tidak cocok, pemberi tugas dapat mengatur ulang kandidat yang akan ditawarkan dengan operasi mengatur kandidat (*set candidate operation*) sehingga tugas tersebut ditawarkan kepada sekelompok kandidat yang baru dan statusnya masih belum ditugaskan (*unassigned status*).

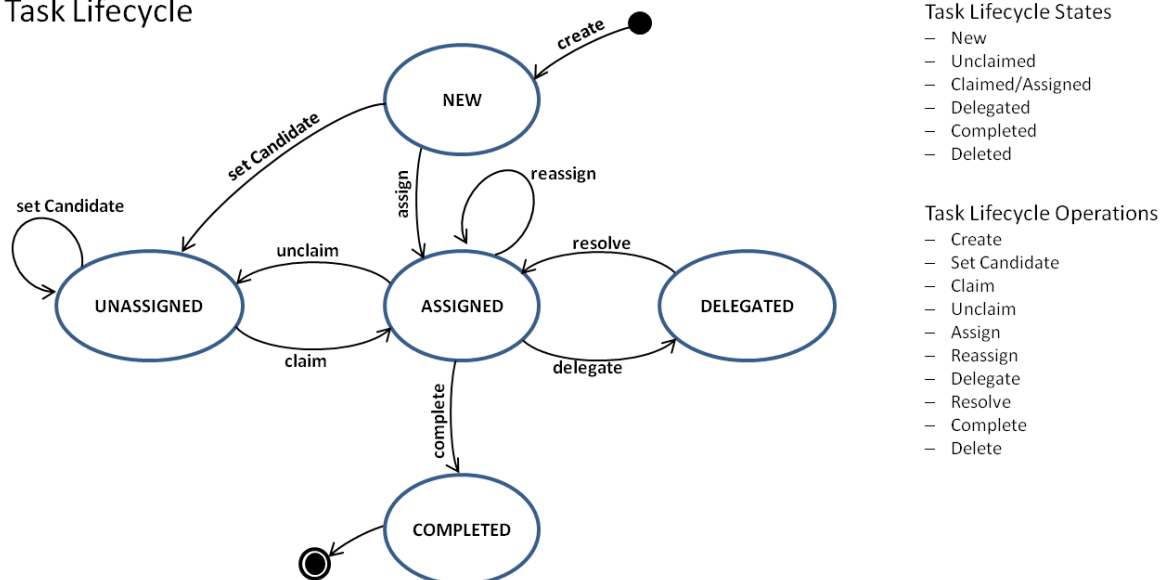
Tugas dengan status telah ditugaskan (*assigned status*) dapat berlanjut pada empat kemungkinan. Kemungkinan pertama entitas yang telah mengambil tugas tersebut ingin membebastugaskan mereka dari tugas tersebut menggunakan operasi batalkan klaim (*unclaim operation*) sehingga tugas tersebut kembali ditawarkan kepada sekelompok kandidat yang telah diatur sebelumnya dan status tugas tersebut menjadi belum ditugaskan (*unassigned status*). Kemungkinan kedua entitas yang telah mengambil tugas tersebut ingin membebastugaskan mereka dari tugas tersebut dan menugaskan tugas tersebut kepada entitas spesifik lainnya menggunakan operasi menugaskan ulang (*reassign operation*) sehingga status tugas tersebut tetap pada status ditugaskan (*assigned status*), namun dengan pengambil tugas yang berbeda. Kemungkinan ketiga entitas yang mengambil tugas tersebut ingin mendelegasikan sebagian tugas tersebut kepada entitas lainnya (yang kemudian tugas akan dikembalikan lagi kepada pengambil tugas) menggunakan operasi mendelegasikan (*delegate operation*) sehingga status tugas menjadi terdelegasikan (*delegated*). Dan kemungkinan terakhir entitas yang mengambil

tugas tersebut telah menyelesaikan tugas tersebut dan memperbaharui pengetahuan bersama dengan operasi menyelesaikan (*complete operation*) sehingga status tugas menjadi terselesaikan (*completed status*).

Tugas yang terdelegasikan memang dirancang untuk dikembalikan lagi kepada entitas yang mengambil tugas. Bila entitas yang didelegasi untuk menyelesaikan sebagian tugas telah menyelesaikan sebagian tugas yang telah didelegasikan kepadanya, entitas yang didelegasi dapat mengembalikan tugas tersebut dengan operasi telah memecahkan masalah (*resolve operation*) sehingga status tugas tersebut kembali menjadi telah ditugaskan (*assigned status*) kepada entitas yang awalnya mengambil tugas tersebut.

Bila tugas telah dalam status diselesaikan (*completed status*), tugas dapat dihilangkan dari daftar tugas dengan operasi menghapus (*delete*) sehingga tugas tersebut tidak tercatat di daftar tugas.

Task Lifecycle



Gambar 2.9 Teori siklus hidup tugas (*task life cycle*) menurut Camunda [32]

2.7.2. Metode Kanban

Kanban adalah metode untuk menentukan (*defining*), mengelola (*managing*), dan meningkatkan (*improving*) jasa (*service*) yang menghasilkan pekerjaan intelektual (*knowledge work*) seperti jasa profesional, pekerjaan kreatif, dan perancangan produk perangkat keras maupun perangkat lunak [33]. Pada awalnya kanban merupakan istilah pada *lean manufacturing* yang kemudian diadaptasikan untuk pengembangan perangkat lunak oleh David Anderson [34]. Pada pengembangan perangkat lunak kanban membantu tim agar cara mereka dalam mengembangkan perangkat lunak menjadi lebih baik.

Pada metode kanban terdapat beberapa praktik yang dapat menstabilkan dan memperbaiki sistem untuk bekerja. Berikut adalah praktik inti (*core practices*) terbaru (red: buku diterbitkan pada 2015) yang dilakukan oleh “Kanban Yahoo! Group” [34]:

Pada awalnya ikuti prinsip dasar:

- Mulai dengan apa yang anda lakukan sekarang (*start with what you do now*)
- Setuju untuk mengejar peningkatan, perubahan evolusioner (*agree to pursue incremental, evolutionary change*)
- Pada awalnya, hargai peran, tanggung jawab, dan titel pekerjaan sekarang (*initially, respect current roles, responsibilities and job titles*)

Kemudian ikuti praktik inti berikut:

- Visualisasi (*visualize*)
- Batasi pekerjaan dalam proses (*limit work in process (WIP)*)
- Kelola alur (*manage flow*)
- Buat kebijakan proses menjadi eksplisit (*make process policies explicit*)
- Terapkan jerat umpan balik (*implement feedback loops*)
- Perbaiki dengan kolaborasi, berevolusi secara eksperimental (dengan menggunakan metode atau model yang berdasar pada sains) (*improve collaboratively, evolve experimentally (using models/scientific method)*)

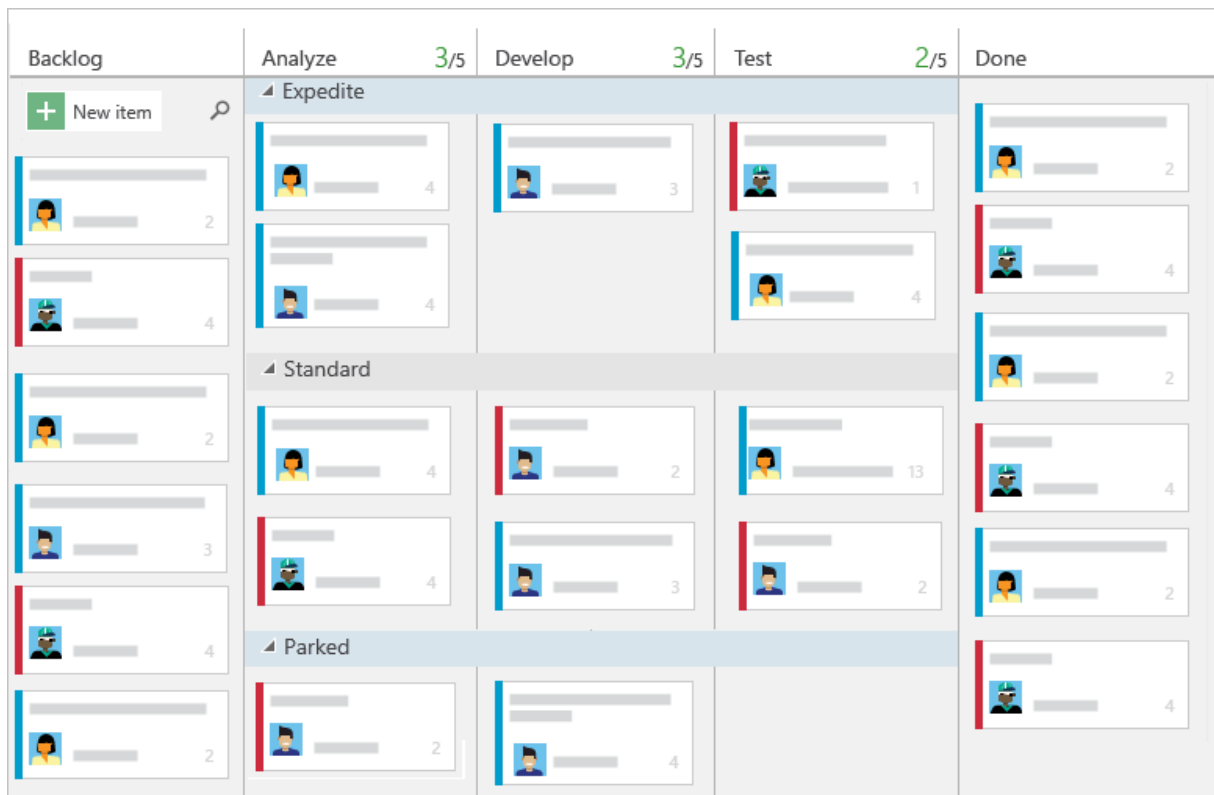
Tidak perlu menerapkan keenam praktik tersebut secara langsung pada penerapan awal. Namun diharapkan untuk menerapkan lebih banyak praktik kedepannya.

Pada metode kanban terdapat alat yang digunakan untuk memvisualisasikan alur kerja (*workflow*), alat tersebut adalah papan kanban (*kanban board*) [34]. Karena metode kanban sudah lazim digunakan, terdapat banyak contoh papan kanban, namun secara umum kita tidak bisa meniru begitu saja papan kanban lainnya karena setiap entitas memiliki alur kerja dan proses bisnis yang berbeda. Namun secara umum berikut adalah komponen papan kanban [33]:

- Aktivitas (*activity*), komponen ini menggambarkan aktivitas yang dikerjakan (*performed*) pada pekerjaan (*work item*). Bila dikumpulkan dari awal hingga akhir, kumpulan aktivitas akan membentuk alur kerja (*workflow*). Pada papan kanban, setiap aktivitas digambarkan dengan suatu kolom. Kolom aktivitas akan diurutkan dari kiri ke kanan berdasarkan urutan pada alur kerja.
- *Swinlane*, komponen ini memisahkan pekerjaan (*work item*) dalam suatu kategori, seperti berdasarkan tipe pekerjaan, konsumen, kelas jasa, ataupun kategori lainnya yang dibutuhkan. Komponen ini merupakan komponen opsional. Pada papan kanban, setiap

kategori ditempatkan pada suatu *swimlane* dan setiap *swimlane* yang digambarkan dengan suatu baris.

- Antrian (*queue*), komponen ini menjadi tempat pada alur kerja untuk menampung pekerjaan (*work item*) yang akan dikerjakan kedepannya. Pada papan kanban, komponen ini digambarkan dengan sebuah kolom yang diletakkan pada posisi paling kiri. Pada pengembangannya, komponen antrian dapat dipisah menjadi kolom antrian (*queue*) dan antrian tanpa batas (*unbounded queue*), dimana pada kolom antrian tanpa batas akan diletakkan pekerjaan yang tidak memiliki batas waktu pengerjaan.
- Kartu (*card*), komponen ini menggambarkan suatu pekerjaan (*work item*). Pada papan kanban, setiap pekerjaan akan diwakilkan dengan satu kartu, dan kartu tersebut akan berada pada suatu kolom bergantung pada tahap pengerjaan pekerjaan tersebut dan berada pada suatu baris bergantung pada jenis pekerjaan tersebut.



Gambar 2.10 Contoh papan kanban pada Microsoft Azure [35]

2.8. WLAN

Local area network (jaringan area lokal) (LAN) adalah jaringan yang dibatasi (*confined*) oleh ruang yang terbatas (*limited space*) seperti seluas bangunan tertentu atau seluas lantai tertentu [36]. LAN menggunakan teknologi jangka pendek (*short-range technology*) seperti Ethernet, Token Ring, dan teknologi sejenisnya. Suatu LAN biasanya dikendalikan oleh perusahaan atau entitas yang menggunakan LAN tersebut.

Selanjutnya LAN dikembangkan sehingga dapat menunjang komunikasi nirkabel. *Wireless LAN* (LAN nirkabel) (WLAN) merupakan jaringan LAN yang dapat terhubung tanpa kabel. Sebagai jaringan nirkabel yang digunakan secara luas, jenis jaringan WLAN distandarkan oleh IEEE (*The Institute of Electrical and Electronics Engineers*) [36].

Salah satu protokol WLAN yang paling terkenal adalah protokol 802.11 yang dipasarkan dengan nama “Wi-Fi” oleh sebuah konsorsium penyedia (*vendor*) teknologi nirkabel yang menamai diri mereka “Wi-Fi Alliance” [36]. Produk dari Wi-Fi Alliance adalah Wi-Fi 4 (IEEE 802.11n), Wi-Fi 5 (IEEE 802.11ac), dan Wi-Fi 6 (IEEE 802.11ax), serta Wi-Fi 7 (IEEE 802.11be) yang direncanakan dipasarkan 2024.

2.9. ERD

Abstraksi tertinggi untuk model data disebut diagram hubungan entitas (*Entity Relationship Diagram / ERD*) [37]. ERD terdiri dari 3 komponen :

- Entitas (*entities*), menggambarkan setiap tabel pada basis data
- Atribut (*attributes*), menggambarkan informasi yang perlu dideskripsikan di setiap entitas
- Hubungan (*relationships*), menggambarkan bagaimana hubungan antar entitas

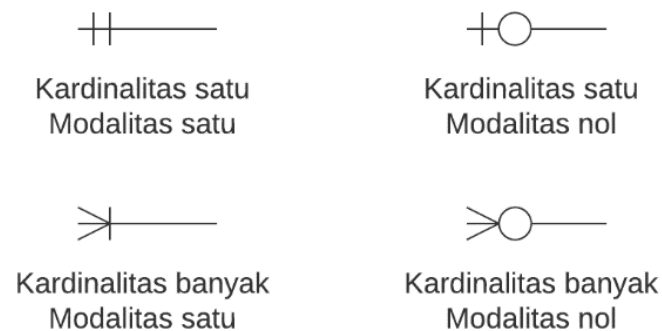
Seperti yang telah dibahas pada sub sub bab 2.6.1. mengenai lapisan abstraksi pada Django, terdapat 3 jenis hubungan antar tabel. Hubungan tersebut adalah hubungan satu-dengan-satu, hubungan satu-dengan-banyak, dan hubungan banyak-dengan-banyak. Bila digambarkan dengan ERD, ketiga hubungan tersebut dapat digambarkan seperti pada contoh pada gambar 2.11.

Pada pengembangan ERD yang lebih jauh menggunakan notasi kaki gagak (*crow's foot notation*) [38]. Dimana diperkenalkan konsep kardinalitas (*cardinality*) dan modalitas (*modality*). Kardinalitas menggambarkan jumlah maksimal hubungan dari suatu entitas. Kardinalitas dapat berupa satu atau banyak. Kardinalitas satu digambarkan dengan garis tegak

lurus di garis hubungan di ujung garis hubungan, sedangkan kardinalitas banyak digambarkan dengan cabang tiga di ujung garis hubungan. Sedangkan modalitas menggambarkan jumlah minimum hubungan suatu entitas. Modalitas dapat berupa nol (tidak ada hubungan) atau satu. Modalitas nol dapat digambarkan dengan lingkaran pada garis hubungan di sebelah dalam notasi kardinalitas, sedangkan modalitas satu digambarkan dengan garis tegak lurus pada garis hubungan di sebelah dalam notasi kardinalitas. Contoh notasi kaki gagak digambarkan pada gambar 2.12.



Gambar 2.11 Hubungan antar-entitas pada basis data



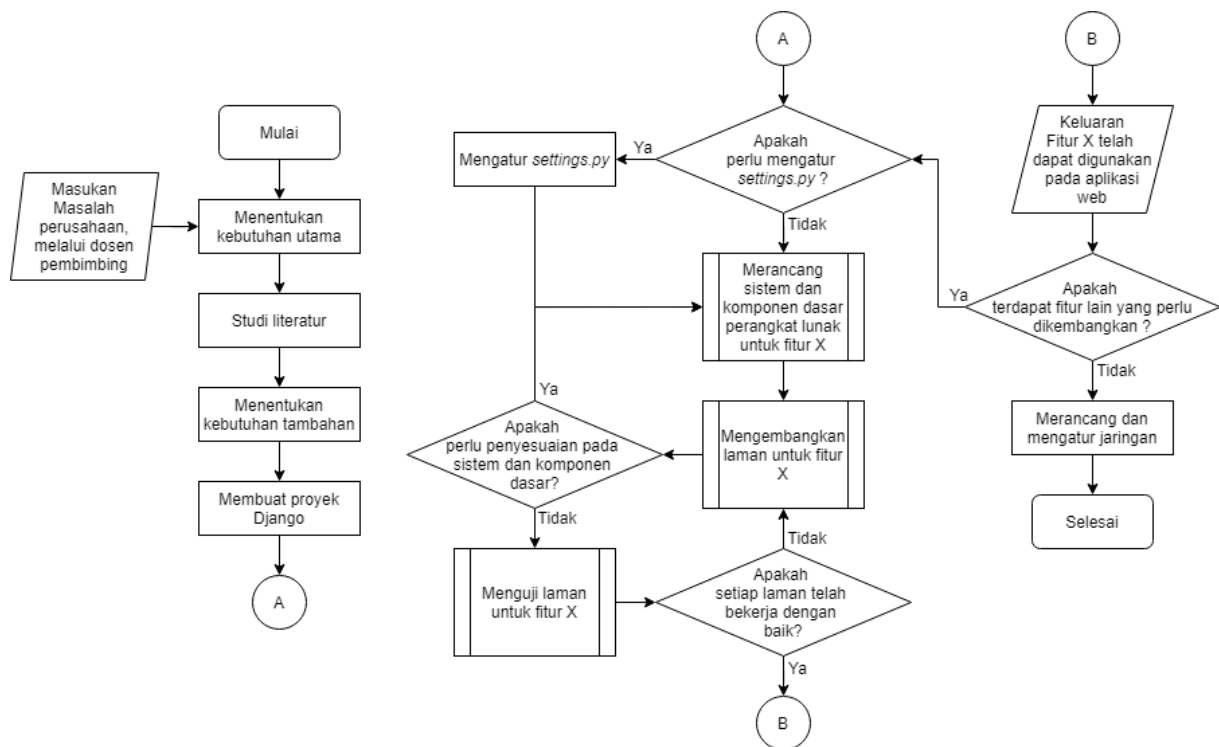
Gambar 2.12 Notasi Kaki Gagak

Bab 3

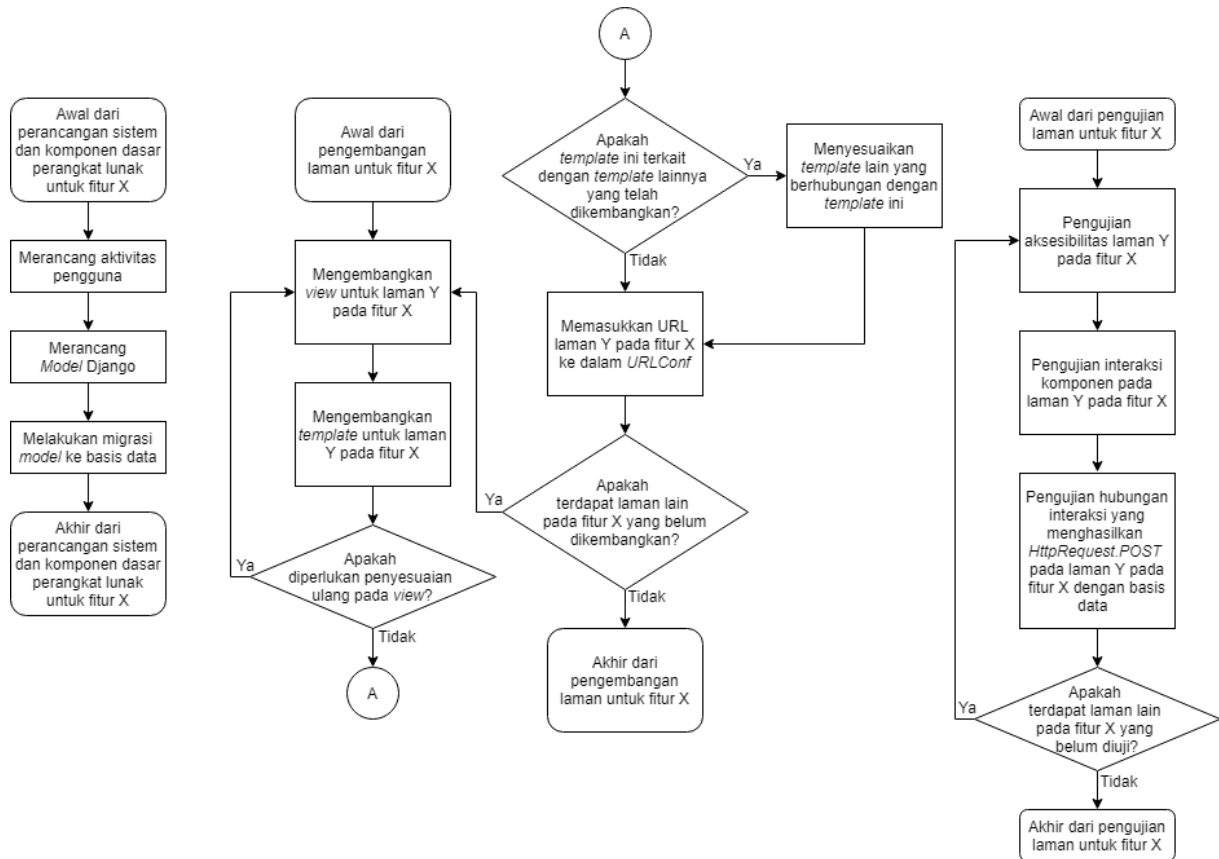
Pengembangan Perangkat Lunak

3.1. Metodologi

Secara garis besar langkah-langkah perancangan dan pengembangan perangkat lunak pada penelitian ini digambarkan melalui diagram alir pada gambar 3.1. Adapun detail dari langkah-langkah pada proses yang telah ditentukan sebelumnya (*predefined process*) yang digambarkan dengan persegi panjang dengan dua garis vertikal akan digambarkan melalui diagram alir pada gambar 3.2.



Gambar 3.1 Diagram alir metodologi penelitian tugas akhir



Gambar 3.2 Diagram alir *predefined processes* pada metodologi tugas akhir

3.2. Analisis Kebutuhan

Tujuan utama dari pembuatan sistem ini adalah membuat platform komunikasi yang mengakomodasi seluruh pihak terkait untuk melapor dan merespons laporan fenomena dalam lingkungan produksi. Untuk menjawab tujuan ini, digunakan definisi kearsipan dari ISO 15489-1:2001 menjadi kerangka berpikir. Sehingga didapat kebutuhan sistem ini sebagai berikut :

1. Dapat membuat laporan
2. Dapat memberi pengakuan dan pengesahan pada laporan
3. Dapat memelihara laporan dengan baik
4. Memudahkan penggunaan laporan
5. Dapat melenyapkan laporan
6. Dapat menampung pengumpulan bukti dari laporan
7. Dapat menyimpan bukti dari laporan

Sistem yang baik adalah sistem yang dijalankan oleh pihak-pihak terkait. Maka dari itu serupa dari semangat *Jishu Hozen* yang menekankan pada inisiatif dan rasa kepemilikan (*ownership*) dari tanggung jawab untuk terus memajukan perusahaan melalui *Kaizen*, kita harus

bisa memfasilitasi agar semua pihak terkait dapat berpartisipasi sesuai dengan peran masing-masing. Namun sangatlah naif bila hanya bergantung pada inisiatif dan rasa kepemilikan. Maka dari itu diterapkan skema gamifikasi (*gamification*) untuk mempengaruhi faktor yang menggerakkan perilaku pihak tersebut. Pada penerapan skema gamifikasi tersebut pihak terkait diberi penguatan (*reinforcements*). Maka dari itu ditambahkan kebutuhan dari sistem sebagai berikut :

8. Dapat diakses dengan mudah oleh seluruh pihak terkait
9. Mengakomodasi pemberian penguatan pada skema gamifikasi

Laporan yang telah dilaporkan, selanjutnya akan ditindaklanjuti. Namun seperti yang kita ketahui, penindaklanjutan laporan membutuhkan manajemen penugasan yang baik. Salah satu komponen inti yang menunjang manajemen tugas yang baik adalah pengetahuan bersama akan penugasan yang terbaharui dan dinamis. Untuk menunjang manajemen tugas yang baik, digunakan juga metode kanban. Pada pengembangan awal sistem ini akan diadopsi empat dari keenam praktik inti pada metode kanban berdasarkan “Kanban Yahoo! Group” dalam mekanisme sistem. Bukan berarti dua praktik sisanya tidak diterapkan, namun kedua praktik tersebut akan dimunculkan bertahap dalam dinamika sistem. Maka dari itu ditambahkan kebutuhan dari sistem sebagai berikut :

10. Dapat memberi tahu apakah suatu tugas sudah diambil
11. Dapat memberi tahu siapa yang mengambil suatu tugas
12. Dapat memvisualisasikan status pekerjaan yang diambil menggunakan papan kanban
13. Dapat mengelola alur kerja berdasarkan siklus hidup tugas
14. Dapat mengakomodasi umpan balik dan kolaborasi pada suatu tugas

Agar mengetahui siapa pengguna yang melaporkan laporan tertentu atau pengguna yang mengambil laporan tertentu, dibutuhkan fitur akun. Dimana setiap pengguna akan mendapatkan satu akun. Sistem pengakunan ini juga menjaga data dan laporan perusahaan agar tidak diakses pihak yang tidak berwenang. Karena berguna untuk merepresentasikan seorang pengguna, data pada suatu akun juga harus dapat diperbaharui sesuai dengan kondisi terkini dari pengguna. Maka dari itu ditambahkan kebutuhan dari sistem sebagai berikut:

15. Terdapat laman masuk akun (*log in page*)
16. Terdapat laman keluar akun (*log out page*)
17. Dapat membuat akun baru
18. Dapat mengatur ulang kata kunci bila lupa
19. Dapat mengatur ulang data pengguna pada akun

Sistem ini akan memiliki beberapa fitur yang akan dirancang berdasarkan kebutuhan di atas. Namun setiap fitur tidak dirancang untuk bisa diakses oleh seluruh pihak, bahkan walaupun pihak tersebut memiliki akun. Ada beberapa fitur yang hanya bisa diakses departemen tertentu atau dapat diakses setelah mencapai tingkatan tertentu di perusahaan. Karena itu dibutuhkan fitur yang bisa mengelola otorisasi suatu akun pada sistem ini. Selain itu dibutuhkan juga fitur yang bisa menggambarkan struktur organisasi perusahaan beserta anggotanya untuk menunjang fitur otorisasi dan fitur lainnya. Maka dari itu ditambahkan kebutuhan dari sistem sebagai berikut:

20. Dapat mengelola otorisasi suatu akun
21. Dapat menggambarkan struktur perusahaan beserta anggotanya

Beberapa pemangku kepentingan (*stakeholder*) dan pengambil keputusan tidak terlalu tertarik dan tidak memiliki cukup waktu untuk berhadapan dengan data yang detail. Yang mereka butuhkan adalah data mengenai gambaran besar yang dapat diperoleh melalui metode visualisasi dan statistik sehingga mudah untuk dibaca dan digunakan. Dan sebagai platform komunikasi, terdapat peluang untuk mengumumkan pengumuman perusahaan atau informasi lainnya melalui sistem ini. Maka dari itu ditambahkan kebutuhan dari sistem sebagai berikut:

22. Dapat menampilkan data gambaran besar
23. Dapat menyalurkan informasi dari perusahaan ke pengguna

3.3. Perancangan Sistem

Secara garis besar sistem akan berupa sistem kearsipan yang didorong dengan skema gamifikasi untuk menunjang kegiatan produksi yang berbasis *lean* manufacturing. Selanjutnya sistem ini akan dirancang berdasarkan dari kebutuhan yang telah dikumpulkan dari analisis pada sub bab 3.2.

Untuk menjawab kebutuhan nomor 3 mengenai pemeliharaan laporan dan kebutuhan nomor 7 mengenai penyimpanan bukti dari laporan, akan dipilih sistem digital. Dengan membuat sistem menjadi sistem digital, pemeliharaan laporan dan data terkait akan menjadi lebih aman karena dapat dicadangkan dengan mudah di tempat yang aman seperti di penyimpanan awan (*cloud storage*). Selain itu dengan tidak menggunakan media penyimpanan fisik seperti kertas, pemeliharaan laporan dan data terkait menjadi lebih mudah dan murah.

Untuk menjawab kebutuhan nomor 8 mengenai kemudahan dalam mengakses sistem, dipilih sistem digital berbasis aplikasi web. Bila dibandingkan dengan situs web, sudah jelas situs web kurang interaktif dan tidak terpersonalisasi untuk setiap akun dan pengguna

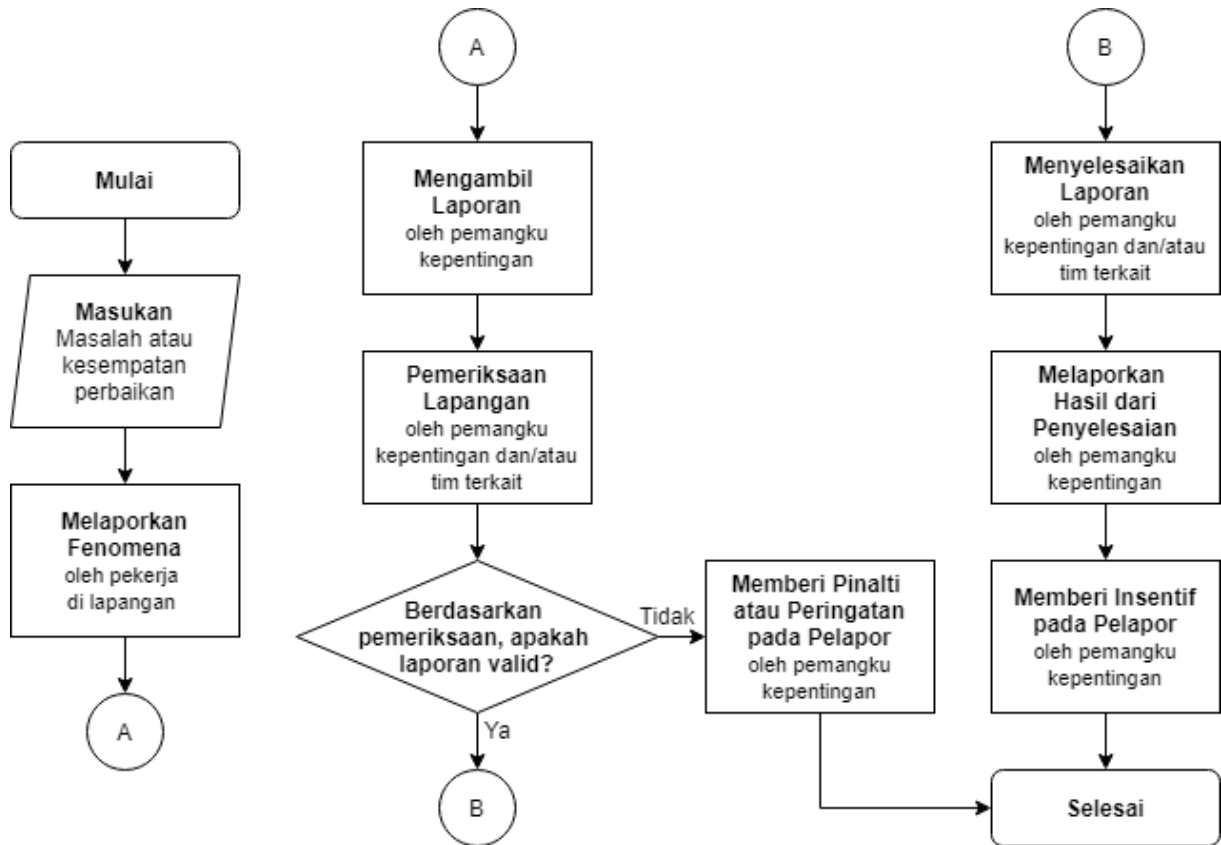
dibandingkan dengan aplikasi web. Bila dibandingkan dengan aplikasi *desktop* (*desktop application*) ataupun aplikasi seluler (*mobile application*), dalam beberapa kasus aplikasi web menjadi kurang terpersonalisasi dengan perangkat pengguna sehingga performa antarmuka aplikasi tersebut menjadi tidak maksimal. Namun untuk tujuan pengembangan awal, digunakan aplikasi web karena dapat dikembangkan dengan lebih mudah, cepat, dan murah tanpa memiliki kekurangan fitur yang esensial. Selain itu aplikasi web dapat digunakan pada komputer maupun ponsel pintar dengan dengan sistem operasi apapun, asalkan perangkat tersebut memiliki peramban web (*web browser*) yang memadai. Aplikasi web juga tidak perlu dipasang (*install*) pada perangkat. Berbeda dengan aplikasi *desktop* maupun aplikasi seluler yang harus dibuat versi yang berbeda untuk setiap sistem operasi dan aplikasi tersebut harus dipasang pada perangkat. Sebagian kebutuhan nomor 8 telah dijawab dengan merancang sistem sebagai aplikasi web, selanjutnya akan dirancang jaringan yang dapat mendukung aplikasi web dalam memenuhi kebutuhan nomor 8.

Adapun untuk mengembangkan aplikasi web, penulis memilih kerangka kerja Django. Mengingat bahasa pemrograman yang paling penulis kuasai adalah python. Django juga merupakan salah satu kerangka kerja pengembangan web berbasis python yang paling populer di samping Flask, bila bukan yang paling populer [39] seperti menurut JetBrains, penyedia perangkat lunak pendukung pengembangan perangkat lunak (penulis menggunakan IDE dari JetBrains, PyCharm). Kerangka kerja pengembangan MTV yang mudah diterapkan, pengaturan (*settings*), berkas (*file*), dan fitur kelengkapan lain yang diciptakan secara otomatis sehingga pengembang fokus pada fungsi, ORM (*object-relational mapper*) yang memudahkan untuk berinteraksi dengan basis data, serta fitur lainnya menjadi dasar kepopuleran Django. Adapun dengan populernya kerangka kerja Django, ekosistem pengembang (*developer*) Django di dunia maya juga menjadi sangat luas dan kuat. Dengan ekosistem yang luas dan kuat, akan lebih banyak *package* yang dikembangkan secara mandiri sehingga lebih banyak pilihan dalam pengembangan web dalam kerangka kerja Django. Materi tutorial serta tanya-jawab mengenai Django juga berlimpah ruah di dunia maya, sehingga Django dapat dipelajari dengan lebih mudah.

3.3.1. Perancangan Aktivitas

Aktivitas utama pada sistem ini adalah melaporkan dan merespons fenomena mengenai masalah maupun kesempatan perbaikan pada lingkungan produksi. Kegiatan pelaporan serupa dengan kegiatan kearsipan, sedangkan perseponan serupa dengan kegiatan penugasan. Maka

dari itu akan dirancang alur kerja (*workflow*) yang diadaptasi dari praktik kearsipan dan praktik manajemen tugas. Maka dari itu didapat alur kerja utama seperti yang digambarkan diagram alir pada gambar 3.3.



Gambar 3.3 Alur kerja utama pada sistem

Alur kerja utama ini terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Melaporkan fenomena, aktivitas ini berupa tindakan membuat laporan pada sistem dan mengunggah laporan tersebut beserta bukti gambar terkait pada sistem. Aktivitas ini dilakukan oleh pekerja dan pihak pihak yang berada di rantai produksi dan menyadari adanya kesalahan maupun kesempatan perbaikan.
2. Mengambil laporan, aktivitas ini berupa tindakan mengambil tanggung jawab dari laporan yang telah dilaporkan pada sistem. Aktivitas ini dilakukan pemangku kepentingan yang biasanya memiliki wewenang lebih seperti pihak dari kantor *Kaizen*, manajer divisi perbaikan, manajer divisi terkait laporan, manajer proyek, atau pihak lainnya yang berwenang untuk menangani masalah.
3. Pemeriksaan lapangan, aktivitas ini akan memeriksa kondisi objek laporan dan mengumpulkan bukti terkait laporan guna menguji validitas laporan. Aktivitas ini

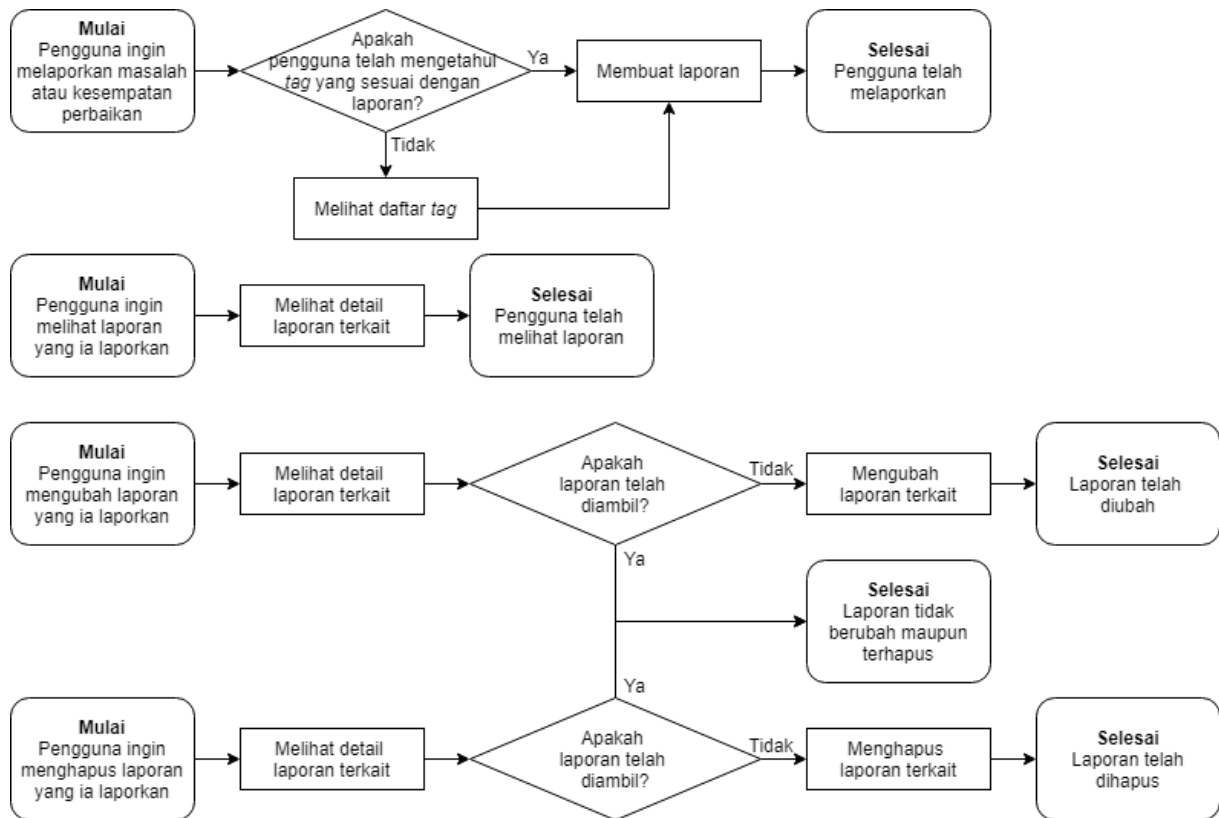
dilakukan oleh pemangku kepentingan yang mengambil laporan atau tim yang dibawah oleh pemangku kepentingan tersebut.

4. Menyelesaikan laporan, aktivitas ini dilakukan bila telah didapati bahwa laporan valid. Aktivitas ini melakukan hal yang dibutuhkan untuk menyelesaikan masalah atau memproses kesempatan perbaikan yang dilaporkan. Aktivitas ini dilakukan oleh pemangku kepentingan yang mengambil laporan atau tim yang dibawah oleh pemangku kepentingan itu. Pemangku kepentingan dapat mendelegasikan proses ini pada instansi lain, namun keberlangsungan aktivitas ini tetap di bawah tanggung jawab pemangku kepentingan tersebut.
5. Melaporkan hasil dari penyelesaian, aktivitas ini akan melaporkan apapun hasil dari penyelesaian masalah pada sistem, baik itu terselesaikan, ditunda, ataupun gagal. Bila ada keterangan ataupun informasi tambahan, dapat dilaporkan juga pada sistem. Aktivitas ini dilakukan oleh pemangku kepentingan yang mengambil laporan tersebut.
6. Memberi insentif kepada pelapor, aktivitas ini akan memberi penguatan (*reinforcement*) berupa insentif kepada pelapor. Aktivitas ini dilakukan oleh pemangku kepentingan yang mengambil laporan tersebut.
7. Memberi penalti atau peringatan pada pelapor, aktivitas ini akan memberi penalti maupun peringatan baik melalui sistem atau di luar sistem kepada pelapor bila didapati bahwa laporan yang dilaporkan tidak valid. Aktivitas ini dilakukan oleh pemangku kepentingan yang mengambil laporan tersebut.

Guna menunjang aktivitas-aktivitas alur kerja utama, sistem juga harus memiliki aktivitas lainnya yang dapat dikelompokkan dalam beberapa fitur. Fitur 1 adalah “Membuat Laporan” (“*Create a Report*”). Fitur ini akan mendukung aktivitas nomor 1 pada alur kerja utama mengenai pelaporan fenomena. Fitur ini akan memfasilitasi empat kasus, yaitu saat pengguna ingin melaporkan laporan, saat pengguna ingin melihat laporan yang ia laporkan, saat pengguna ingin mengubah laporan yang ia laporkan, dan saat pengguna ingin menghapus laporan yang telah ia laporkan. Aktivitas pada fitur 1, “Membuat Laporan” (“*Create a Report*”) dapat digambarkan dalam diagram alir sebagai pada gambar 3.4.

Fitur 1 terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Melihat daftar *tag*, aktivitas ini dilakukan pengguna bila tidak mengetahui *tag* yang sesuai dengan laporan. *Tag* sendiri adalah suatu label mengenai topik tertentu. Laporan akan ditandai dengan *tag* yang sesuai dengan topik laporan tersebut.



Gambar 3.4 Aktivitas pada fitur 1, “Membuat Laporan” (“*Create a Report*”)

2. Membuat laporan, aktivitas ini dilakukan pengguna setelah mengetahui *tag* yang sesuai. Pada aktivitas ini pengguna akan mengunggah laporan, bukti laporan, *tag* terkait, beserta informasi lainnya.
3. Melihat detail laporan terkait, aktivitas ini dilakukan untuk mengetahui detail dari laporan yang telah pengguna laporkan sebelumnya.
4. Mengubah laporan terkait, aktivitas ini dilakukan bila pengguna ingin mengubah laporan yang telah ia buat dan laporan tersebut belum diambil oleh pemangku kepentingan. Aktivitas ini akan mengubah isi laporan dan mengunggahnya kembali pada sistem. Laporan dirancang untuk tidak dapat diubah setelah diambil agar tidak terjadi kesalahan dalam penanganan baik karena disengaja maupun tidak. Pengguna harus bisa mempertanggung jawabkan laporannya bila telah diambil.
5. Menghapus laporan terkait, aktivitas ini dilakukan bila pengguna ingin mengubah laporan yang telah ia buat dan laporan tersebut belum diambil oleh pemangku kepentingan. Aktivitas ini akan menghapus laporan dari sistem. Laporan dirancang untuk tidak dapat dihapus oleh pelapor setelah diambil. Namun pengambil laporan dapat

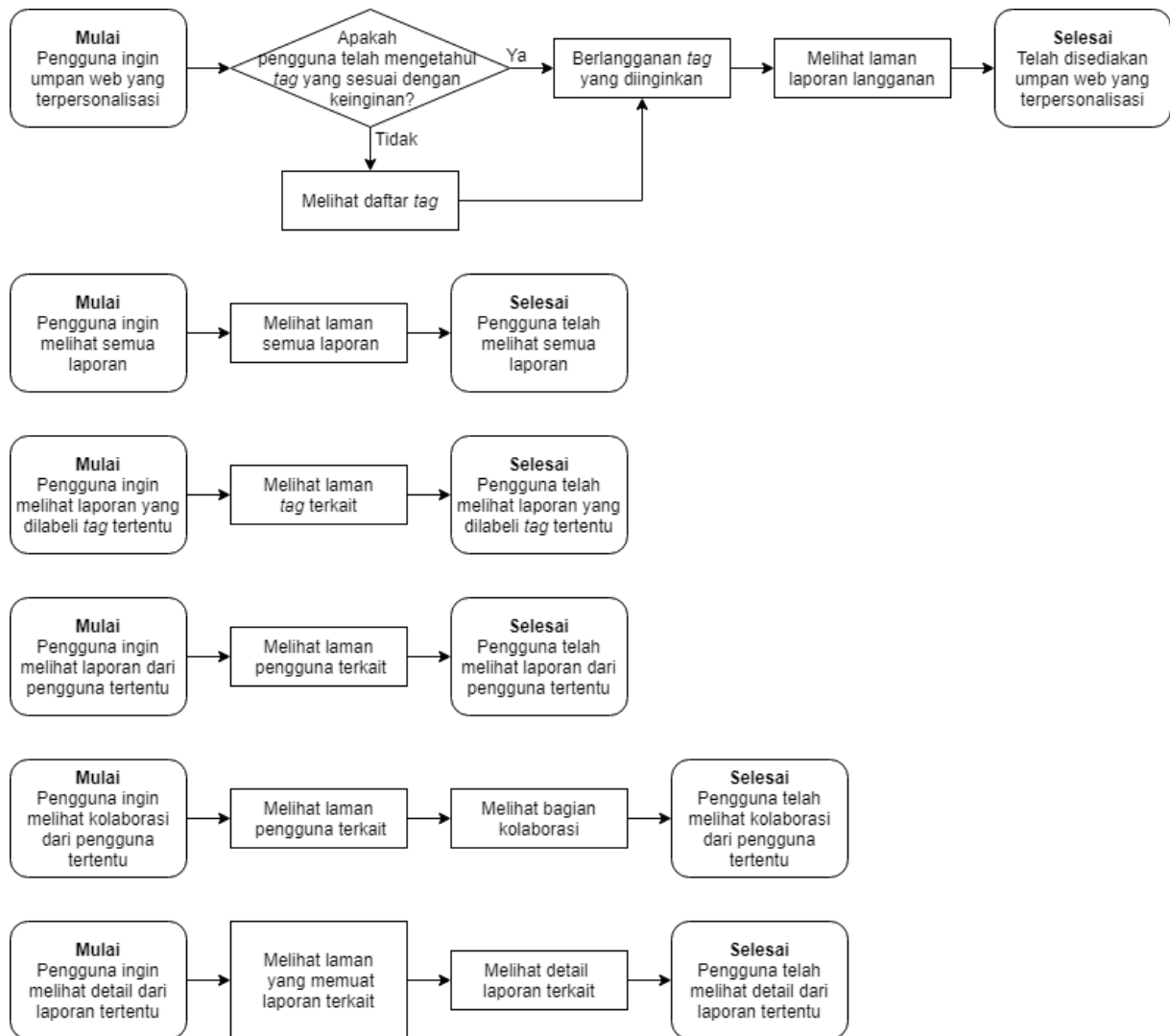
menghapus laporan karena laporan tersebut telah berada pada tanggung jawab pengambil laporan.

Aktivitas nomor 2 pada fitur 1 mengenai pembuatan laporan akan menjawab kebutuhan nomor 1 mengenai pembuatan laporan dan kebutuhan nomor 6 mengenai pengakomodasian pengumpulan bukti dari laporan. Aktivitas nomor 3 pada fitur 1 mengenai penglihatan laporan akan menjawab kebutuhan nomor 4 mengenai penggunaan laporan. Syarat aktivitas nomor 4 pada fitur 1 mengenai pembatasan perubahan laporan setelah diambil, akan memperkuat solusi yang menjawab kebutuhan nomor 2 mengenai pengesahan laporan. Aktivitas nomor 5 pada fitur 1 mengenai penghapusan laporan menjawab kebutuhan nomor 5 mengenai pelenyapan laporan.

Fitur 2 adalah “Melihat Laporan” (“*View Reports*”). Fitur ini akan mendukung aktivitas nomor 2 pada alur kerja utama mengenai pengambilan laporan. Dengan dapat melihat laporan, pengguna dapat memilih dan memutuskan laporan mana yang akan ia ambil. Selain itu fitur ini memungkinkan pengguna mengakses lagi laporan saat dibutuhkan dilain waktu. Fitur ini akan memfasilitasi enam kasus, yaitu saat pengguna ingin umpan web (*web feed*) yang terpersonalisasi sesuai dengan ketertarikan pengguna, saat pengguna ingin melihat semua laporan yang ada, saat pengguna ingin melihat laporan yang dilabeli *tag* tertentu, saat pengguna ingin melihat laporan yang diunggah pengguna tertentu, saat pengguna ingin melihat kolaborasi yang diunggah pengguna tertentu, dan saat pengguna ingin melihat detail dari laporan tertentu. Aktivitas pada fitur 2, “Melihat Laporan” (“*View Reports*”) dapat digambarkan dalam diagram alir sebagai pada gambar 3.5.

Fitur 2 terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Melihat daftar *tag*, aktivitas ini serupa dengan aktivitas nomor 1 pada fitur 1 mengenai penglihatan daftar tag. Aktivitas ini dilakukan agar pengguna tahu topik yang tersedia.
2. Berlangganan *tag* yang diinginkan, aktivitas ini dilakukan oleh pengguna pada *tag* dengan topik yang dirasa menarik atau sesuai dengan bidang pekerjaan atau keahlian pengguna. Laporan yang dilabeli *tag* yang telah dilanggan akan masuk pada laman laporan langganan.
3. Melihat laman laporan langganan, aktivitas ini dilakukan untuk mengetahui laporan-laporan yang dilabeli *tag* yang telah dilanggan. Laman ini dirancang agar pengguna dapat fokus pada laporan dengan topik yang diinginkan pengguna. Laman ini berguna bila pengguna ingin meningkatkan produktivitas dengan meningkatkan fokus.



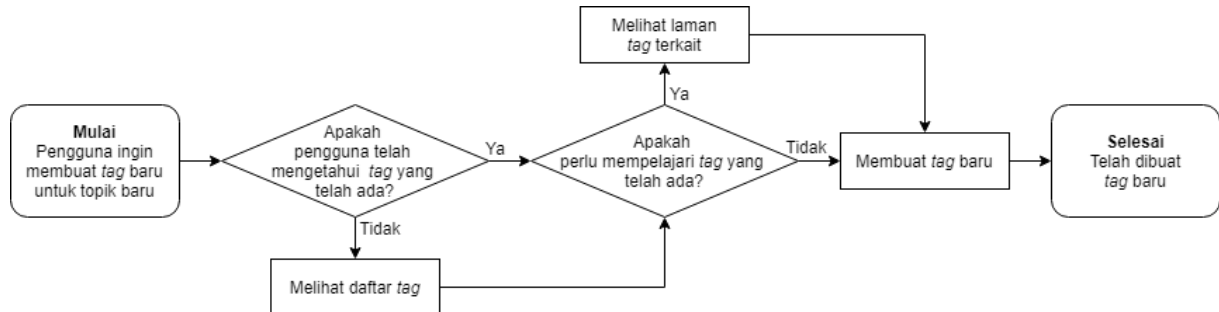
Gambar 3.5 Aktivitas pada fitur 2, “Melihat Laporan” (“View Reports”)

4. Melihat laman semua laporan, aktivitas ini dilakukan untuk mengetahui semua laporan yang diunggah pada sistem. Laman ini dirancang agar pengguna dapat mencari laporan lain yang tidak masuk dalam topik yang ia langgan. Laman ini berguna bila pengguna ingin mengurangi bias persepsi akibat fenomena gelembung tersaring (*filter bubble*).
5. Melihat laman *tag* terkait, aktivitas ini dilakukan untuk mengetahui semua laporan yang dilabeli suatu *tag*. Laman ini dirancang agar pengguna dapat fokus hanya pada satu topik. Laman ini berguna bila pengguna ingin mempelajari satu topik tertentu.
6. Melihat laman pengguna terkait, aktivitas ini dilakukan untuk mengetahui semua laporan yang diunggah oleh seorang pengguna. Laman ini berguna bila pengguna ingin memantau aktivitas pengguna terkait.

7. Melihat (laman pengguna terkait) bagian kolaborasi, aktivitas ini dilakukan untuk mengetahui semua kolaborasi yang diunggah oleh seorang pengguna. Laman ini berguna bila pengguna ingin memantau aktivitas pengguna terkait.
8. Melihat detail dari laporan terkait, aktivitas ini dilakukan bila pengguna ingin melihat detail dari suatu laporan beserta kolaborasi yang ditambahkan pada laporan tersebut. Aktivitas ini dapat diakses setelah melihat laman yang memuat laporan terkait. Yang dimaksud dengan “laman yang memuat laporan terkait” adalah laman seperti laman laporan langganan, laman semua laporan, laman *tag* terkait, laman pengguna terkait, dan laman bagian kolaborasi dari pengguna terkait.

Aktivitas nomor 3, nomor 4, nomor 5, nomor 6, nomor 7, dan nomor 8 pada fitur 2 mengenai penglihatan laporan dan detail laporan ditujukan untuk menjawab kebutuhan nomor 4 mengenai memudahkan penggunaan laporan. Setiap penglihatan laporan pada aktivitas di atas dirancang untuk digunakan pada kondisi dan tujuan yang berbeda.

Fitur 3 adalah “Membuat *Tag*” (“*Create a Tag*”). Fitur ini akan mendukung konsep pengelompokan topik dengan *tag*. Aktivitas pada fitur 3, “Membuat *Tag*” (“*Create a Tag*”) dapat digambarkan dalam diagram alir sebagai pada gambar 3.6.



Gambar 3.6 Aktivitas pada fitur 3, “Membuat *Tag*” (“*Create a Tag*”)

Fitur 3 terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Melihat daftar *tag*, aktivitas ini serupa dengan aktivitas nomor 1 pada fitur 1 mengenai penglihatan daftar tag. Aktivitas ini dilakukan agar pengguna mengetahui akan *tag* yang telah ada dan mencegah terbuat tag baru dengan topik yang sama dengan *tag* yang telah ada.
2. Melihat laman *tag* terkait, aktivitas ini serupa dengan aktivitas nomor 5 pada fitur 2 mengenai penglihatan laman *tag*. Aktivitas ini dilakukan untuk mempelajari lebih dalam *tag* yang telah ada agar tidak terbuat *tag* dengan topik yang telah ada.

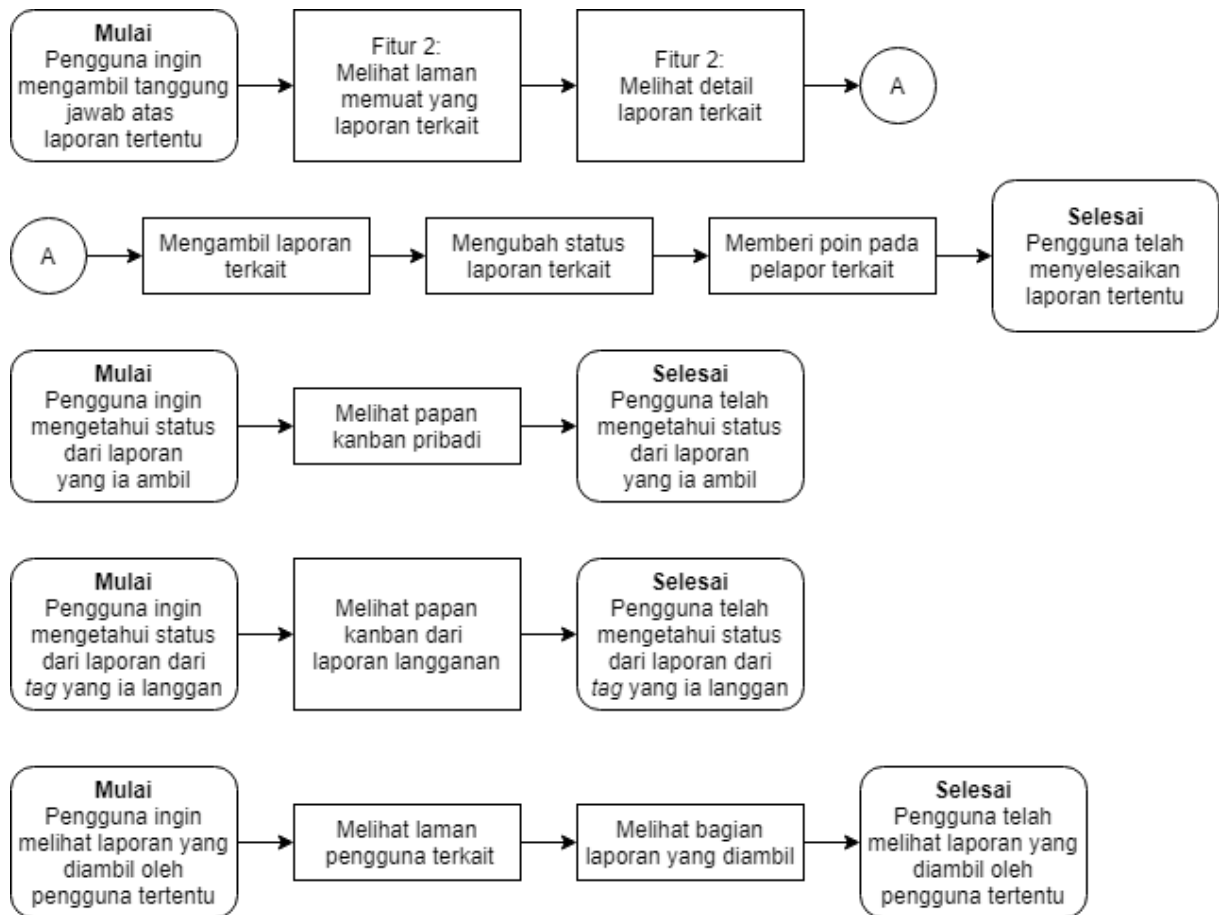
3. Membuat *tag* baru, aktivitas ini akan menghasilkan *tag* baru dengan topik yang baru. Aktivitas ini dilakukan bila terdapat laporan dengan topik yang baru dan tidak ada *tag* yang telah ada yang dapat memberi label pada topik baru tersebut.

Aktivitas nomor 3 pada fitur 3 mengenai pembuatan *tag* ditujukan untuk menunjang kebutuhan nomor 4 mengenai memudahkan penggunaan laporan. *Tag* yang dibuat akan menjadi dasar pengelompokan laporan berdasarkan topik dan memudahkan penggunaan laporan.

Fitur 4 adalah “Mengambil Laporan” (“*Take a Report*”). Fitur ini berkaitan dengan manajemen tugas dari pengambilan laporan hingga laporan selesai ditangani. Fitur ini akan mendukung aktivitas nomor 2, nomor 3, nomor 4, nomor 5, nomor 6, dan nomor 7 pada alur kerja utama mengenai aktivitas pada manajemen tugas. Fitur ini akan memfasilitasi empat kasus, yaitu saat pengguna ingin mengambil tanggung jawab atas laporan tertentu, saat pengguna ingin mengetahui status dari laporan yang ia ambil, saat pengguna ingin mengetahui status dari laporan dari *tag* yang ia langgan, dan saat pengguna ingin melihat laporan yang diambil oleh pengguna tertentu. Fitur ini sangat terkait dengan Fitur 2 “Melihat Laporan” (“*View Reports*”) karena keputusan pengambilan laporan harus didukung dengan pengetahuan mengenai laporan yang ada. Tanpa akses pada fitur 2, fitur 4 tidak dapat dijalankan dengan maksimal. Aktivitas pada fitur 4, “Mengambil Laporan” (“*Take a Report*”) dan beberapa aktivitas dari fitur 2 yang terkait dapat digambarkan dalam diagram alir sebagai pada gambar 3.7.

Fitur 4 terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Mengambil laporan terkait, aktivitas ini dilakukan bila suatu pengguna ingin mengambil tanggung jawab untuk menyelesaikan suatu laporan. Pengambilan laporan dapat dilakukan di laman detail laporan.
2. Mengubah status laporan terkait, aktivitas ini dilakukan bila terdapat kemajuan dalam penyelesaian laporan. Status dari laporan akan berdasarkan siklus hidup laporan dan status laporan akan berupa “tugas telah diambil” (“*task taken*”), “pemeriksaan lapangan” (“*field checking*”), “diterima” (“*approved*”), “tidak diterima” (“*not approved*”), “sedang diselesaikan” (“*responding*”), “terselesaikan” (“*resolved*”), “ditunda” (“*postponed*”), atau “gagal” (“*failed*”),



Gambar 3.7 Aktivitas pada fitur 4, “Mengambil Laporan” (“Take a Report”)

3. Memberi poin pada pelapor terkait, aktivitas ini dilakukan saat laporan telah ditangani. Poin yang diberikan dapat berupa bilangan bulat positif, atau bilangan bulat negatif atau dengan kata lain pengurangan poin. Pemberian poin merupakan bentuk penguatan (*reinforcement*) dan pengurangan poin adalah bentuk hukuman pada skema gamifikasi.
4. Melihat papan kanban pribadi, aktivitas ini dilakukan agar mengetahui laporan apa saja yang telah pengguna ambil dan mengetahui status dari laporan-laporan tersebut. Hal ini dilakukan agar pengguna tahu beban tanggung jawab yang ia ambil dan bagaimana kemajuan dari laporan yang ia tangani.
5. Melihat papan kanban dari laporan langgan, aktivitas ini dilakukan agar mengetahui laporan apa saja yang berhubungan dengan topik langgan pengguna dan mengetahui status dari laporan-laporan tersebut. Hal ini dilakukan agar pengguna tahu beban tanggung jawab rekan kerjanya dan bagaimana kemajuan dari laporan yang rekan kerjanya tangani.

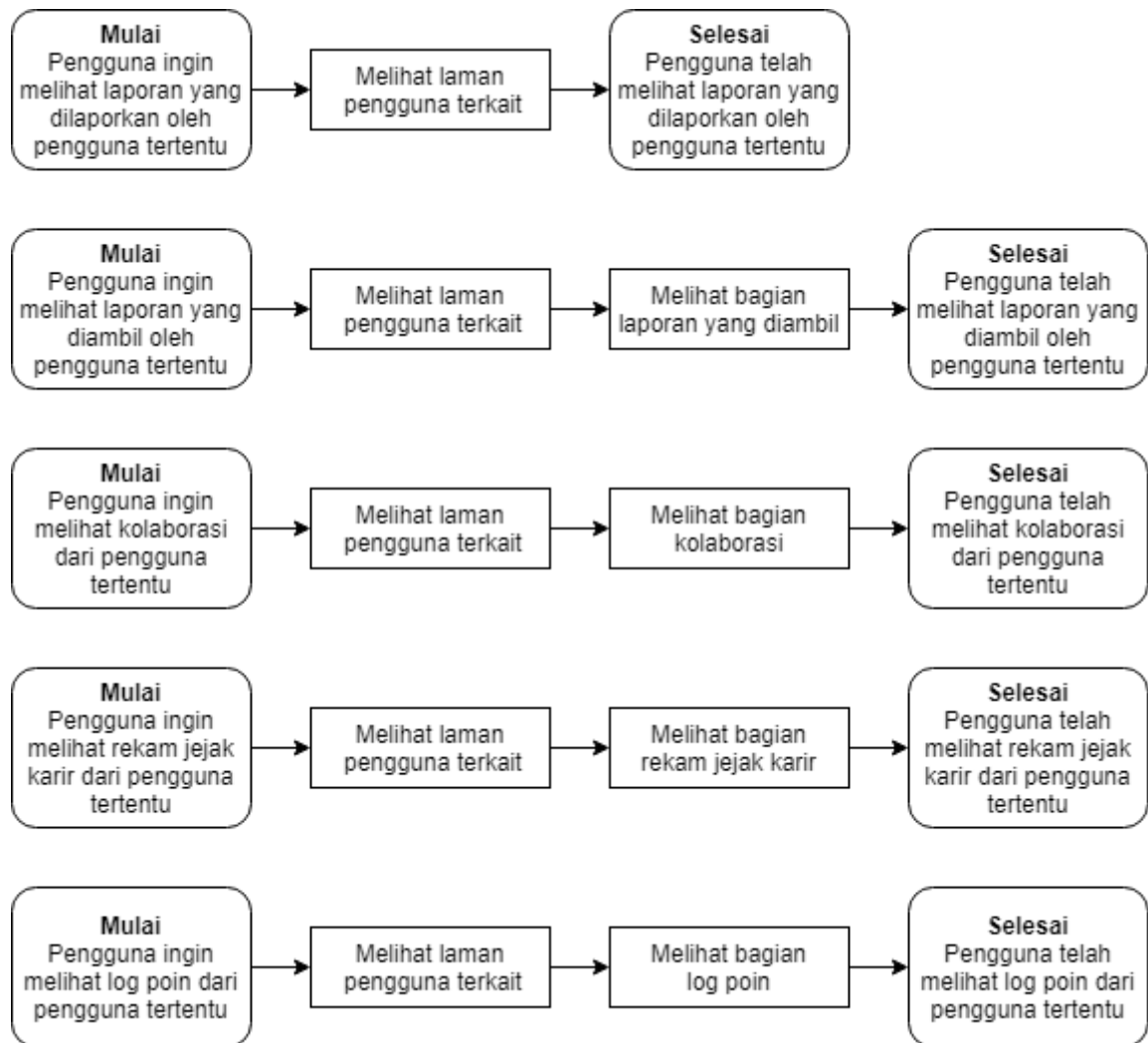
6. Melihat (laman pengguna terkait) bagian laporan yang diambil, aktivitas ini dilakukan untuk mengetahui semua laporan yang diambil oleh seorang pengguna. Laman ini berguna bila pengguna ingin memantau aktivitas pengguna terkait.

Aktivitas nomor 1 pada fitur 4 mengenai pengambilan laporan dan aktivitas nomor 6 pada fitur 4 mengenai penglihatan laporan yang diambil ditujukan untuk menjawab kebutuhan nomor 10 dan nomor 11 mengenai pengetahuan bersama akan status pengambilan suatu laporan. Aktivitas nomor 2 pada fitur 4 mengenai pembaruan status laporan ditujukan untuk menjawab kebutuhan nomor 2 mengenai pengesahan laporan dan kebutuhan nomor 13 mengenai pengelolaan alur kerja. Aktivitas nomor 3 pada fitur 4 mengenai pemberian poin ditujukan untuk menjawab kebutuhan nomor 9 mengenai skema gamifikasi. Aktivitas nomor 4 dan nomor 5 pada fitur 4 mengenai penglihatan papan kanban ditujukan untuk menjawab kebutuhan nomor 12 mengenai visualisasi status kerja.

Fitur 5 adalah “Melihat Aktivitas Pengguna” (*“View User Activities”*). Fitur ini berkaitan dengan pemantauan rekam jejak pengguna pengguna. Data mengenai rekam jejak pengguna berguna untuk mempelajari dan menilai suatu pengguna. Pengetahuan dan penilaian kepada suatu pengguna akan berguna untuk menjadi pertimbangan dalam menggunakan fitur 6 mengenai keorganisasian, fitur 7 mengenai otorisasi, fitur 8 mengenai poin, dan melacak adanya kecurangan (*fraud*) pada sistem. Fitur ini akan memfasilitasi lima kasus, yaitu saat pengguna ingin melihat laporan yang dilaporkan oleh pengguna tertentu, saat pengguna ingin melihat laporan yang diambil oleh pengguna tertentu, saat pengguna ingin melihat kolaborasi dari pengguna tertentu, saat pengguna ingin melihat rekam jejak karir dari pengguna tertentu, dan saat pengguna ingin melihat log poin dari pengguna tertentu. Aktivitas pada fitur 5, “Melihat Aktivitas Pengguna” (*“View User Activities”*) dapat digambarkan dalam diagram alir sebagai pada gambar 3.8.

Fitur 5 terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Melihat laman pengguna terkait, aktivitas ini serupa dengan aktivitas nomor 6 pada fitur 2 mengenai penglihatan laman pengguna.
2. Melihat (laman pengguna terkait) bagian laporan yang diambil, aktivitas ini serupa dengan aktivitas nomor 6 pada fitur 4 mengenai penglihatan laman pengguna bagian laporan yang diambil.
3. Melihat (laman pengguna terkait) bagian kolaborasi, aktivitas ini serupa dengan aktivitas nomor 7 pada fitur 2 mengenai laman pengguna bagian kolaborasi.



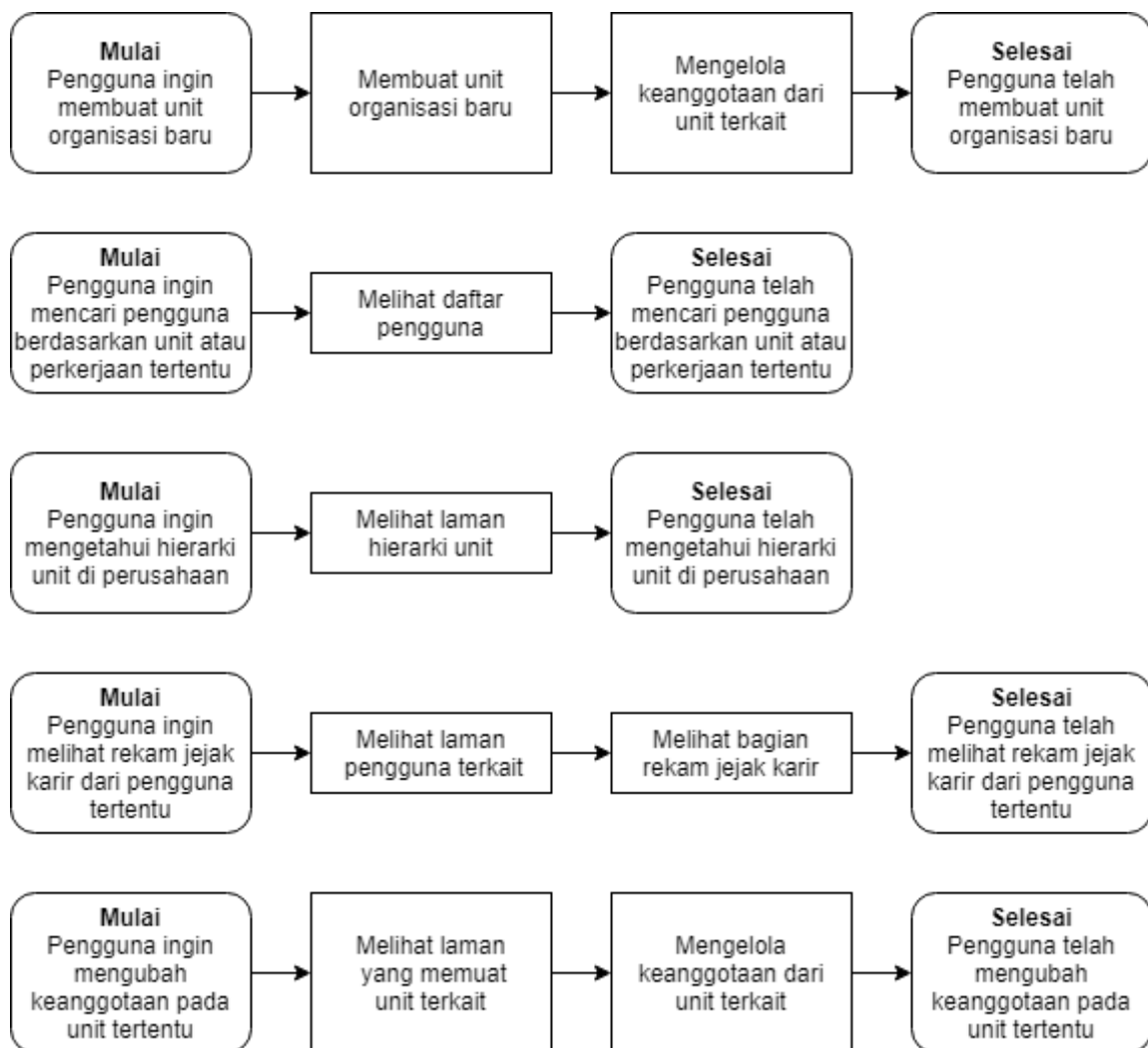
Gambar 3.8 Aktivitas pada fitur 5, “Melihat Aktivitas Pengguna” (“View User Activities”)

4. Melihat (laman pengguna terkait) bagian rekam jejak karir, aktivitas ini dilakukan untuk mengetahui semua catatan mengenai posisi dan unit seorang pengguna selama bekerja di perusahaan yang tercatat pada sistem. Aktivitas ini berhubungan dengan Fitur 6 mengenai keorganisasian dan berguna untuk departemen personalia (*human resource*) dan pihak yang mengurus otorisasi dari sistem ini.
5. Melihat (laman pengguna terkait) bagian log poin, aktivitas ini dilakukan untuk mengetahui semua catatan mengenai bertambah dan berkurangnya poin seorang pengguna yang tercatat pada sistem. Aktivitas ini berhubungan dengan Fitur 8 mengenai poin dan berguna untuk departemen keuangan atau pihak yang mengurus realisasi poin menjadi penguatan (*reinforcement*) di dunia nyata.

Kelima aktivitas pada fitur 5 berfokus pada pemantauan aktivitas pengguna tertentu. Data dan penilaian dari pemantauan akan mendukung kebutuhan nomor 11 mengenai

pengambilan tugas, kebutuhan nomor 20 mengenai otorisasi, dan kebutuhan nomor 21 mengenai keorganisasian.

Fitur 6 adalah “Mengelola Organisasi” (*“Manage Organization”*). Fitur ini berkaitan dengan unit-unit yang menyusun suatu organisasi dan anggota di dalam unit tersebut. Oleh karena itu, fitur ini sangat lekat dengan unit yang mengurus personalia (*human resource*). Selain itu, fitur ini juga akan menjadi dasar dari fitur 7 mengenai pengelolaan otorisasi pada sistem. Fitur ini akan memfasilitasi lima kasus, yaitu saat pengguna ingin membuat unit organisasi baru, saat pengguna ingin mencari pengguna berdasarkan unit atau pekerjaan tertentu, saat pengguna ingin mengetahui hierarki unit di perusahaan, saat pengguna ingin melihat rekam jejak karir dari pengguna tertentu, dan saat pengguna ingin mengubah keanggotaan pada unit tertentu. Aktivitas pada fitur 6, “Mengelola Organisasi” (*“Manage Organization”*) dapat digambarkan dalam diagram alir sebagai pada gambar 3.9.



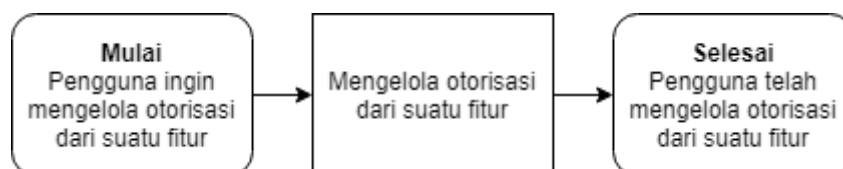
Gambar 3.9 Aktivitas pada fitur 6, “Mengelola Organisasi” (*“Manage Organization”*)

Fitur 6 terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Membuat unit organisasi baru, aktivitas ini dilakukan pengguna untuk menambah unit baru pada suatu organisasi. Unit ini dapat dikaitkan dengan unit atasannya atau unit bawahannya agar bisa membentuk suatu struktur organisasi.
2. Mengelola keanggotaan dari unit terkait, aktivitas ini dilakukan saat membuat unit baru atau bila ada perubahan keanggotaan dari unit terkait. Pada aktivitas ini ditentukan kepala unit dan anggota unit. Kepala unit dapat diganti, anggota unit dapat ditambahkan maupun dikeluarkan, dan pekerjaan anggota unit juga dapat diubah. Aktivitas ini dapat diakses setelah melihat laman yang memuat unit terkait. Yang dimaksud dengan “laman yang memuat unit terkait” adalah laman seperti laman daftar pengguna, laman hierarki unit, dan laman pengguna terkait bagian rekam jejak karir.
3. Melihat daftar pengguna, aktivitas ini dilakukan bila pengguna ingin mencari pengguna dengan kriteria keorganisasian tertentu. Pengguna dapat dicari berdasarkan nama pengguna (*username*), unit pengguna sekarang, maupun pekerjaan yang pernah dikerjakan pengguna tersebut.
4. Melihat laman hierarki unit, aktivitas ini dilakukan untuk melihat struktur unit pada organisasi. Pengguna dapat mengetahui unit mana yang menjadi bawahan suatu unit.
5. Melihat (laman pengguna terkait) bagian rekam jejak karir, aktivitas ini serupa dengan aktivitas 4 pada fitur 5 mengenai penglihatan bagian karir.

Kelima aktivitas pada fitur 6 berfokus pada pengelolaan dan penyediaan data terkait keorganisasian. Kelima aktivitas ini akan menjawab kebutuhan nomor 21 mengenai keorganisasian.

Fitur 7 adalah “Mengelola Otorisasi” (*“Manage Authorization”*). Fitur ini berkaitan dengan bisa atau tidaknya seorang pengguna untuk melakukan suatu aktivitas pada sistem. Aktivitas pada fitur 7, “Mengelola Otorisasi” (*“Manage Authorization”*) dapat digambarkan dalam diagram alir sebagai pada gambar 3.10.

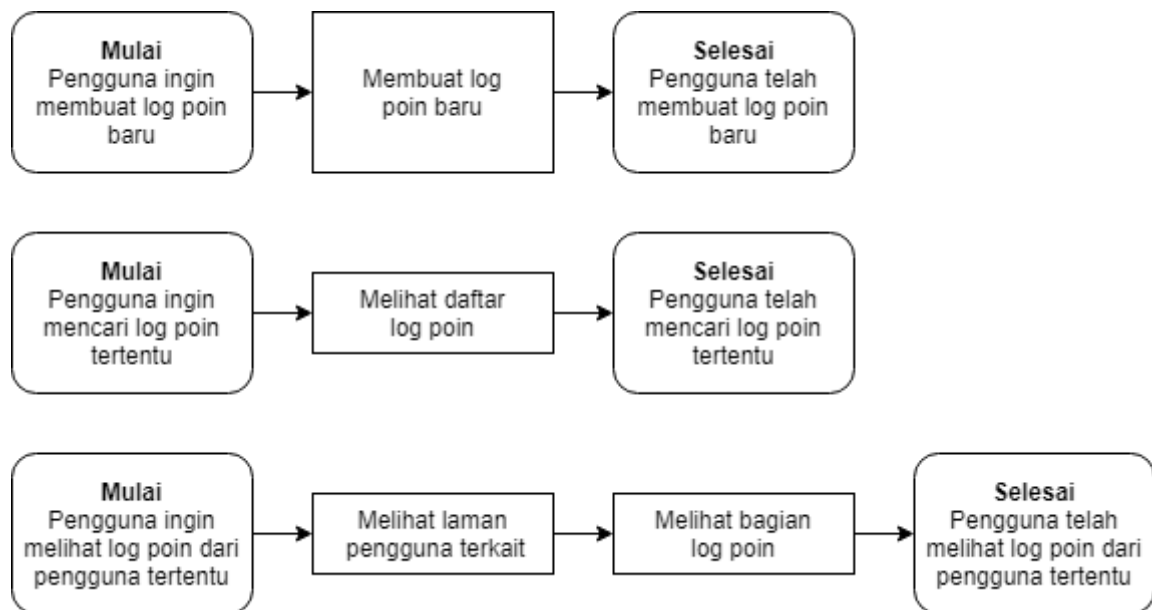


Gambar 3.10 Aktivitas pada fitur 7, “Mengelola Otorisasi” (*“Manage Authorization”*)

Fitur 7 hanya terdiri dari aktivitas yaitu mengelola otoritas dari suatu fitur. Dengan mengelola otoritas suatu fitur, pengguna dapat mengizinkan atau membatasi seorang pengguna,

anggota dari suatu unit, maupun pengguna dengan tingkat (*level*) organisasi tertentu untuk melakukan aktivitas pada fitur tersebut. Dengan memiliki otoritas pada fitur 7, seorang pengguna dapat memberi otorisasi fitur lainnya pada dirinya sendiri, sehingga pengguna tersebut dapat memiliki otoritas fitur yang lain. Aktivitas pada fitur 7 mengenai pengelolaan otorisasi ditujukan untuk menjawab kebutuhan nomor 20 mengenai pengelolaan otorisasi.

Fitur 8 adalah “Mengelola Poin” (*“Manage Point”*). Fitur ini mengenai pengelolaan poin dan akses terhadap data yang mendukung pengelolaan poin. Oleh karena itu, fitur ini sangat lekat dengan unit yang mengurus realisasi poin ke penguatan dan hukuman di dunia nyata. Bila poin direalisasikan ke dalam bentuk uang atau aset setara kas (*cash equivalent*), unit yang mengurus perihal keuangan mungkin berkepentingan untuk mengakses fitur ini. Fitur ini akan memfasilitasi tiga kasus, yaitu saat pengguna ingin membuat log poin baru, saat pengguna ingin mencari log poin tertentu, dan saat pengguna ingin melihat log poin dari pengguna tertentu. Aktivitas pada fitur 8, “Mengelola Poin” (*“Manage Point”*) dapat digambarkan dalam diagram alir sebagai pada gambar 3.11.



Gambar 3.11 Aktivitas pada fitur 8, “Mengelola Poin” (*“Manage Point”*)

Fitur 8 terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Membuat log poin baru, aktivitas ini digunakan pengguna untuk menambahkan penambahan atau pengurangan poin pada seorang pengguna. Hasil dari aktivitas ini serupa dengan aktivitas nomor 3 pada fitur 4 mengenai pemberian poin pada pelapor. Namun aktivitas ini ditujukan untuk kasus yang berbeda dengan aktivitas nomor 3 pada

fitur 4. Aktivitas nomor 3 pada fitur 4 ditujukan khusus untuk mengapresiasi atau memenalti seorang pelapor atas suatu laporan, sedangkan aktivitas ini ditujukan untuk tujuan lain selain mengapresiasi atau memenalti seorang pelapor atas suatu laporan. Contoh kasus untuk aktivitas ini adalah saat seorang pengguna menukarkan poinnya untuk insentif tertentu, sehingga perlu dicatatkan log pengurangan poin yang dimiliki pengguna terkait.

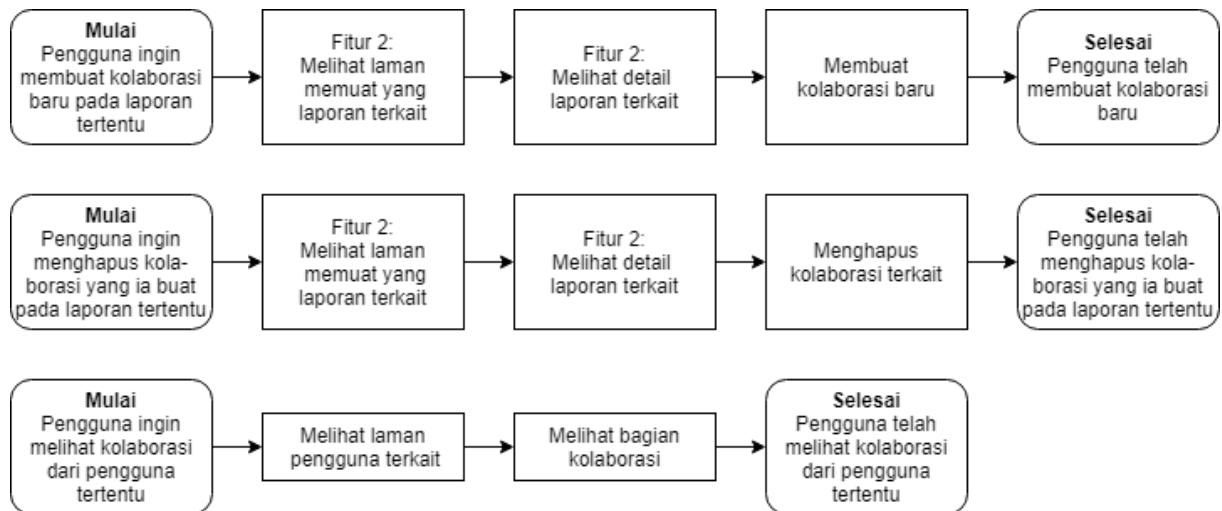
2. Melihat daftar log poin, aktivitas ini digunakan bila pengguna ingin mencari log poin dengan kriteria tertentu. Pengguna dapat mencari log poin berdasarkan tanggal pencatatan log, jumlah poin yang ditambahkan atau dikurangkan, nama pemberi poin, nama penerima poin, dan catatan terkait log poin.
3. Melihat (laman pengguna terkait) bagian log poin, aktivitas ini serupa dengan aktivitas 5 pada fitur 5 mengenai penglihatan bagian log poin.

Aktivitas nomor 1, nomor 2, dan nomor 3 pada fitur 8 mengenai pengelolaan poin dan data terkait pengelolaan poin ditujukan untuk mendukung kebutuhan nomor 9 mengenai skema gamifikasi.

Fitur 9 adalah “Berkolaborasi pada Laporan” (*“Collaborate on a Report”*). Fitur ini mengakomodasi kolaborasi pengguna untuk menambahkan informasi pada suatu laporan. Fitur ini akan memfasilitasi tiga kasus, yaitu saat pengguna ingin membuat kolaborasi baru pada laporan tertentu, saat pengguna ingin menghapus kolaborasi yang ia buat pada laporan tertentu, dan saat pengguna ingin melihat kolaborasi dari pengguna tertentu. Fitur ini sangat terkait dengan Fitur 2 “Melihat Laporan” (*“View Reports”*) karena kolaborasi hanya bisa dilakukan bila pengguna mengetahui suatu laporan. Tanpa akses pada fitur 2, fitur 9 tidak dapat dijalankan dengan maksimal. Aktivitas pada fitur 8, “Berkolaborasi pada Laporan” (*“Collaborate on a Report”*) dan beberapa aktivitas dari Fitur 2 yang terkait dapat digambarkan dalam diagram alir sebagai pada gambar 3.12.

Fitur 9 terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Membuat kolaborasi baru, aktivitas ini digunakan bila pengguna ingin menambahkan informasi terkait pada suatu laporan. Pengguna juga dapat menambahkan bukti tambahan berupa gambar pada kolom kolaborasi. Kolom kolaborasi dirancang seperti kolom komentar pada media sosial, dan terletak di bawah laporan pada laman detail laporan.
2. Melihat menghapus kolaborasi terkait, aktivitas ini digunakan bila pengguna ingin menghapus kolaborasi yang telah ia buat pada suatu laporan.



Gambar 3.12 Aktivitas pada fitur 9, “Berkolaborasi pada Laporan” (“*Collaborate on a Report*”)

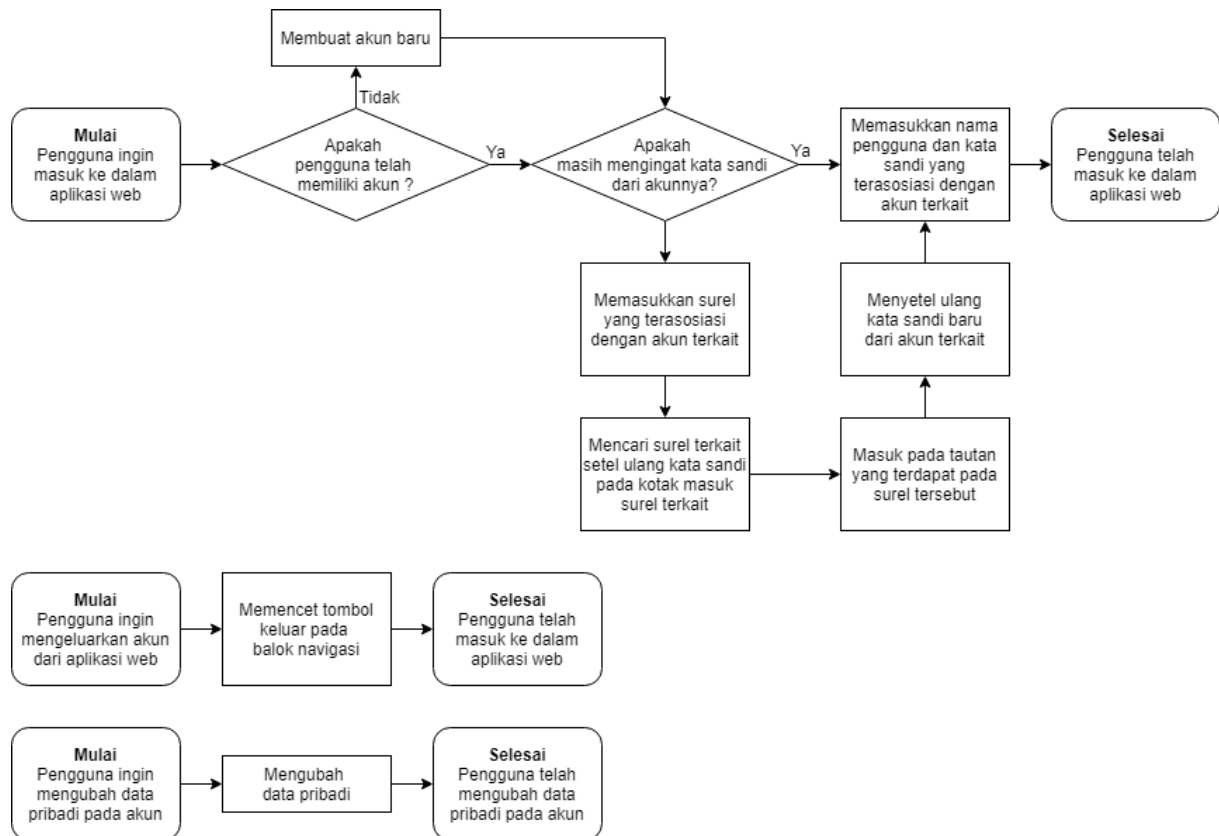
3. Melihat (laman pengguna terkait) bagian kolaborasi, aktivitas ini serupa dengan aktivitas 7 pada fitur 2 mengenai penglihatan bagian kolaborasi.

Aktivitas nomor 1, nomor 2, dan nomor 3 pada fitur 9 mengenai kolaborasi pada suatu laporan ditujukan untuk menjawab kebutuhan nomor 14 mengenai umpan balik dan kolaborasi.

Selain kesembilan fitur utama yang telah disebutkan di atas, diperlukan juga fitur dasar yang mendukung proses validasi suatu pengguna dalam menjalankan suatu aktivitas. Fitur “Akun Pengguna” akan menangani hal ini. Fitur ini akan memfasilitasi tiga kasus, yaitu saat pengguna ingin masuk ke dalam aplikasi web, saat pengguna ingin mengeluarkan akun dari aplikasi web, dan saat pengguna ingin mengubah data pribadi pada akun. Aktivitas pada fitur “Akun Pengguna” dapat digambarkan dalam diagram alir sebagai pada gambar 3.13.

Fitur “Akun Pengguna” terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Membuat akun baru, aktivitas ini digunakan bila pengguna ingin masuk ke dalam aplikasi web, namun pengguna tidak memiliki akun. Pada pembuatan akun baru, pengguna akan mendaftarkan nama pengguna (*username*), alamat surel (*email address*) terkait, dan kata sandi (*password*) untuk akun terkait.
2. Memasukkan surel yang terasosiasi dengan akun terkait, aktivitas ini dilakukan bila pengguna telah memiliki akun, namun pengguna lupa kata sandi untuk akun tersebut. Oleh karena itu pengguna akan diarahkan ke laman lupa kata sandi, dan disana pengguna diminta memasukkan alamat surel yang didaftarkan untuk akun tersebut.



Gambar 3.13 Aktivitas pada fitur “Akun Pengguna”

3. Mencari surel terkait setel ulang kata sandi pada kotak masuk surel terkait. Setelah pengajuan lupa kata sandi diajukan pada surel tertentu, akan dikirimkan pesan ke alamat surel terkait. Maka dari itu pengguna dapat mencari surel terkait lupa kata sandi untuk aplikasi web.
4. Masuk pada tautan yang terdapat pada surel tersebut, aktivitas ini dilakukan dengan memencet tautan pada surel terkait yang dikirimkan oleh sistem aplikasi web. Setelah masuk pada tautan tersebut, pengguna akan diarahkan ke laman penyetelan ulang password.
5. Menyetel ulang kata sandi baru dari akun terkait, aktivitas ini dilakukan dengan memasukkan kata sandi baru.
6. Memasukkan nama pengguna dan kata sandi yang terasosiasi dengan akun terkait, aktivitas ini dilakukan untuk masuk pada sistem aplikasi web dengan suatu akun.
7. Memencet tombol keluar pada balok navigasi, aktivitas ini dilakukan bila pengguna ingin mengeluarkan akun dari aplikasi web. Balok navigasi dapat ditemukan di bagian atas laman dari semua laman pada aplikasi web.

8. Mengubah data pribadi, aktivitas ini dilakukan dengan masuk pada laman profil, mengubah data pribadi pengguna pada kolom terkait, dan menyimpan perubahan tersebut. Pengguna dapat mengganti alamat surel, *tag* yang dilanggan, dan gambar profil (*profile picture*), namun tidak dengan nama pengguna dan kata sandi dari akun miliknya.

Aktivitas nomor 1 pada fitur “Akun Pengguna” mengenai pembuatan akun ditujukan untuk menjawab kebutuhan nomor 17 mengenai pembuatan akun. Aktivitas nomor 2, nomor 3, nomor 4, dan nomor 5 pada fitur “Akun Pengguna” mengenai penyetelan ulang kata sandi ditujukan untuk menjawab kebutuhan nomor 18 mengenai pengaturan ulang kata sandi. Aktivitas nomor 6 pada fitur “Akun Pengguna” mengenai masuk pada sistem ditujukan untuk menjawab kebutuhan nomor 15 mengenai laman masuk. Aktivitas nomor 7 pada fitur “Akun Pengguna” ditujukan untuk menjawab kebutuhan nomor 16 mengenai laman keluar. Aktivitas nomor 8 pada fitur “Akun Pengguna” ditujukan untuk menjawab kebutuhan nomor 19 mengenai pengaturan ulang data pengguna.

Dan fitur yang terakhir adalah fitur “Tambahan”. Fitur ini ditujukan untuk mendukung kebutuhan pemberian informasi tambahan. Fitur ini akan memfasilitasi tiga kasus, yaitu saat pengguna ingin mengetahui data gambaran besar sistem, saat pengguna ingin mengetahui pengumuman yang diumumkan sistem, dan saat pengguna ingin mengetahui hal mengenai aplikasi web ini. Aktivitas pada fitur “Tambahan” dapat digambarkan dalam diagram alir sebagai pada gambar 3.14.



Gambar 3.14 Aktivitas pada fitur “Tambahan”

Fitur “Tambahan” terdiri dari beberapa aktivitas yang akan dijelaskan sebagai berikut:

1. Melihat laman dasbor, aktivitas ini dilakukan saat pengguna ingin melihat data gambaran besar sistem. Pada laman ini akan disajikan data statistik mengenai laporan dan status laporan tersebut pada siklus tugas. Bila dibutuhkan data lainnya, pengguna dapat meminta kepada pengelola sistem untuk menyajikan data lainnya.
2. Melihat batang samping di laman tersebut, aktivitas ini dilakukan bila pengguna ingin melihat pengumuman dari perusahaan. Batang samping (*sidebar*) akan terlihat di bagian kanan laman apa saja, asalkan pengguna telah masuk (*log in*) kedalam sistem. Pada perangkat ponsel, batang samping akan terletak di bawah laman.
3. Melihat laman tentang kami, aktivitas ini dilakukan bila pengguna ingin tahu lebih banyak tentang pengelola atau pengembang dari sistem dan aplikasi web ini.

Aktivitas nomor 1 pada fitur “Tambahan” mengenai penglihatan dasbor akun ditujukan untuk menjawab kebutuhan nomor 22 mengenai data gambaran besar. Aktivitas nomor 2 pada fitur “Tambahan” mengenai batang samping ditujukan untuk menjawab kebutuhan nomor 23 mengenai penyaluran informasi dari perusahaan. Aktivitas nomor 3 pada fitur “Tambahan” mengenai laman tentang ditujukan untuk menjawab keinginan penulis, yang juga sebagai perancang dan pengembang sistem untuk menyampaikan informasi mengenai penulis dan sistem aplikasi web.

3.3.2. Perancangan Model

Untuk merealisasikan fitur-fitur yang telah dijabarkan pada sub sub bab 3.3.1 pada aplikasi web berbasis Django, maka dibutuhkan abstraksi dari data yang menstrukturkan data dan menghubungkan dengan basis data. Pada kerangka kerja Django, akan digunakan konsep *model* untuk menggambarkan abstraksi tersebut.

Model pertama adalah “*User*”, objek dengan *model* ini menggambarkan akun dari suatu pengguna. *Model* ini menjadi dasar dari sistem aplikasi web ini, baik untuk tujuan keamanan, maupun untuk aktivitas fitur lainnya. *Model* “*Tag*” tidak dibuat oleh penulis, namun terbuat otomatis bersama dengan pembuatan proyek Django. Oleh karena itu ada beberapa bidang (*field*) yang tidak penulis gunakan dalam perancangan sistem. Bidang (*field*) pada *model* “*User*” yang digunakan akan digambarkan di tabel 3.1.

Tabel 3.1 Bidang yang digunakan pada *model "User"*

Bidang	Tipe Data
id	integer
password	varchar(128)
is_superuser	bool
username	varchar(150)
email	varchar(254)

Bidang (*field*) pada *model "User"* yang digunakan akan dijelaskan sebagai berikut:

1. id, bidang ini otomatis terbuat saat *model* dibuat. Bidang ini akan menjadi kunci primer (*primary key*) dari objek yang memiliki *model "User"*.
2. password, bidang ini berisi kata sandi dari suatu pengguna. Bidang ini memiliki tipe data `varchar(128)` yang berarti berisi string dengan panjang maksimal 128 karakter. Kata sandi yang tersimpan akan terenkripsi, sehingga bila ada yang pihak yang bertanggung jawab mengakses basis data, kata sandi tetap tidak akan diketahui.
3. is_superuser, bidang ini bertipe boolean dan menggambarkan apakah suatu user merupakan *superuser*. *Superuser* adalah pengguna yang dapat mengakses segala fitur, mulai dari fitur yang terbuat oleh kerangka kerja Django maupun fitur yang dibuat oleh penulis.
4. username, bidang ini berisi nama pengguna pada suatu akun pengguna.
5. email, bidang ini berisi alamat surel pada suatu akun pengguna.

Model kedua adalah "*Tag*", objek dengan *model* ini menggambarkan suatu topik dan digunakan untuk melabeli suatu laporan. *Model* ini menjadi dasar dari fitur 3 "Membuat *Tag*" dan berkaitan dengan fitur 1 "Membuat Laporan", fitur 2 "Melihat Laporan", dan fitur 4 "Mengambil Laporan". *Model "Tag"* memiliki beberapa bidang (*field*) yang akan digambarkan di tabel 3.2.

Tabel 3.2 Bidang pada *model "Tag"*

Bidang	Kelas Bidang	Catatan
id	IntegerField	primary key
name	CharField	unique = True
description	TextField	
creator	ForeignKey	to User model

Model “Tag” terdiri dari beberapa bidang yang akan dijelaskan sebagai berikut:

1. id, bidang ini otomatis terbuat saat *model* dibuat. Bidang ini akan menjadi kunci primer (*primary key*) dari objek yang memiliki *model “Tag”*.
2. name, bidang ini menggambarkan judul dari suatu objek dengan *model “Tag”*. Bidang ini memiliki kelas bidang *CharField* yang dibatasi dengan panjang maksimal 35 karakter. Bidang ini juga bersifat unik, yang berarti suatu objek *“Tag”* tidak bisa memiliki nama yang sama dengan objek *“Tag”* yang lain.
3. description, bidang ini berisi deskripsi mengenai suatu objek *“Tag”*. Bidang ini memiliki kelas bidang *TextField*, hal tersebut berarti tidak ada batasan karakter dalam menuliskan deskripsi suatu objek *“Tag”*.
4. creator, bidang ini menghubungkan suatu objek *“Tag”* dengan objek *“User”*. Karena kelas bidangnya *ForeignKey*, suatu objek *“User”* dapat terhubung dengan beberapa objek *“Tag”*, namun suatu objek *“Tag”* hanya terhubung dengan satu objek *“User”*. Hubungan ini menggambarkan siapa pengguna yang membuat objek *“Tag”* ini.

Model ketiga adalah *“Report”*, objek dari ini merepresentasikan suatu laporan. *Model* ini menjadi dasar dari fitur 1 “Membuat Laporan”, fitur 2 “Melihat Laporan”, dan fitur 4 “Mengambil Laporan”. *Model “Report”* memiliki beberapa bidang (*field*) yang akan digambarkan di tabel 3.3.

Tabel 3.3 Bidang pada *model “Report”*

Bidang	Kelas Bidang	Catatan
id	IntegerField	primary key
title	CharField	
content	TextField	
date_reported	DateTimeField	default = timezone.now
reporter	ForeignKey	to User model
image	ImageField	
tag	ManyToManyField	to Tag model
urgency	IntegerField	choices = Level.choices
importance	IntegerField	choices = Level.choices
progress	IntegerField	choices = TaskLevel.choices
taker	ForeignKey	to User model

Tabel 3.4 Bidang pada *model "Report"* (lanjutan)

Bidang	Kelas Bidang	Catatan
date_last_progress	DateTimeField	
progress_note	TextField	
point	IntegerField	

Model "Report" terdiri dari beberapa bidang yang akan dijelaskan sebagai berikut:

1. id, bidang ini otomatis terbuat saat *model* dibuat. Bidang ini akan menjadi kunci primer (*primary key*) dari objek yang memiliki *model "Report"*.
2. title, bidang ini menggambarkan judul dari suatu objek dengan *model "Report"*. Bidang ini memiliki kelas bidang CharField yang dibatasi dengan panjang maksimal 100 karakter.
3. content, bidang ini berisi isi mengenai suatu objek *"Report"*. Bidang ini memiliki kelas bidang TextField, hal tersebut berarti tidak ada batasan karakter dalam menuliskan isi suatu objek *"Report"*.
4. date_reported, bidang ini memuat waktu dilaporkannya suatu laporan. Bidang ini dirancang untuk mencatatkan waktu saat objek *"Report"* dibuat.
5. reporter, bidang ini menghubungkan suatu objek *"Report"* dengan objek *"User"*. Hubungan ini menggambarkan siapa melaporkan laporan ini.
6. image, bidang ini memuat alamat direktori disimpannya suatu gambar dari suatu objek *"Report"*. Gambar yang menjadi bukti suatu laporan tidak disimpan pada basis data SQL, namun disimpan pada suatu direktori.
7. tag, bidang ini menghubungkan suatu objek *"Report"* dengan objek *"Tag"*. Karena kelas bidangnya ManyToManyField, suatu objek *"Report"* dapat terhubung dengan beberapa objek *"Tag"*, begitu juga dengan suatu objek *"Tag"* dapat terhubung dengan beberapa objek *"Report"*. Hubungan ini menggambarkan topik apa yang berhubungan dengan laporan ini.
8. urgency, bidang ini menggambarkan seberapa mendesak suatu laporan untuk dikerjakan. Tingkat keterdesakan dapat dipilih pada pilihan pada kelas Level(model.IntegerChoices).
9. importance, bidang ini menggambarkan seberapa penting suatu laporan. Tingkat kepentingan dapat dipilih pada pilihan pada kelas Level(model.IntegerChoices).

Tabel 3.5 Pilihan pada kelas Level(model.IntegerChoices)

Label	Nilai
High	1
Medium	2
Low	3

10. progress, bidang ini menggambarkan status pengerjaan laporan yang telah diambil. Status pengerjaan dapat dipilih pada pilihan pada kelas TaskLevel(models.IntegerChoices). Kelas TaskLevel dirancang untuk mengikuti siklus hidup laporan yang telah dirancang seperti pada alur kerja utama.

Tabel 3.6 Pilihan pada kelas TaskLevel(model.IntegerChoices)

Label	Nilai
Task Taken	1
Field Checking	2
Approved	3
Not Approved	4
Responding	5
Resolved	6
Postponed	7
Failed	8

11. taker, bidang ini menghubungkan suatu objek “*Report*” dengan objek “*User*”. Hubungan ini menggambarkan siapa mengambil tanggung jawab untuk mengerjakan laporan ini.
12. date_last_progress, bidang ini mencatat kapan terakhir kali terdapat perubahan pada progress maupun progress_note.
13. progress_note, bidang ini berisi catatan mengenai penyelesaian suatu objek “*Report*”. Bidang ini memiliki kelas bidang TextField, hal tersebut berarti tidak ada batasan karakter dalam menuliskan progress_note.
14. Point, bidang ini memuat berapa poin yang diberikan kepada reporter.

Model keempat adalah “*Collaboration*”, objek dengan *model* ini menggambarkan suatu kolaborasi pada suatu laporan. *Model* ini menjadi dasar dari fitur 9 “Berkolaborasi pada suatu

Laporan”. Model “*Collaboration*” memiliki beberapa bidang (*field*) yang akan digambarkan di tabel 3.6.

Tabel 3.7 Bidang pada model “*Collaboration*”

Bidang	Kelas Bidang	Catatan
id	IntegerField	primary key
report	ForeignKey	to Report model
subject	CharField	
content	TextField	
date	DateTimeField	default=timezone.now
collaborator	ForeignKey	
image	ImageField	

Model “*Collaboration*” terdiri dari beberapa bidang yang akan dijelaskan sebagai berikut:

1. id, bidang ini otomatis terbuat saat *model* dibuat. Bidang ini akan menjadi kunci primer (*primary key*) dari objek yang memiliki *model* “*Collaboration*”.
2. report, bidang ini menghubungkan suatu objek “*Collaboration*” dengan objek “*Report*”. Hubungan ini menggambarkan kepada laporan yang mana kolaborasi ini ditujukan.
3. subject, bidang ini menggambarkan judul dari suatu objek dengan model “*Collaboration*”. Bidang ini memiliki kelas bidang CharField yang dibatasi dengan panjang maksimal 100 karakter.
4. content, bidang ini berisi isi mengenai suatu objek “*Collaboration*”. Bidang ini memiliki kelas bidang TextField, hal tersebut berarti tidak ada batasan karakter dalam menuliskan isi suatu objek “*Collaboration*”.
5. date, bidang ini memuat waktu diunggahnya suatu kolaborasi. Bidang ini dirancang untuk mencatatkan waktu saat objek “*Collaboration*” dibuat.
6. collaborator, bidang ini menghubungkan suatu objek “*Collaboration*” dengan objek “*User*”. Hubungan ini menggambarkan siapa yang berkolaborasi dengan kolaborasi ini.
7. image, bidang ini memuat alamat direktori disimpannya suatu gambar dari suatu objek “*Collaboration*”. Gambar yang menjadi bukti suatu kolaborasi disimpan pada direktori tertentu, sama halnya dengan gambar pada object “*Report*”.

Model kelima adalah “*Unit*”, objek dengan *model* ini merupakan manifestasi suatu unit pada suatu organisasi. Hubungan atasan dan bawahan antar-unit membentuk struktur hierarki suatu organisasi. Objek “*Unit*” yang dimaksud disini bukan berarti tingkatan unit secara harfiah (seperti tingkatan departemen atau tingkatan divisi), namun suatu tim yang berisi sekumpulan orang. Objek “*Unit*” dengan tingkatan rendah dapat merepresentasikan unit pada perusahaan, objek “*Unit*” dengan tingkatan menengah dapat merepresentasikan suatu divisi atau departemen pada perusahaan, begitu juga objek “*Unit*” dengan tingkatan tertinggi dapat merepresentasikan suatu perusahaan. *Model* ini menjadi dasar dari fitur 6 “Mengelola Organisasi” dan fitur 7 “Mengelola Otorisasi”. *Model* “*Unit*” memiliki beberapa bidang (*field*) yang akan digambarkan di tabel 3.7.

Tabel 3.8 Bidang pada *model* “*Unit*”

Bidang	Kelas Bidang	Catatan
id	IntegerField	primary key
name	CharField	
superior	ForeignKey	to Unit model
head	OneToOneField	to User model
date_created	DateTimeField	default=timezone.now
date_deactivated	DateTimeField	
status	IntegerField	choices = Status.choices

Model “*Unit*” terdiri dari beberapa bidang yang akan dijelaskan sebagai berikut:

1. id, bidang ini otomatis terbuat saat *model* dibuat. Bidang ini akan menjadi kunci primer (*primary key*) dari objek yang memiliki *model* “*Unit*”.
2. name, bidang ini menggambarkan nama dari suatu objek dengan *model* “*Unit*”. Bidang ini memiliki kelas bidang CharField yang dibatasi dengan panjang maksimal 100 karakter.
3. superior, bidang ini menghubungkan suatu objek “*Unit*” dengan objek “*Unit*” lainnya. Unit pada bidang superior ini adalah unit atasan dari objek “*Unit*” terkait. Hubungan antar-unit menggunakan bidang superior dan kelas bidang ForeignKey dirasa sesuai karena pada umumnya suatu unit hanya memiliki 1 unit atasan dan dapat memiliki lebih dari 1 unit bawahan. Terdapat kasus dimana suatu unit memiliki lebih dari satu atasan

seperti pada struktur organisasi matriks, namun penelitian ini lebih memfokuskan pada struktur organisasi yang umum digunakan.

4. head, bidang ini menghubungkan suatu objek “Unit” dengan objek “User”. Hubungan ini menggambarkan siapa yang mengepalai suatu unit. Hubungan OneToOne dianggap sesuai karena satu unit umumnya dikepalai oleh satu orang, dan satu orang umumnya mengepalai satu unit.
5. date_created, bidang ini memuat waktu dibuatnya suatu unit. Bidang ini dirancang untuk mencatatkan waktu saat objek “Unit” dibuat.
6. date_deactivated, bidang ini memuat waktu dimatikannya suatu unit bila sudah tidak berfungsi lagi.
7. status, bidang ini menggambarkan masih ada atau tidaknya suatu unit. Status kehidupan unit dapat dipilih pada pilihan pada kelas Status(model.IntegerChoices). Secara asali (*default*), status suatu unit adalah aktif. Kecuali bila suatu unit dinonaktifkan.

Tabel 3.9 Pilihan pada kelas Status(model.IntegerChoices)

Label	Nilai
Inactive	0
Active	1

Seperti kelas pada umumnya, *model* “Unit” dapat memiliki metode (*method*), fungsi yang dimiliki oleh suatu kelas. Beberapa metode yang dimiliki *Model* “Unit” akan dijelaskan sebagai berikut:

1. level(), metode ini digunakan untuk mengetahui tingkatan (*level*) dari suatu unit. Unit dengan tingkatan teratas akan memiliki tingkatan 1, dan bawahan langsungnya kan memiliki tingkatan 2, dan seterusnya. Metode level ini berguna untuk memastikan tidak adanya kesalahan penempatan unit atasan pada bidang superior. Kesalahan tersebut dapat membuat hierarki organisasi menjadi sebuah jerat (*loop*).
2. superior_list(), metode ini digunakan untuk mengetahui unit apa yang menjadi atasan dari suatu unit, unit apa yang menjadi atasan dari atasan unit terkait, dan seterusnya hingga unit dengan tingkatan tertinggi. Metode ini berguna untuk mengetahui rantai komando ke atas.
3. subordinate_list(), metode ini digunakan untuk mengetahui unit apa saja yang menjadi bawahan dari suatu unit, unit apa saja yang menjadi bawahan dari bawahan unit terkait,

dan seterusnya hingga suatu unit tidak memiliki bawahan lagi. Metode ini berguna untuk mengetahui unit dan sumber daya yang di bawah komando suatu unit.

4. `subordinate_list_recursive()`, metode ini tidak ditujukan untuk digunakan oleh pengguna, namun ditujukan sebagai bagian dari metode `subordinate_list()`. Mengingat unit bawahan dapat berjumlah lebih dari satu, dan unit bawahan dari suatu unit bawahan juga dapat berjumlah lebih dari satu, maka digunakan fungsi rekursif untuk memetakan setiap unit bawahan dan unit bawahan dari unit bawahan.

Model keenam adalah “*Profile*”, objek dengan *model* ini merupakan perpanjangan dari *model* “*Unit*”. Karena merupakan *model* bawaan, *model* “*Unit*” tidak memiliki beberapa bidang untuk kebutuhan spesifik pada aplikasi web ini. Maka dari itu *model* “*Profile*” yang akan mengakomodasi bidang yang dimaksud. *Model* ini berkaitan dengan fitur 5 “Melihat Aktivitas Pengguna” dan fitur “Akun Pengguna”. *Model* “*Profile*” memiliki beberapa bidang (*field*) yang akan digambarkan di tabel 3.9.

Tabel 3.10 Bidang pada *model* “*Profile*”

Bidang	Kelas Bidang	Catatan
id	IntegerField	primary key
user	OneToOneField	to User model
image	ImageField	
tag	ManyToManyField	to Tag model
unit	ForeignKey	to Unit model

Model “*Profile*” terdiri dari beberapa bidang yang akan dijelaskan sebagai berikut:

1. id, bidang ini otomatis terbuat saat *model* dibuat. Bidang ini akan menjadi kunci primer (*primary key*) dari objek yang memiliki *model* “*Profile*”.
2. user, bidang ini menghubungkan suatu objek “*Profile*” dengan objek “*User*”. Hubungan ini menggambarkan profil ini merupakan perpanjangan dari pengguna yang mana. Karena merupakan perpanjangan, maka satu profil untuk satu pengguna dan satu pengguna untuk satu profil, hubungan OneToOne.
3. image, bidang ini memuat alamat direktori disimpannya suatu gambar profil (*profile picture*) dari suatu objek “*Profile*”.
4. tag, bidang ini menghubungkan suatu objek “*Profile*” dengan objek “*Tag*”. Hubungan ini menggambarkan topik apa yang dilanggan suatu pengguna. Suatu *tag* dapat

dilanggan beberapa pengguna, begitu juga dengan suatu pengguna dapat berlangganan beberapa *tag*, maka dari itu digunakan hubungan ManyToMany.

5. unit, bidang ini menghubungkan suatu objek "*Profile*" dengan objek "*Unit*". Hubungan ini menggambarkan di unit mana suatu pengguna bekerja. Suatu pengguna hanya bisa bekerja pada satu unit, namun suatu unit dapat menampung beberapa pengguna, maka dari itu dipilih hubungan ManyToOne.

Seperti kelas pada umumnya, *model "Unit"* dapat memiliki metode (*method*), fungsi yang dimiliki oleh suatu kelas. Beberapa metode yang dimiliki *Model "Unit"* akan dijelaskan sebagai berikut:

1. level(), metode ini digunakan untuk mengetahui tingkatan (*level*) dari suatu pengguna. Kepala dari suatu unit akan memiliki tingkatan yang sama dengan tingkatan unit tersebut. Sedangkan anggota dari suatu unit akan memiliki tingkatan lebih satu ($x + 1$) dari tingkatan unit tersebut. Maka dari itu hanya ada satu pengguna dengan tingkatan satu, yaitu kepala dari unit tertinggi atau terkadang dapat berupa direktur utama dari perusahaan (tanpa mempertimbangkan komisaris dalam struktur organisasi).
2. point(), metode ini digunakan untuk mengetahui berapa akumulasi poin dari suatu pengguna. Metode ini mengakumulasikan semua log poin yang ditujukan untuk pengguna tersebut.

Model ketujuh adalah "*CareerHistory*", objek dengan *model* ini dapat diibaratkan sebagai log karir suatu pengguna. *Model* ini menjadi dasar dari fitur 6 "Mengelola Organisasi". *Model "CareerHistory"* memiliki beberapa bidang (*field*) yang akan digambarkan di tabel 3.10.

Tabel 3.11 Bidang pada *model "CareerHistory"*

Bidang	Kelas Bidang	Catatan
id	IntegerField	primary key
user	ForeignKey	to User model
unit	ForeignKey	to Unit model
job	CharField	
date_started	DateTimeField	
date_ended	DateTimeField	

Model "Auth" terdiri dari beberapa bidang yang akan dijelaskan sebagai berikut:

1. id, bidang ini otomatis terbuat saat *model* dibuat. Bidang ini akan menjadi kunci primer (*primary key*) dari objek yang memiliki *model* “*CareerHistory*”.
2. user, bidang ini menghubungkan suatu objek “*CareerHistory*” dengan objek “*User*”. Hubungan ini menggambarkan siapa pengguna yang digambarkan dalam log karir ini.
3. unit, bidang ini menghubungkan suatu objek “*CareerHistory*” dengan objek “*Unit*”. Hubungan ini menggambarkan unit yang mana tempat pengguna bekerja saat itu.
4. job, bidang ini menggambarkan pekerjaan apa yang dikerjakan pengguna pada saat itu.
5. date_started, bidang ini menggambarkan kapan pengguna terkait memulai pekerjaan tersebut.
6. date_ended, bidang ini menggambarkan kapan pengguna terkait mengakhiri pekerjaan tersebut.

Model kedelapan adalah “*Auth*”, objek dengan *model* ini dapat diibaratkan sebagai sebuah tiket untuk mengakses suatu fitur. *Model* ini menjadi dasar dari fitur 7 “Mengelola Otorisasi”. *Model* “*Auth*” memiliki beberapa bidang (*field*) yang akan digambarkan di tabel 3.11.

Tabel 3.12 Bidang pada *model* “*Auth*”

Bidang	Kelas Bidang	Catatan
id	IntegerField	primary key
feature	IntegerField	choices = Feature.choices
auth_user	ForeignKey	to User model
auth_unit	ForeignKey	to Unit model
auth_level	IntegerField	

Model “*Auth*” terdiri dari beberapa bidang yang akan dijelaskan sebagai berikut:

1. id, bidang ini otomatis terbuat saat *model* dibuat. Bidang ini akan menjadi kunci primer (*primary key*) dari objek yang memiliki *model* “*Auth*”.
2. feature, bidang ini menunjukkan fitur apa yang terkait dengan “tiket” otorisasi ini. Fitur yang diatur otorisasinya dapat dipilih pada pilihan pada kelas Feature(models.IntegerChoices). Pilihan fitur akan berdasarkan 9 fitur utama yang telah dirancang pada sub sub bab 3.3.1 mengenai perancangan aktivitas. Fitur “Akun Pengguna” dan fitur “Tambahkan” tidak masuk dalam pilihan karena semua pengguna yang telah masuk (*log in*) dapat mengakses fitur ini.

Tabel 3.13 Pilihan pada kelas Feature(model.IntegerChoices)

Label	Nilai
Create a Report	1
View Reports	2
Create a Tag	3
Take a Report	4
View User Activities	5
Manage Organization	6
Manage Authorization	7
Manage Point	8
Collaborate on a Report	9

3. `auth_user`, bidang ini menghubungkan suatu objek “Auth” dengan objek “User”. Hubungan ini menggambarkan pengguna mana yang diberi izin untuk mengakses fitur pada bidang feature.
4. `auth_unit`, bidang ini menghubungkan suatu objek “Auth” dengan objek “Unit”. Hubungan ini menggambarkan pengguna dari unit mana yang diberi izin untuk mengakses fitur pada bidang feature.
5. `auth_level`, bidang ini berisi angka yang menggambarkan pengguna dengan tingkatan berapa yang diberi izin untuk mengakses fitur pada bidang feature.

Model terakhir adalah “*PointHistory*”, objek dengan *model* ini ini dapat diibaratkan sebagai log poin yang diberikan atau dikurangi pada sistem aplikasi web. *Model* ini menjadi dasar dari fitur 8 “Mengelola Poin”. *Model* “*PointHistory*” memiliki beberapa bidang (*field*) yang akan digambarkan di tabel 3.13.

Tabel 3.14 Bidang pada *model* “*PointHistory*”

Bidang	Kelas Bidang	Catatan
id	IntegerField	primary key
user	ForeignKey	to User model
point	IntegerField	
note	TextField	
date	DateTimeField	default = timezone.now
writer	ForeignKey	to User model

Model “PointHistory” terdiri dari beberapa bidang yang akan dijelaskan sebagai berikut:

1. id, bidang ini otomatis terbuat saat *model* dibuat. Bidang ini akan menjadi kunci primer (*primary key*) dari objek yang memiliki *model “PointHistory”*.
2. user, bidang ini menghubungkan suatu objek *“PointHistory”* dengan objek *“User”*. Hubungan ini menggambarkan siapa pengguna yang mendapat (atau dikurangi) poin pada log poin ini.
3. point, bidang ini menggambarkan jumlah poin yang diberikan, bila berupa bilangan positif, atau yang dikurangkan, bila berupa bilangan negatif, pada seorang pengguna pada log poin ini.
4. note, bidang ini memuat keterangan mengenai log poin ini. Karena memiliki kelas bidang *TextField*, catatan log poin tidak dibatasi karakter. Poin yang berasal dari apresiasi atau pinalti dari laporan akan dicatatkan pada log poin dengan format catatan yang terstandarisasi.
5. date, bidang ini menggambarkan kapan log poin dicatatkan pada sistem.
6. writer, bidang ini menghubungkan suatu objek *“PointHistory”* dengan objek *“User”*. Hubungan ini menggambarkan siapa pengguna yang memberi (atau mengurangi) poin pada log poin ini.

3.3.3. Basis Data Hasil Perancangan

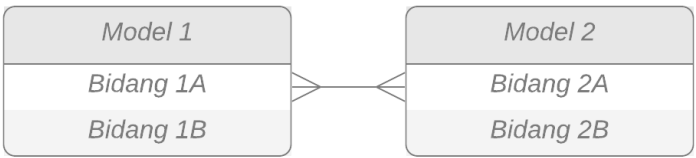
Dengan membuat *model* yang telah dirancang pada sub sub bab 3.3.2 mengenai perancangan model, Django akan secara otomatis membuat tabel basis data tersebut. Penulis ingin menggambarkan seluruh tabel basis data dari seluruh *model* beserta hubungannya dalam ERD menggunakan *Crow’s Foot Notation* . Namun karena keterbatasan tempat maka tabel basis data akan digambarkan dalam 3 kluster, kluster sistem pelaporan, kluster sistem keorganisasian, dan kluster sistem poin. Pada penggambaran ini terdapat model yang terulang karena terlibat pada lebih dari satu kluster.

Pada paradigma basis data SQL, hubungan banyak-dengan-banyak sebenarnya tidak menghubungkan kedua entitas secara langsung. Namun terdapat suatu tabel lain yang memuat hubungan kunci primer (*primary key*) dari suatu *model* dengan kunci primer dari *model* yang berhubungan. Namun untuk menyederhanakan penggambaran basis data ini, hubungan banyak-

dengan-banyak akan disederhanakan sesuai dengan keinginan awal pengembang yang dapat dilihat pada gambar 3.15.

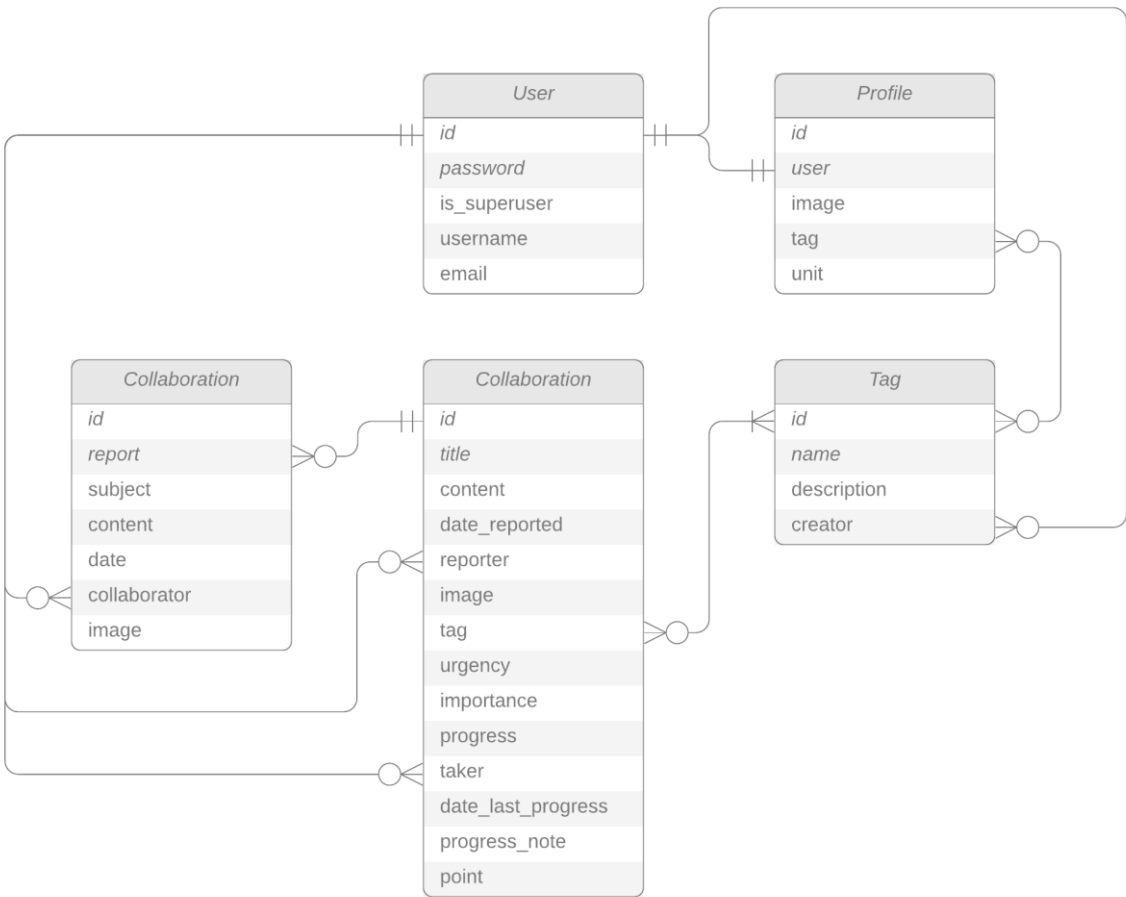


Yang terbuat di basis data SQL



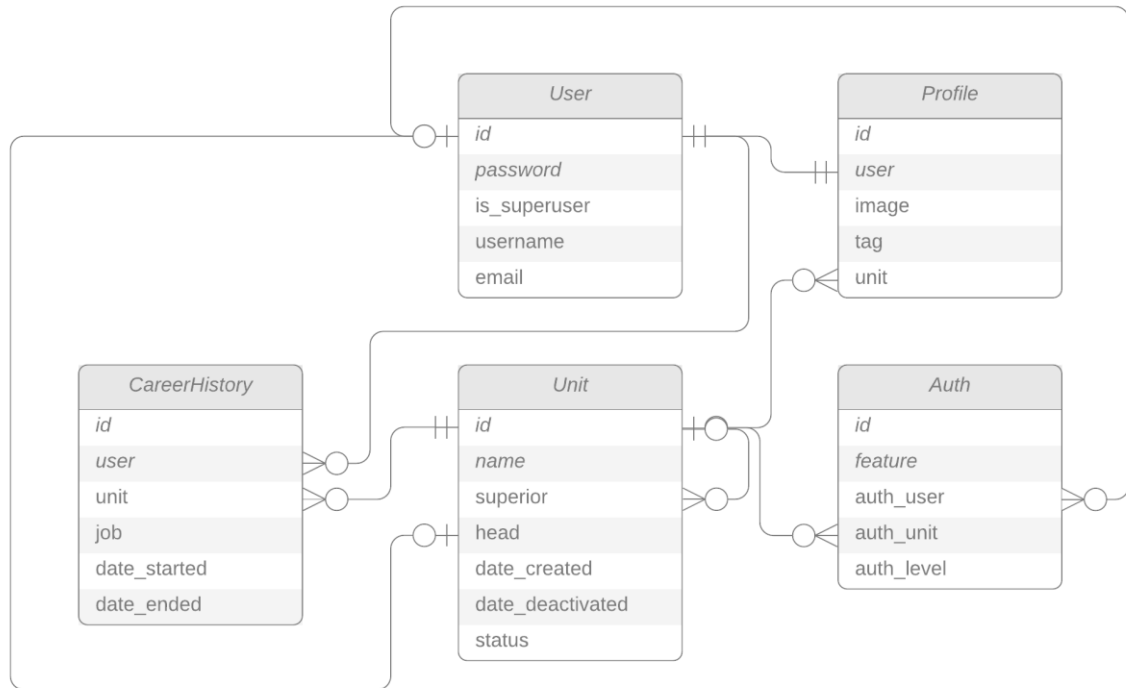
Keinginan pengembang dan penyederhaan dari penulis

Gambar 3.15 Penyederhanaan penggambaran hubungan banyak-dengan-banyak

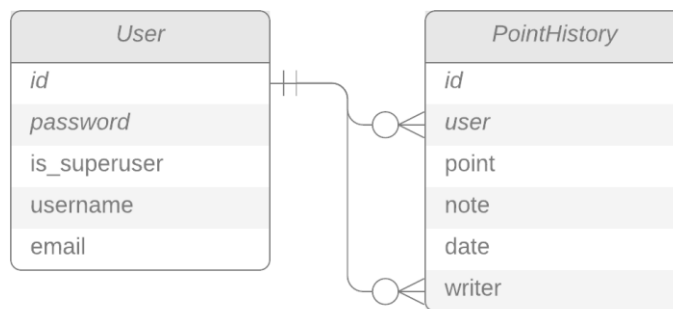


Gambar 3.16 ERD kluster sistem pelaporan

Kluster sistem pelaporan berisi tabel basis data dari *model* “*Report*”, “*Tag*”, “*Collaboration*”, “*User*”, dan “*Profile*”. ERD dari kluster sistem pelaporan digambarkan pada gambar 3.16. Kluster sistem keorganisasian tabel basis data dari *model* “*User*”, “*Profile*”, “*Unit*”, “*CareerHistory*”, dan “*Auth*”. ERD dari kluster sistem keorganisasian digambarkan pada gambar 3.17. Dan kluster sistem pelaporan berisi tabel basis data dari *model* “*User*” dan “*PointHistory*”. ERD dari kluster sistem poin digambarkan pada gambar 3.18.



Gambar 3.17 ERD kluster sistem keorganisasian



Gambar 3.18 ERD kluster sistem poin

3.4. Pengembangan Perangkat Lunak

Pada sub bab 3.3 mengenai perancangan sistem, telah ditentukan rancangan untuk menjawab kebutuhan pada sub bab 3.2. Selanjutnya adalah mengembangkan perangkat lunak sesuai dengan rancangan.

Dimulai dengan membuat proyek Django (*Django project*) untuk membuat direktori dan lingkungan virtual (*virtual environment*). Selanjutnya penulis membuat dua aplikasi, aplikasi “*report*” dan aplikasi “*user*”, untuk memisahkan *model*, *view*, dan direktori dari fitur dan laman berdasarkan kegunaan dasarnya. Pemisahan aplikasi ini sebenarnya hanya upaya membuat direktori kode menjadi lebih teratur dan mudah dikelola. Bila ingin, sebenarnya kedua aplikasi tersebut digabung menjadi satu aplikasi.

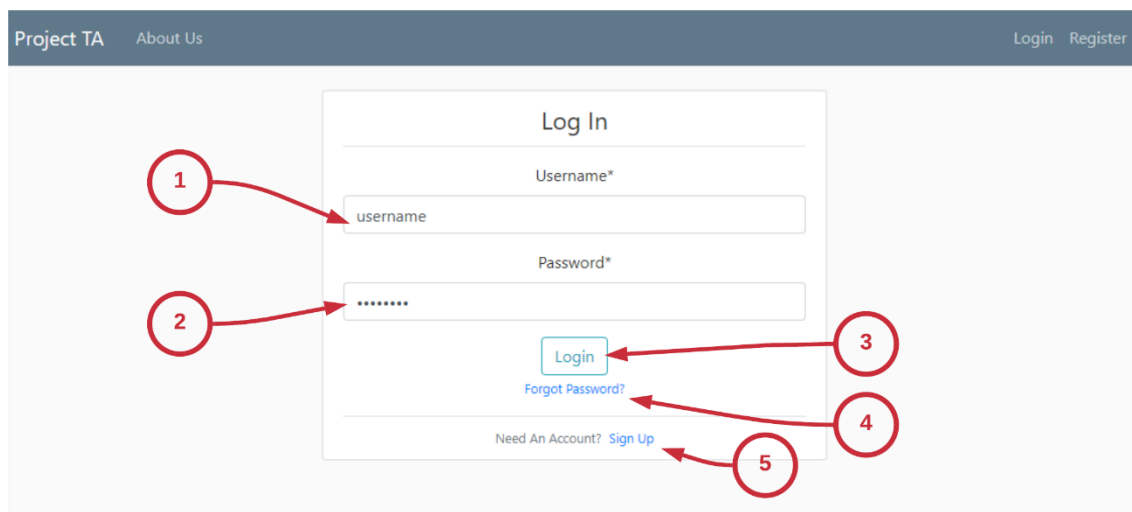
Aplikasi yang telah dibuat akan dihubungkan dengan proyek dengan memasukkan aplikasi tersebut pada *list* `INSTALLED_APPS` di berkas `settings.py`. Selain itu pada *list* `INSTALLED_APPS` juga ditambahkan *package* eksternal yang akan digunakan kedepannya seperti `crispy_forms` dan `django_filters`. Pada *list* `TEMPLATES` di berkas `settings.py` juga ditaur direktori *template* untuk setiap *view*. Diatur juga zona waktu menjadi area ‘Asia/Jakarta’. Dan yang terakhir ditambahkan pengaturan untuk mengatur direktori disimpannya berkas yang diunggah (*upload*) pengguna seperti gambar untuk gambar profil (*profile picture*) maupun bukti laporan.

Model yang telah dirancang di sub sub bab 3.3.2 akan dibuat di berkas (*file*) `model.py` di aplikasi terkait. *Model* “*User*” tidak dibuat penulis karena telah terbuat saat membuat proyek Django. *Model* “*Tag*”, “*Report*”, dan “*Collaboration*” akan ditempatkan pada aplikasi “*Report*”, sedangkan *Model* “*Unit*”, “*Profile*”, “*CareerHistory*”, “*Auth*”, dan “*PointHistory*” akan ditempatkan pada aplikasi “*User*”.

Selanjutnya dikembangkan laman-laman yang mengakomodasi aktivitas-aktivitas dari fitur yang telah dirancang pada sub sub bab 3.3.1. Pengembangan laman ini akan mengikuti alur kerja yang digambarkan pada diagram alir sebelah tengah di gambar 3.2. Pertimbangan dan pengembangan dari *view* dan *template* dari setiap laman akan dijabarkan pada sub sub bab berikut. Namun untuk menjaga agar laporan tugas akhir ini tidak terlalu panjang, sebagian pertimbangan dan pengembangan dari *view* dan *template* tersebut akan dilampirkan di lampiran. Adapun untuk penentuan URL, akan diawali dengan `127.0.0.1:8000` atau `localhost:8000` karena dalam pengembangan ini jaringan yang digunakan jaringan lokal IPv4 dengan hos lokal (*localhost*).

3.4.1. Pengembangan Laman ‘login’

Laman bernama ‘login’ ini dapat diakses dengan URL 127.0.0.1:8000/login/. Laman ini ditujukan untuk mengakomodasi aktivitas 6 fitur “Akun Pengguna” mengenai masuk ke dalam aplikasi web. Laman ini menggunakan *view* `django.view.auth.LoginView`. `LoginView` merupakan kelas *view* bawaan (*built-in*) dari Django, jadi penulis tidak perlu membuat fungsi *view* khusus untuk masuk (*login*). *Template* yang digunakan penulis terdapat di berkas `user/login.html`.



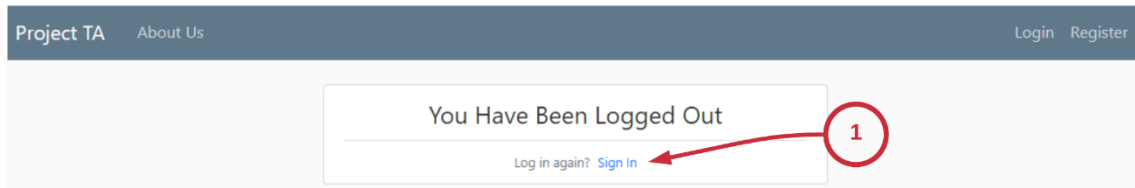
Gambar 3.19 Cuplikan layar laman ‘login’ dengan petunjuk

Pengguna dapat masuk ke aplikasi web melalui laman web. Pada awalnya pengguna dapat memasukkan nama pengguna (*username*) dan kata sandi (*password*) pada kolom isian yang ditunjuk nomor 1 dan nomor 2 secara berurutan pada gambar 3.19. Setelah memasukkan kedua data tersebut, pengguna dapat memencet tombol yang ditunjuk nomor 3 untuk masuk pada aplikasi web. Bila pengguna melupakan kata sandi, pengguna dapat memencet elemen yang ditunjuk nomor 4. Dan bila pengguna belum memiliki akun, pengguna dapat memencet elemen yang ditunjuk nomor 5.

3.4.2. Pengembangan Laman ‘logout’

Laman bernama ‘logout’ ini dapat diakses dengan URL 127.0.0.1:8000/logout/. Laman ini menjadi respons umpan balik kepada pengguna yang telah melakukan aktivitas 7 fitur “Akun Pengguna” mengenai keluar dari aplikasi web. Laman ini menggunakan *view* `django.view.auth.LogoutView`. `LogoutView` merupakan kelas *view* bawaan (*built-in*) dari

Django, jadi penulis tidak perlu membuat fungsi *view* khusus untuk menunjukkan bahwa pengguna telah keluar. *Template* yang digunakan penulis terdapat di berkas `user/logout.html`.

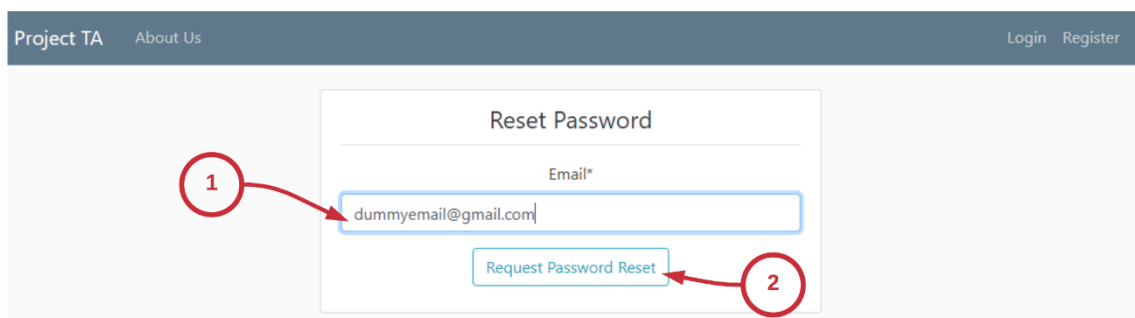


Gambar 3.20 Cuplikan layar laman *'logout'* dengan petunjuk

Setelah pengguna keluar dari aplikasi web, akan terakses laman *'logout'*. Laman ini membuat pengguna tahu bahwa ia telah keluar dari aplikasi web. Bila pengguna ingin masuk pada aplikasi web lagi, pengguna dapat memencet elemen yang ditunjuk nomor 1 pada gambar 3.20.

3.4.3. Pengembangan Laman *'password_reset'*

Laman bernama *'password_reset'* ini dapat diakses dengan URL `127.0.0.1:8000/password_reset/`. Laman ini ditujukan untuk mengakomodasi aktivitas 2 fitur “Akun Pengguna” mengenai memasukkan surel bila lupa kata sandi. Laman ini menggunakan *view* `django.view.auth.PasswordResetView`. `PasswordResetView` merupakan kelas *view* bawaan (*built-in*) dari Django, jadi penulis tidak perlu membuat fungsi *view* khusus untuk mengatur ulang kata sandi. *Template* yang digunakan penulis terdapat di berkas `user/password_reset.html`.



Gambar 3.21 Cuplikan layar laman *'password_reset'* dengan petunjuk

Bila pengguna melupakan kata sandi dari akunnya, pengguna dapat masuk ke laman ini. Pada laman ini pengguna dapat memasukkan alamat surel yang terasosiasi dengan akun tersebut

pada kolom isian yang ditunjuk nomor 1 pada gambar 3.21. Selanjutnya pengguna dapat memencet elemen yang ditunjuk nomor 2.

Laman yang menjadi tindak lanjut dari penyetelan ulang kata sandi akan berupa laman *'password_reset'*, *'password_reset'*, dan *'password_reset'*. Penjelasan mengenai pengembangan laman-laman tersebut akan dijabarkan pada lampiran sub bab A.1, A.2, dan A.3.

3.4.4. Pengembangan Laman *'register'*

Laman bernama *'register'* ini dapat diakses dengan URL 127.0.0.1:8000/register/. Laman ini ditujukan untuk mengakomodasi aktivitas 1 fitur “Akun Pengguna” mengenai pembuatan akun baru. Laman ini menggunakan fungsi *'register()'* sebagai *view*.

Fungsi *'register()'* ditulis penulis di berkas *user/views.py*. Fungsi ini akan memanggil formulir *'UserRegisterForm'* yang telah dibuat di berkas *user/form.py*. Formulir ini akan meminta bidang (*fields*) *username*, *email*, *password1*, dan *password2* untuk objek *'User'* baru. Kemudian formulir tersebut disalurkan ke *template* di berkas *user/register.html*.

Pada saat menerima data dari pengguna dengan metode POST (*request.POST*), formulir yang dikirim kembali akan diperiksa validitasnya. Bila form tersebut valid, pengguna akan dialihkan ke laman *'login'* dan terdapat pesan umpan balik yang memberi tahu bahwa akun telah dibuat. Bila formulir tidak valid, pengguna akan terus berada pada laman *'register'*.

The image shows a web browser window with a dark blue header. On the left, it says "Project TA" and "About Us". On the right, there are links for "Login" and "Register". The main content area is white and contains a "Register" form. The form has five input fields: "Username*", "Email*", "Password*", and "Password confirmation*". Below the "Password*" field, there are four bullet points: "Your password can't be too similar to your other personal information.", "Your password must contain at least 8 characters.", "Your password can't be a commonly used password.", and "Your password can't be entirely numeric." Below the "Password confirmation*" field, there is a text label: "Enter the same password as before, for verification." At the bottom of the form, there is a "Sign Up" button. Below the button, there is a link: "Already Have An Account? Sign In". There are six red circles with numbers 1 through 6, each with an arrow pointing to a specific part of the form: 1 points to the "Username*" input field, 2 points to the "Email*" input field, 3 points to the "Password*" input field, 4 points to the "Password confirmation*" input field, 5 points to the "Sign Up" button, and 6 points to the "Sign In" link.

Gambar 3.22 Cuplikan layar laman *'register'* dengan petunjuk

Pada laman '*register*', pengguna diharapkan memasukkan beberapa informasi pribadi untuk akun baru. Nama pengguna dapat diisi pada kolom isian yang ditunjuk oleh nomor 1 pada gambar 3.22. Selanjutnya alamat surel pengguna, kata sandi, dan konfirmasi kata sandi diisi pada kolom isian yang ditunjuk nomor 2, nomor 3, dan nomor 4 secara berurutan. Bila pengguna telah yakin dengan informasi yang dimasukkan, pengguna dapat memencet elemen yang ditunjuk nomor 5 untuk (mengirim informasi ke peladen (*server*) melalui metode POST dan) membuat akun baru. Bila pengguna telah memiliki akun, pengguna dapat diarahkan ke laman '*login*' dengan memencet elemen yang ditunjuk nomor 6.

3.4.5. Pengembangan Laman '*profile*'

Laman bernama '*profile*' ini dapat diakses dengan URL 127.0.0.1:8000/profile/. Laman ini ditujukan untuk mengakomodasi aktivitas 8 fitur "Akun Pengguna" mengenai perubahan data pribadi. Laman ini menggunakan fungsi '*profile()*' sebagai *view*.

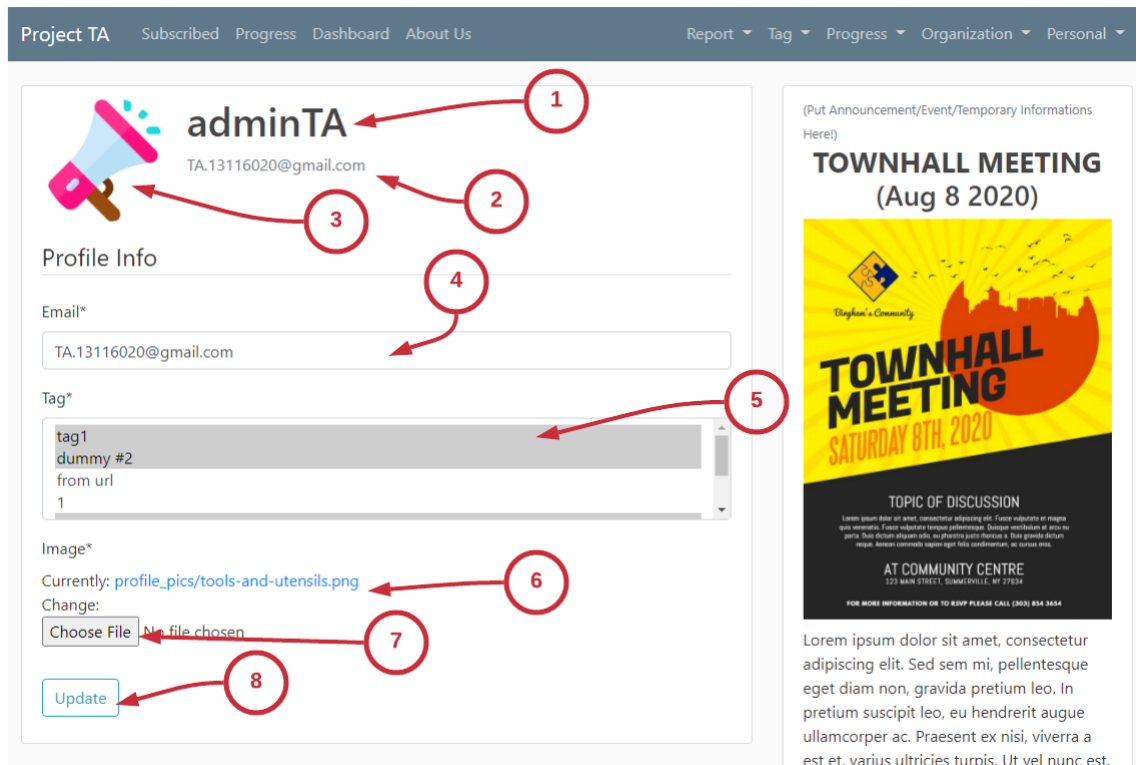
Fungsi '*profile()*' ditulis penulis di berkas user/views.py. Fungsi ini akan memanggil formulir '*UserUpdateForm*' dan '*ProfileUpdateForm*' yang telah dibuat di berkas user/form.py. Formulir '*UserUpdateForm*' akan meminta bidang (*field*) email untuk objek '*User*', sedangkan formulir '*ProfileUpdateForm*' akan meminta bidang (*fields*) tag dan image untuk objek Profile. Kemudian formulir tersebut disalurkan ke *template* di berkas user/profile.html.

Pada saat menerima data dari pengguna dengan metode POST (request.POST), formulir yang dikirim kembali akan diperiksa validitasnya. Bila form tersebut valid, data pengguna akan terganti dan terdapat pesan umpan balik. Bila formulir tidak valid, data pengguna tidak akan terganti.

Laman '*profile*' juga dilengkapi perintah keamanan. Pengguna yang tidak masuk (*login*) tidak dapat mengakses laman ini dan dialihkan ke laman '*login*'.

Pada lama '*profile*' pengguna dapat melihat nama pengguna, alamat surel, gambar pengguna (*profile picture*), dan nama berkas gambar pengguna yang tersimpan saat ini dengan melihat elemen yang ditunjuk oleh nomor 1, nomor 2, nomor 3, dan nomor 6 pada gambar 3.23 secara berurutan. Bila pengguna ingin mengubah alamat surel, pengguna dapat mengisi kolom isian yang ditunjuk nomor 4. Bila pengguna ingin mengubah *tag* yang dilanggan, pengguna dapat memilih *tag* yang tersedia di elemen yang ditunjuk nomor 5. Dan bila pengguna ingin mengubah gambar profil, pengguna dapat mengunggah gambar profil baru di elemen yang ditunjuk nomor 7. Dan bila pengguna telah yakin dengan informasi pengguna yang baru,

pengguna dapat memencet elemen yang ditunjuk nomor 8 untuk (mengirim informasi ke peladen (*server*) melalui metode POST dan) mengubah informasi pengguna.



Gambar 3.23 Cuplikan layar laman 'profile' dengan petunjuk

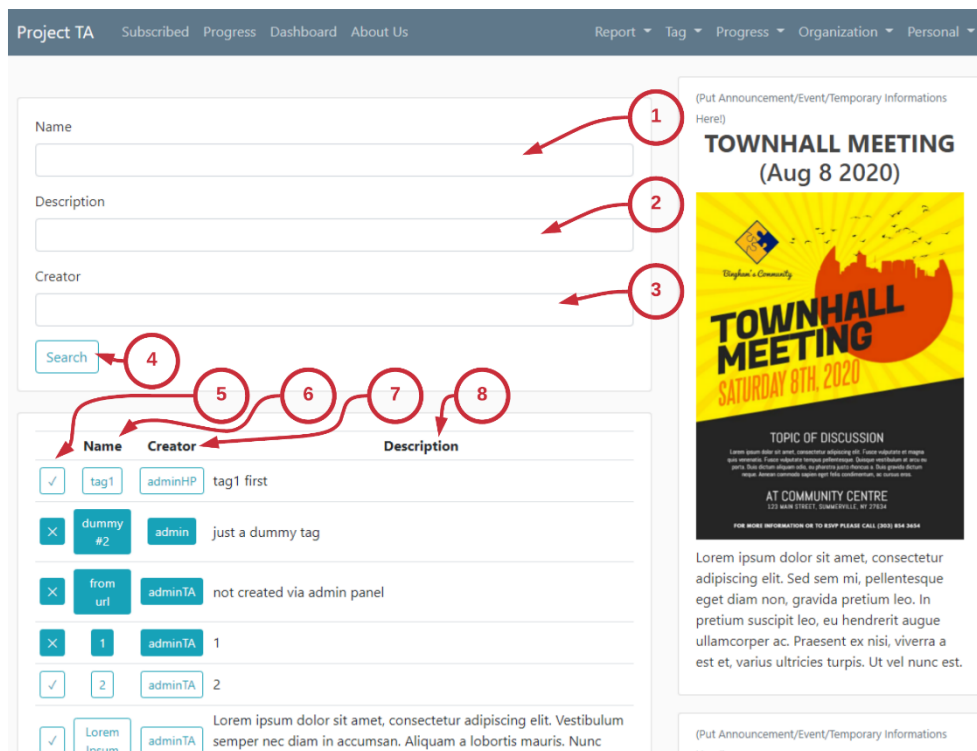
3.4.6. Pengembangan Laman 'tag-list'

Laman bernama 'tag-list' ini dapat diakses dengan URL 127.0.0.1:8000/tag-list/. Laman ini ditujukan untuk mengakomodasi aktivitas 1 fitur 1 "Membuat Laporan", aktivitas 1 fitur 2 "Melihat Laporan", dan aktivitas 1 fitur 3 "Membuat Tag" mengenai penglihatan daftar tag. Laman ini menggunakan fungsi 'TagList()' sebagai *view*.

Fungsi 'TagList()' ditulis penulis di berkas report/views.py. Fungsi ini akan mengumpulkan semua tag untuk ditampilkan kepada pengguna. Agar pengguna dapat mencari tag yang diinginkan dengan lebih mudah, penulis menambahkan kelas TagFilter yang merupakan subkelas dari kelas FilterSet yang berasal dari package django_filters. Kelas TagFilter ditulis di berkas report/filters.py. TagFilter dapat menyaring nama, deskripsi, maupun pembuat suatu tag. Fungsi 'TagList()' akan ditampilkan melalui template di berkas report/tag_list.html.

Laman 'tag-list' juga dilengkapi dengan perintah keamanan. Pertama diperiksa apakah pengguna sudah masuk (*login*). Bila belum pengguna akan dialihkan ke laman masuk. Kedua

diperiksa apakah pengguna memiliki otoritas atas salah satu dari fitur 1, fitur 2, atau fitur 3. Bila pengguna tidak memiliki otoritas, pengguna akan dialihkan ke laman 'home'.



Gambar 3.24 Cuplikan layar laman 'tag-list' dengan petunjuk

Pada laman 'tag-list' pengguna dapat mencari *tag* yang diinginkan. Kolom yang ditunjuk nomor 5, nomor 6, nomor 7, dan nomor 8 pada gambar 3.24 menunjukkan status langganan pengguna, nama, pembuat, dan deskripsi dari suatu *tag* secara berurutan. Satu *tag* ditampilkan pada satu baris. *Tag* yang telah dilanggan berwarna terang, sedangkan *tag* yang tidak dilanggan akan berwarna gelap. Untuk mempermudah pencarian pengguna dapat menggunakan filter yang disediakan. Dengan memasukkan sebagian dari nama, sebagian dari deskripsi, maupun sebagian dari nama pembuat dari suatu *tag* pada kolom isian yang ditunjuk nomor 1, nomor 2, dan nomor 3 secara berurutan, pengguna dapat menyaring *tag* yang ditampilkan.

Terdapat juga dua laman serupa yang ditujukan untuk mencari suatu objek *model* dengan bantuan filter, yaitu laman 'point-list' dan 'user-list'. Penjelasan mengenai pengembangan laman-laman tersebut akan dijabarkan pada lampiran sub bab A.4 dan A.5.

3.4.7. Pengembangan Laman ‘report-create’

Laman bernama ‘report-create’ ini dapat diakses dengan URL 127.0.0.1:8000/report-new/. Laman ini ditujukan untuk mengakomodasi aktivitas 2 fitur 1 “Membuat Laporan” mengenai pembuatan laporan baru. Laman ini menggunakan kelas ‘ReportCreateView’ sebagai *view*.

Kelas ‘ReportCreateView’ ditulis penulis di berkas report/views.py. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) CreateView sehingga menerima pewarisan (*inheritance*) dari kelas CreateView. Kelas CreateView telah menangani pembuatan formulir untuk membuat objek ‘Report’ baru dan fungsi untuk menyimpan objek tersebut. Pada form ini diminta bidang title, content, image, tag, urgency, dan importance pada objek ‘Report’. Untuk bidang reporter akan dihubungkan dengan objek ‘User’ yang terasosiasi dengan pengguna yang sedang membuat laporan melalui metode form_valid(). Kelas ini akan ditampilkan melalui *template* di berkas report/report_form.html.

The screenshot shows the 'New Report' form in the Project TA application. The form has a dark blue header with navigation links: Project TA, Subscribed, Progress, Dashboard, About Us, Report, Tag, Progress, Organization, and Personal. The form itself is white with a light blue border. It contains the following fields and elements:

- Title***: A text input field with a red circle 1 pointing to it.
- Content***: A large text area with a red circle 2 pointing to it.
- Image***: A section with a 'Choose File' button and 'No file chosen' text. A red circle 3 points to the 'Choose File' button.
- Tag***: A dropdown menu showing 'tag1', 'dummy #2', 'from url', and '1'. A red circle 4 points to the dropdown.
- Urgency***: A dropdown menu with a red circle 5 pointing to it.
- Importance***: A dropdown menu with a red circle 6 pointing to it.
- Upload**: A blue button with a red circle 7 pointing to it.

To the right of the form, there are two sample announcement cards. The top card is titled 'TOWNHALL MEETING (Aug 8 2020)' and features a yellow background with a red sun and the text 'TOWNHALL MEETING SATURDAY 8TH, 2020'. The bottom card is titled 'INTERNAL BADMINTON SPARRING (Aug 17 2020)' and has a white background with a blue border.

Gambar 3.25 Cuplikan layar laman ‘report-create’ dengan petunjuk

Laman *'report-create'* juga dilengkapi dengan perintah keamanan. Digunakan `LoginRequiredMixin` untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga `UserPassesTestMixin` yang dijelaskan dengan metode `test_func()` untuk memastikan pengguna memiliki otoritas atas fitur 1 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari `LoginRequiredMixin` pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari `UserPassesTestMixin` tidak dipenuhi, pengguna akan diarahkan ke laman *'home'* menggunakan metode `handle_no_permission()`.

Pada laman *'report-create'* pengguna dapat memasukkan judul dan konten laporan pada kolom isian yang ditunjuk nomor 1 dan 2 pada gambar 3.25 secara berurutan. Gambar dapat diunggah melalui elemen yang ditunjuk nomor 3. *Tag* terkait dapat dipilih pada pilihan yang ditunjuk nomor 4. Tingkat keterdesakan dan kepentingan laporan juga dapat dipilih pada elemen yang ditunjuk nomor 5 dan nomor 6 secara berurutan. Bila pengguna telah yakin untuk membuat laporan baru, pengguna dapat memencet tombol yang ditunjuk nomor 7 untuk membuat objek *'Report'* baru.

Terdapat juga tiga laman serupa yang ditujukan untuk membuat suatu objek model dengan kelas super (*superclass*) `CreateView`, yaitu laman *'tag-create'*, *'unit-create'*, dan *'point-create'*. Penjelasan mengenai pengembangan laman-laman tersebut akan dijabarkan pada lampiran sub bab A.6, A.7, dan A.8.

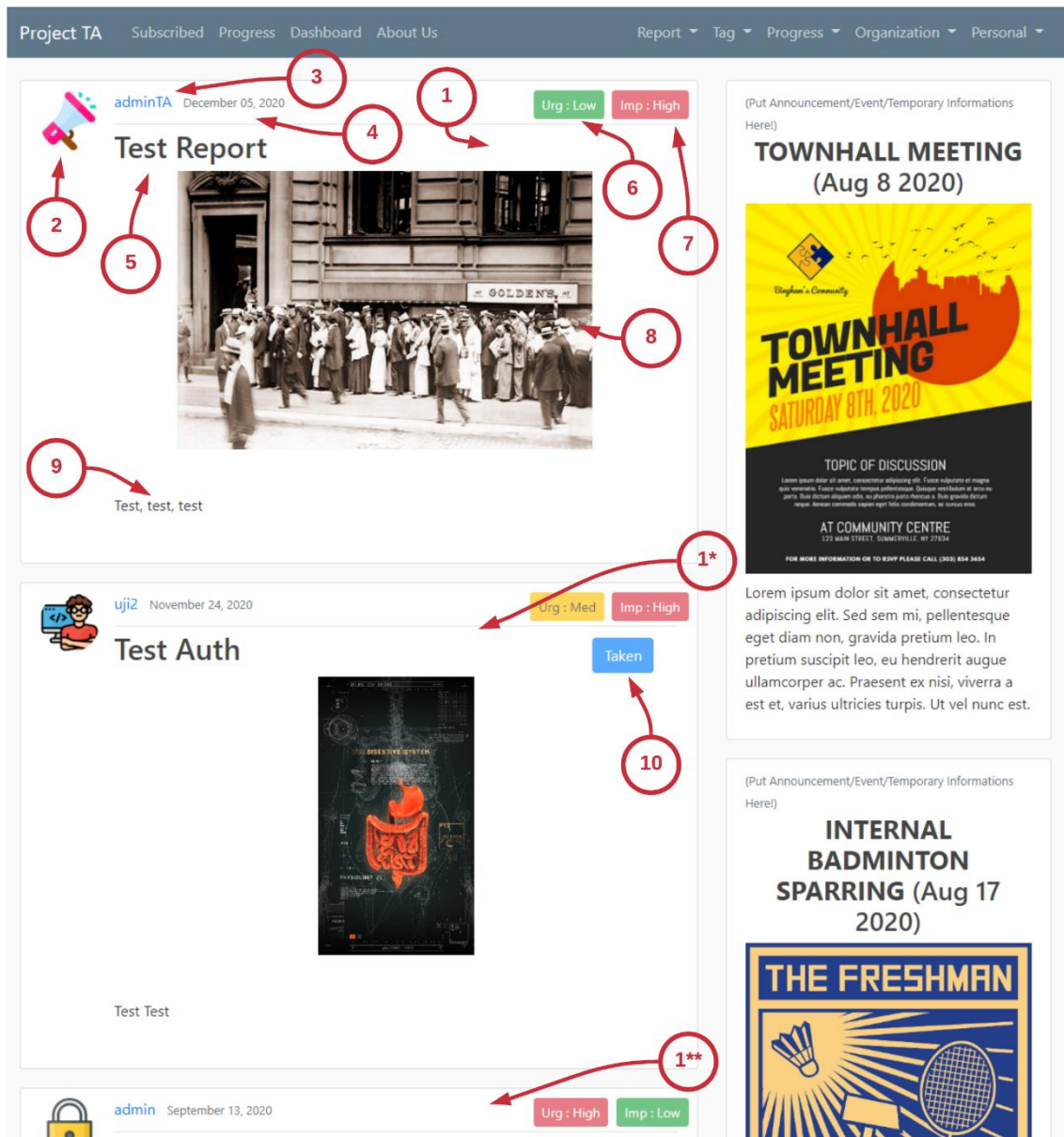
3.4.8. Pengembangan Laman *'all-reports'*

Laman bernama *'report-create'* ini dapat diakses dengan URL `127.0.0.1:8000/report-all/`. Laman ini ditujukan untuk mengakomodasi aktivitas 4 fitur 2 “Melihat Laporan” mengenai penglihatan semua laporan. Laman ini menggunakan kelas `'ReportListView'` sebagai *view*.

Kelas `'ReportListView'` ditulis penulis di berkas `report/views.py`. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) `ListView` sehingga menerima pewarisan (*inheritance*) dari kelas `ListView`. Kelas `ReportListView` akan memunculkan semua objek *'Report'* yang ada pada sistem. Selanjutnya sejumlah objek *'Report'* akan disusun berdasarkan tanggal laporannya, kemudian berdasarkan kemendesakannya (*urgency*), dan yang terakhir berdasarkan kepentingannya (*importance*). Kelas ini akan ditampilkan melalui *template* di berkas `report/report_list.html`.

Laman *'all-reports'* juga dilengkapi dengan perintah keamanan. Digunakan `LoginRequiredMixin` untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses

laman ini. Digunakan juga `UserPassesTestMixin` yang dijelaskan dengan metode `test_func()` untuk memastikan pengguna memiliki otoritas atas fitur 2 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari `LoginRequiredMixin` pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari `UserPassesTestMixin` tidak dipenuhi, pengguna akan diarahkan ke laman 'home' menggunakan metode `handle_no_permission()`.



Gambar 3.26 Cuplikan layar laman 'all-reports' dengan petunjuk

Pada laman 'all-reports' pengguna dapat melihat semua laporan yang ada pada sistem. Pada gambar 3.26 pada nomor 1, nomor 1*, dan nomor 1** ditunjukkan 3 laporan yang ditampilkan di laman ini. Terlihat bahwa setiap laporan berada dalam satu seksi konten (*content*

section). Di dalam suatu seksi konten terdapat informasi mengenai suatu laporan. Sebagai contoh, pada laporan pada seksi konten yang ditunjuk nomor 1, ditampilkan gambar profil pelapor, nama pelapor, tanggal dibuat, judul, tingkat keterdesakan, tingkat kepentingan, gambar bukti, dan deskripsi dari laporan tersebut pada elemen yang ditunjuk nomor 2, nomor 3, nomor 4, nomor 5, nomor 6, nomor 7, nomor 8, dan nomor 9 secara berurutan. Pengguna dapat melihat laporan lainnya dengan menggulirkan (*scroll*) laman ke bawah.

Pada laporan yang ditunjuk nomor 1, dapat diketahui bahwa laporan tersebut belum diambil, karena tidak ada tanda bertuliskan “*Taken*”. Namun bila kita melihat laporan yang ditunjuk nomor 1*, terlihat tanda bertuliskan “*Taken*” ditunjuk oleh nomor 10. Hal tersebut berarti laporan yang ditunjuk nomor 1* telah diambil untuk ditindaklanjuti.

Terdapat juga dua laman serupa yang ditujukan untuk menampilkan objek ‘*Report*’ dengan kelas super (*superclass*) *DetailView*, yaitu laman ‘*tag-reports*’ dan ‘*home*’. Penjelasan mengenai pengembangan laman-laman tersebut akan dijabarkan pada lampiran sub bab A.9 dan A.10.

3.4.9. Pengembangan Laman ‘*report-detail*’

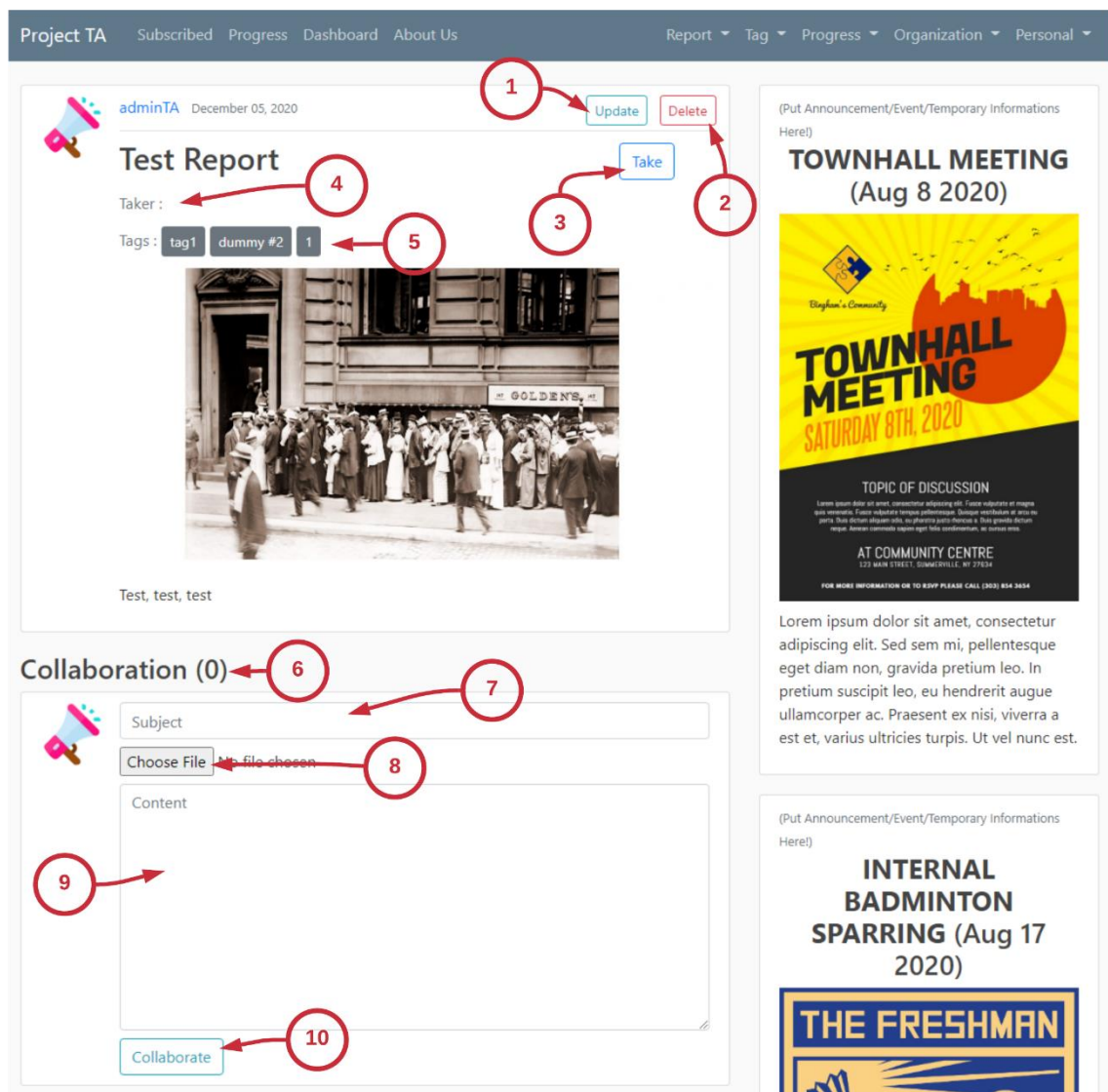
Laman bernama ‘*report-detail*’ ini dapat diakses dengan URL 127.0.0.1:8000/report/<int:pk>/. Dimana <int:pk> adalah kunci primer (*primary key*) dari objek ‘*Report*’ yang ingin diakses dalam bentuk bilangan bulat (*integer*). Laman ini ditujukan untuk mengakomodasi aktivitas 3 fitur 1 “Membuat Laporan” dan aktivitas 8 fitur 2 “Melihat Laporan” mengenai penglihatan detail laporan, aktivitas 1 dan aktivitas 2 dari fitur 9 “Berkolaborasi pada Laporan” mengenai mengelola kolaborasi, serta aktivitas 1, aktivitas 2, dan aktivitas 3 dari fitur 4 “Mengambil Laporan” mengenai pengambilan laporan dan penanganannya setelahnya. Laman ini menggunakan kelas ‘*ReportDetailView*’ sebagai *view*.

Kelas ‘*ReportDetailView*’ ditulis penulis di berkas report/views.py. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) *DetailView* sehingga menerima pewarisan (*inheritance*) dari kelas *DetailView*. Kelas *ReportDetailView* ditujukan untuk menunjukkan segala detail informasi mengenai suatu objek ‘*Report*’. Ditampilkan juga semua objek ‘*Collaboration*’ yang terkait dengan laporan tersebut. Kelas ini akan ditampilkan melalui *template* di berkas report/report_detail.html.

Kelas *ReportDetailView* menerima data dari pengguna dengan metode POST (*request.POST*). Secara garis besar terdapat 4 kelompok data yang dikirim pengguna. Kelompok data pertama adalah data mengenai pengambilan suatu tugas oleh seorang pengguna.

Kelompok data kedua adalah mengenai pembuatan objek ‘*Collaboration*’ saat pengguna ingin berkolaborasi. Kelompok data ketiga adalah saat seorang pengguna ingin menghapus objek ‘*Collaboration*’ yang ia buat. Dan kelompok data terakhir mengenai pembaruan (*update*) informasi mengenai penanganan laporan yang telah diambil oleh suatu pengguna.

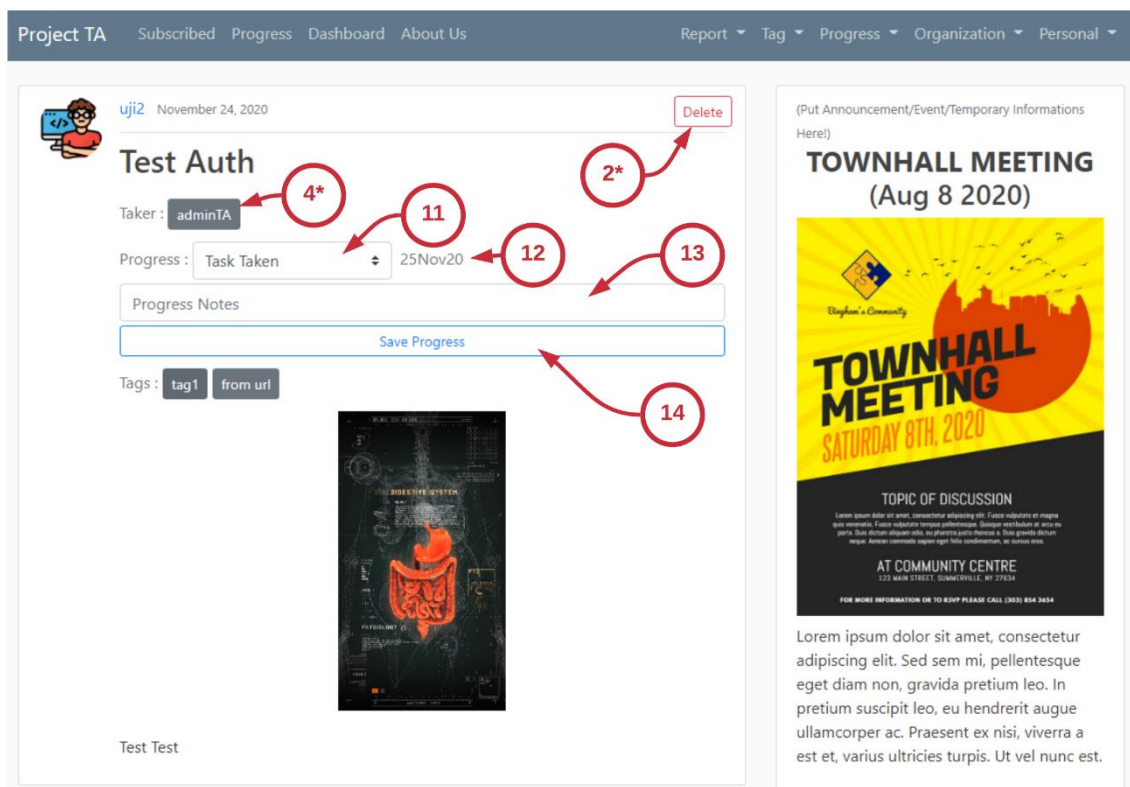
Laman ‘*report-detail*’ juga dilengkapi dengan perintah keamanan. Digunakan LoginRequiredMixin untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga UserPassesTestMixin yang dijelaskan dengan metode *test_func()* untuk memastikan bahwa pengguna adalah pelapor dari laporan ini atau pengguna memiliki otoritas atas fitur 2 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari LoginRequiredMixin pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari UserPassesTestMixin tidak dipenuhi, pengguna akan diarahkan ke laman ‘*home*’ menggunakan metode *handle_no_permission()*.



Gambar 3.27 Cuplikan layar laman ‘*report-detail*’ pada kasus 1 dengan petunjuk

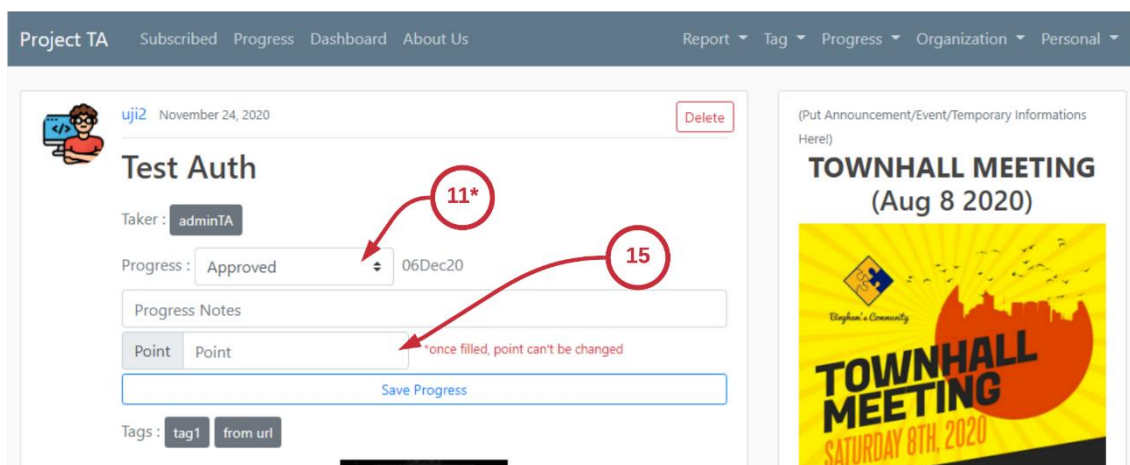
Karena menunjukkan informasi dari suatu laporan, laman *'report-detail'* memiliki kesamaan dengan laman *'all-reports'*. Maka dari itu informasi dasar mengenai laporan dapat dilihat di sub sub bab 3.4.8. Namun laman *'report-detail'* menyediakan informasi yang lebih seperti pengambil dari laporan dan *tag* yang dikaitkan dengan laporan terkait yang ditunjuk nomor 4 dan nomor 5 pada gambar 3.27. Pada kasus ini terlihat bahwa belum ada pengguna yang mengambil laporan ini, sehingga elemen yang ditunjuk nomor 4 masih kosong. Karena laporan tersebut belum diambil, akan terlihat tombol untuk mengambil laporan bertuliskan *"Take"* seperti yang ditunjuk nomor 3. Bila pengguna ingin mengambil laporan tersebut, pengguna dapat memencet tombol tersebut.

Bila suatu pengguna diberi otorisasi untuk mengakses fitur 9 *"Berkolaborasi pada Laporan"*, pengguna dapat melihat seksi konten (*content section*) di bagian paling bawah laman *'report-detail'* yang ditujukan untuk membuat kolaborasi baru. Pada seksi konten tersebut terdapat kolom isian untuk judul dan isi kolaborasi pada elemen yang ditunjuk nomor 7 dan nomor 9 secara berurutan. Pengguna juga dapat mengunggah gambar mengenai kolaborasi pada elemen yang ditunjuk nomor 8. Bila pengguna sudah yakin dengan kolaborasi yang akan dibuat, pengguna dapat memencet tombol yang ditunjuk nomor 10. Jumlah kolaborasi dari laporan terkait akan diperlihatkan oleh elemen yang ditunjuk nomor 6.



Gambar 3.28 Cuplikan layar laman *'report-detail'* pada kasus 2 dengan petunjuk

Bila pengguna telah mengambil suatu laporan, tampilan laman *'report-detail'* akan sedikit berubah menjadi seperti pada gambar 3.28. Terlihat bahwa elemen yang ditunjuk nomor 4* pada gambar 3.28 memperlihatkan bahwa laporan ini telah diambil oleh pengguna dengan nama pengguna *'adminTA'*. Dalam kasus ini penulis masuk menggunakan akun dengan nama pengguna *'adminTA'*, berarti gambar 3.28 menunjukkan tampilan suatu laporan kepada pengambilnya. Untuk pengambil suatu laporan akan disediakan pilihan untuk mengubah status suatu laporan (berdasarkan siklus hidup laporan pada alur kerja utama yang telah dirancang pada sub bab 3.3.1) dan kolom isian untuk memberi catatan terhadap laporan yang pengguna tersebut ambil pada elemen yang ditunjuk nomor 11 dan nomor 13 secara berurutan. Bila pengguna telah yakin untuk mengubah status dan/atau memberi catatan, pengguna dapat memencet tombol yang ditunjuk nomor 14 untuk menyimpan perubahan tersebut. Tanggal terakhir kali perubahan dilakukan akan ditunjukkan pada elemen nomor 12.

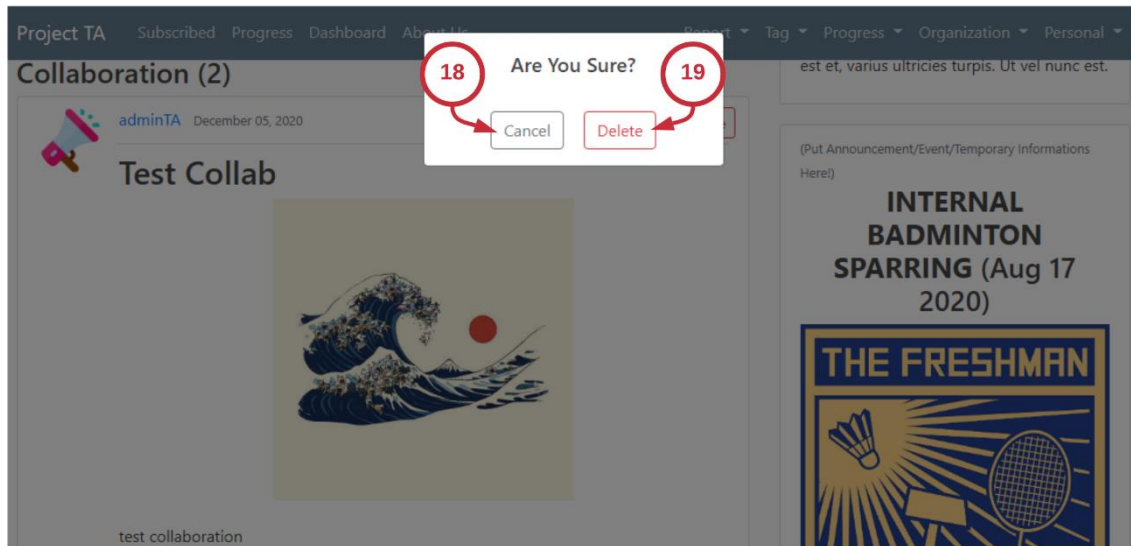


Gambar 3.29 Cuplikan layar laman *'report-detail'* pada kasus 3 dengan petunjuk

Berdasarkan siklus hidupnya, bila suatu laporan telah disetujui (*"Approved"*) atau ditolak (*"Not Approved"*), pelapor dari laporan tersebut dapat diberi insentif poin sebagai penguatan atau hukuman skema gamifikasi. Poin dapat diberikan setelah laporan selesai, baik karena laporan terselesaikan (*"Resolved"*), gagal (*"Failed"*), maupun ditunda (*"Postponed"*), namun tidak harus menunggu selesai karena penyelesaian laporan mungkin membutuhkan waktu yang lama. Pada gambar 3.29 ditunjukkan kasus bahwa suatu laporan telah disetujui, hal tersebut ditunjukkan dari elemen yang ditunjuk nomor 11* pada gambar 3.29. Karena telah disetujui, akan muncul kolom isian untuk memberi poin yang ditunjuk oleh nomor 15. Poin yang telah diberikan tidak dapat ditarik atau diubah jumlahnya, jadi pengambil laporan harus memikirkannya matang-matang sebelum memberi poin.

Bila laporan telah diambil pengguna lain, namun bukan kita tampilan laman *'report-detail'* akan berbeda lagi seperti pada gambar 3.30. Pada kasus ini penulis masuk dengan akun *'adminTA'*, namun yang mengambil laporan ini adalah pengguna dengan akun *'Ujihp2'*. Maka dari itu pada kasus ini penulis melihat dari sudut pandang orang ketiga, bukan sebagai pembuat laporan maupun sebagai pengambil laporan. Berbeda dengan kasus pada gambar 3.28 dan gambar 3.29 dimana penulis menjadi pengambil laporan, penulis dapat mengubah status laporan dan memberi catatan kepada suatu laporan. Pada kasus pada gambar 3.30, sebagai orang ketiga, penulis hanya bisa melihat status dan catatan dari suatu laporan tanpa bisa mengubahnya seperti yang terlihat pada elemen yang ditunjuk nomor 11* dan 13* pada gambar 3.30.

Pada gambar 3.30 juga diperlihatkan contoh kasus bila terdapat kolaborasi pada suatu laporan. Dapat terlihat seperti yang ditunjuk nomor 16 dan 16* pada gambar 3.30 bahwa setiap kolaborasi akan berada pada satu seksi konten (*content section*). Pada seksi konten yang ditunjuk nomor 16, terlihat bahwa kolaborasi tersebut dibuat oleh penulis, yang masuk dengan akun bernama *'adminTA'*. Sedangkan pada seksi konten yang ditunjuk nomor 16*, terlihat bahwa kolaborasi tersebut bukan dibuat oleh penulis. Terdapat perbedaan antara kedua kolaborasi tersebut yaitu keberadaan tombol untuk menghapus kolaborasi yang bertuliskan *"Delete"* seperti elemen yang ditunjuk nomor 17. Hal tersebut karena kita hanya bisa menghapus kolaborasi yang kita buat.



Gambar 3.31 Cuplikan layar laman *'report-detail'* pada kasus 5 dengan petunjuk

Bila pengguna ingin menghapus kolaborasi yang telah ia buat, akan muncul pertanyaan untuk mengkonfirmasi penghapusan kolaborasi tersebut. Akan muncul elemen *modal* (yang

dibuat dengan Bootstrap 4) seperti pada gambar 3.31. Bila pengguna ingin mengurungkan keinginan menghapusnya, pengguna dapat memencet tombol yang ditunjuk nomor 18 pada gambar 3.31. Sedangkan bila pengguna telah yakin untuk menghapus kolaborasi tersebut, pengguna dapat memencet tombol yang ditunjuk nomor 19.

Perihal menghapus dan mengubah laporan dilakukan melalui laman *'report-detail'*. Pelapor dari suatu laporan mengubah suatu laporan sebelum laporan tersebut diambil. Pengubahan dapat dilakukan dengan memencet tombol yang ditunjuk nomor 1 pada gambar 3.27. Sedangkan penghapusan suatu laporan dapat dilakukan oleh pelapor suatu laporan bila laporan tersebut belum diambil. Namun bila suatu laporan telah diambil, hanya pengambil laporan yang berhak menghapus laporan tersebut. Hal tersebut karena laporan tersebut telah di bawah tanggung jawab pengambil laporan. Penghapusan dapat dilakukan dengan memencet tombol yang ditunjuk nomor 2 pada gambar 3.27 atau nomor 2* yang ditunjuk pada gambar 3.28. Tombol penubahan akan mengalihkan pengguna ke laman *'report-update'*, sedangkan tombol penghapusan akan mengalihkan pengguna ke laman *'report-delete'*. Penjelasan mengenai pengembangan laman *'report-update'* dan *'report-delete'* akan dijabarkan pada lampiran sub bab A.11 dan A.12.

3.4.10. Pengembangan Laman *'progress-taken'*

Laman bernama *'progress-taken'* ini dapat diakses dengan URL 127.0.0.1:8000/progress-taken/. Laman ini ditujukan untuk mengakomodasi aktivitas 4 fitur 4 “Mengambil Laporan” mengenai penglihatan papan kanban pribadi. Laman ini menggunakan fungsi *'ProgressTaken()'* sebagai *view*.

Fungsi *'ProgressTaken()'* ditulis penulis di berkas *report/views.py*. Fungsi ini akan mengumpulkan semua laporan yang telah diambil pengguna dan memisahkannya berdasarkan status laporan tersebut serta diurut berdasarkan keterdesakan, kemudian berdasarkan kepentingan, dan yang terakhir berdasarkan tanggal dilaporkan. Secara garis besar laporan yang diambil akan dipisahkan menjadi 3 kelompok, laporan yang masih berjalan, laporan yang sudah selesai, dan laporan yang ditolak. Fungsi *'ProgressTaken()'* akan ditampilkan melalui *template* di berkas *report/progress_taken.html*.

Laman *'progress-taken'* juga dilengkapi dengan perintah keamanan. Pertama diperiksa apakah pengguna sudah masuk (*login*). Bila belum pengguna akan dialihkan ke laman masuk. Kedua diperiksa apakah pengguna memiliki otoritas atas fitur 4 “Mengambil Laporan”. Bila pengguna tidak memiliki otoritas, pengguna akan dialihkan ke laman *'home'*.

Laman *'progress-taken'* ditujukan untuk menunjukkan laporan yang telah diambil. Secara garis besar laman *'progress-taken'* dibagi menjadi 3 bagian, bagian laporan yang sedang ditangani, bagian laporan yang telah selesai ditangani, dan bagian laporan yang ditolak. Setiap bagian ditampilkan pada satu seksi konten (*content section*).

Project TA Subscribed Progress Dashboard About Us Report ▾ Tag ▾ Progress ▾ Organization ▾ Personal ▾

Ongoing (7)

	Report	Field Checking	Approved	Responding	Note
07Dec20 10:35	Urg-Imp Hi-Hi		✓		Telah disetujui
07Dec20 10:34	Test Kanban 1				Baru diambil
07Dec20 10:35	Test Auth	✓			Sedang diperiksa
07Dec20 10:36	Urg-Imp Med-Med	✓			Dalam pemeriksaan
27Oct20 22:31	test progress-susbscribed				
07Dec20 10:36	Urg-Imp Low-Low		✓		Sedang diselesaikan
07Dec20 10:36	report 3		✓		Hampir selesai

Finished (3)

	Report	Resolved	Postponed	Failed	Note
13Sep20 13:28	image2	✓			
13Sep20 13:27	Ini gelas	✓			gelas telah pecah
12Sep20 20:01	test image	✓			

Not Approved (1)

	Report	Reporter	Note
13Sep20 13:44	a	adminTA	Bohong

TOWNHALL MEETING (Aug 8 2020)

Elayham's Community

TOWNHALL MEETING SATURDAY 8TH, 2020

TOPIC OF DISCUSSION

AT COMMUNITY CENTRE
123 MAIN STREET, SUMMERSVILLE, NY 11764

FOR MORE INFORMATION OR TO RSVP PLEASE CALL (800) 854 3434

INTERNAL BADMINTON SPARRING (Aug 17 2020)

THE FRESHMAN

Gambar 3.32 Cuplikan layar laman *'progress-taken'* dengan petunjuk

Pada bagian laporan yang sedang ditangani akan menampilkan laporan dengan status *'Task Taken'*, *'Field Checking'*, *'Approved'*, dan *'Responding'*. Pada bagian ini setiap laporan akan ditunjukkan pada satu baris, dan informasinya terletak pada kolom. Elemen yang ditunjuk nomor 1 pada gambar 3.32 menampilkan jumlah laporan yang sedang ditangani pengguna. Kolom yang ditunjuk nomor 2 menunjukkan tanggal terakhir diubahnya status atau catatan pada

laporan tersebut. Kolom yang ditunjuk nomor 3 menunjukkan tombol dengan nama laporan tersebut dan warnanya menunjukkan tingkat keterdesakan (*urgency level*) dari laporan tersebut. Tombol berwarna yang berisi nama laporan tersebut dapat dipencet bila pengguna ingin ke laman '*report-detail*' dari laporan terkait. Kolom yang ditunjuk nomor 4, nomor 5, dan nomor 6 menunjukkan status dari laporan terkait. Bila suatu baris tidak memiliki tanda centang, berarti status laporan tersebut baru diambil ('*Task Taken*'). Bila terdapat tanda centang pada kolom yang ditunjuk nomor 4 pada baris tersebut, berarti laporan tersebut sedang diperiksa ('*Field Checking*'). Bila terdapat tanda centang pada kolom yang ditunjuk nomor 5 pada baris tersebut, berarti laporan tersebut sudah disetujui ('*Approved*'). Dan bila terdapat tanda centang pada kolom yang ditunjuk nomor 6 pada baris tersebut, berarti laporan tersebut sedang ditangani atau dikerjakan ('*Responding*'). Kolom yang ditunjuk nomor 7 menampilkan catatan dari laporan tersebut bila ada.

Pada bagian laporan yang telah selesai ditangani akan menampung laporan dengan status '*Resolved*', '*Postponed*', dan '*Failed*'. Cara membaca bagian ini serupa dengan cara membaca pada bagian laporan yang sedang ditangani. Elemen yang ditunjuk nomor 8 menampilkan jumlah laporan yang telah diselesaikan pengguna. Kolom yang ditunjuk nomor 9 dan nomor 10 menunjukkan tanggal terakhir diubahnya status atau catatan laporan dan nama laporan secara berurutan. Tombol yang bertuliskan nama laporan juga dapat dipencet bila pengguna ingin melihat laman '*report-detail*' dari laporan terkait. Tanda centang pada kolom yang ditunjuk nomor 11, nomor 12, atau nomor 13 menunjukkan status dari laporan yang telah selesai berupa terselesaikan, ditunda, atau gagal secara berurutan. Dan kolom yang ditunjuk nomor 14 menunjukkan catatan dari laporan yang telah selesai ditangani.

Pada bagian laporan yang ditolak akan menampung laporan dengan status '*Not Approved*'. Cara membaca bagian ini sama dengan cara membaca pada dua bagian di atasnya. Elemen yang ditunjuk nomor 15 menampilkan jumlah laporan yang ditolak pengguna. Kolom yang ditunjuk nomor 16 dan nomor 17 menunjukkan tanggal terakhir diubahnya status atau catatan laporan dan nama laporan secara berurutan. Kolom yang ditunjuk nomor 18 menunjukkan tombol yang bertuliskan nama pelapor dari laporan yang ditolak. Tombol pada kolom yang ditunjuk nomor 17 dapat mengalihkan pengguna ke laman '*report-detail*' dari laporan terkait, sedangkan tombol pada kolom yang ditunjuk nomor 18 dapat mengalihkan pengguna ke laman '*user-reports*' dari pelapor terkait. Dan kolom yang ditunjuk nomor 19 menunjukkan catatan dari laporan yang ditolak.

Terdapat juga laman serupa yang ditujukan untuk melihat status laporan yang sedang ditangani, laman tersebut adalah laman *'progress-subscribed'*. Penjelasan mengenai pengembangan laman tersebut akan dijabarkan pada lampiran sub bab A.13.

3.4.11. Pengembangan Laman *'unit-detail'*

Laman bernama *'unit-detail'* ini dapat diakses dengan URL 127.0.0.1:8000/unit/<int:pk>/. Dimana <int:pk> adalah kunci primer (*primary key*) dari unit terkait dalam bentuk bilangan bulat (*integer*). Laman ini ditujukan untuk mengakomodasi aktivitas 2 fitur 6 “Mengelola Organisasi” mengenai pengelolaan anggota pada unit terkait. Laman ini menggunakan fungsi *'UnitDetail()'* sebagai *view*.

Fungsi *'UnitDetail()'* ditulis penulis di berkas *user/views.py*. Fungsi ini akan mengumpulkan semua anggota unit ini beserta pekerjaan anggota tersebut dan kepala unit untuk ditampilkan. Fungsi ini juga mengumpulkan semua pengguna yang tidak bekerja di unit apapun agar bisa dijadikan pilihan bila unit ini membutuhkan anggota baru. Selain itu fungsi ini juga menggunakan metode *'subordinate_list()'* yang telah dibuat pada kelas *'Unit'* untuk mengumpulkan unit anak buah dari unit ini. Lalu fungsi *'UnitDetail()'* juga mengumpulkan seluruh unit kecuali unit ini sendiri, kemudian dikurangi dengan unit anak buah yang telah dikumpulkan sebelumnya agar bisa dijadikan pilihan sebagai unit atasan dari unit ini bila unit ini ingin mengganti unit atasannya. Fungsi *'UnitDetail()'* akan ditampilkan melalui *template* di berkas *user/unit_detail.html*.

Fungsi *'UnitDetail()'* menerima data dari pengguna dengan metode POST (*request.POST*). Secara garis besar terdapat 3 kelompok data yang dikirim pengguna. Kelompok data pertama adalah data mengenai unit atasan dari suatu unit. Kelompok data kedua adalah mengenai kepala dari suatu unit. Dan kelompok data terakhir adalah mengenai nama dan pekerjaan dari anggota unit.

Berkenaan dengan kelompok data pertama, bila tingkatan (*level*) dari suatu unit lebih besar dari satu, berarti unit tersebut memiliki unit atasan (*superior unit*). Unit yang memiliki unit atasan dapat mengubah atasannya dengan unit lainnya. Namun atasan barunya tidak boleh unit itu sendiri maupun unit bawahannya untuk mencegah terjadinya jerat (*loop*) dalam hierarki organisasi dan agar dapat memberi tingkatan (*level*) pada setiap unit. Bila unit atasan baru dapat membuat jerat pada organisasi, pengubahan tidak akan disimpan dan terdapat pesan peringatan. Namun bila unit atasan baru sesuai syarat akan, pengaturan akan disimpan dan muncul pesan umpan balik (*feedback*) yang memberi tahu bahwa unit atasan telah diganti. Untuk mencegah

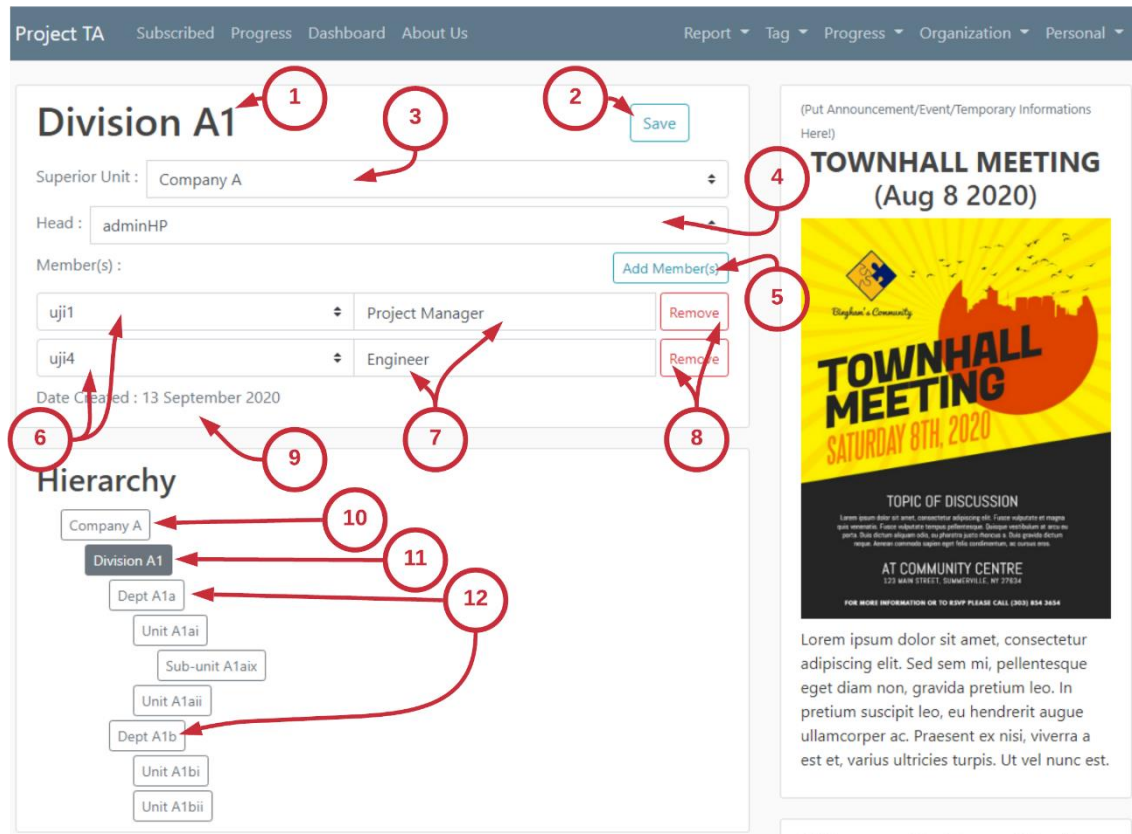
terjadinya jerat pada organisasi, penulis telah berusaha membatasi pilihan unit atasan baru seperti yang dijabarkan pada dua paragraf sebelum ini.

Berkenaan dengan kelompok data kedua, bila suatu unit tidak memiliki kepala unit, akan muncul pesan peringatan untuk mengatur kepala unit untuk unit tersebut. Bila unit tersebut memiliki kepala unit dan data unggahan pengguna berisi kepala unit yang sama, tidak akan terjadi perubahan pada pengaturan unit. Sedangkan bila kepala unit baru yang diunggah berbeda dengan kepala unit yang lama, maka kepala unit yang lama akan diberhentikan dari posisi kepala unit untuk unit tersebut dan kepala unit baru akan mengisi posisi tersebut. Pada log karir juga akan dicatatkan mengenai pemberhentian dan pengisian posisi tersebut.

Berkenaan dengan kelompok data ketiga, akan diterima data yang diunggah pengguna menggunakan metode POST. Pertama data mengenai anggota dan pekerjaan tersebut akan diperiksa. Bila salah satu dari data anggota maupun pekerjaan kosong, isian tersebut akan ditolak dan muncul pesan umpan balik. Sama halnya bila ada masukan ganda seperti contoh di atas, terdapat masukan bahwa pengguna XX diberi pekerjaan sebagai YY dan terdapat masukan setelahnya bahwa pengguna XX diberi pekerjaan ZZ. Karena pengguna XX hanya bisa mengerjakan satu pekerjaan di satu waktu, maka masukan yang belakangan (pengguna XX diberi pekerjaan ZZ) akan ditolak, dan akan muncul pesan peringatan.

Bila kelompok data ketiga telah dibersihkan dari ketidak selarasan seperti yang telah disebutkan, secara garis besar data masukan dibagi menjadi tiga tipe. Tipe pertama yaitu pengguna yang sebelumnya tidak ada di unit tersebut, namun kemudian menjadi anggota di unit tersebut. Hal tersebut berarti anggota baru ditambahkan, maka akan dibuat log karir untuk penambahan anggota tersebut. Tipe kedua yaitu anggota yang sebelumnya ada di unit tersebut, namun kemudian tidak lagi ada di unit tersebut. Hal tersebut berarti anggota terkait dilepastugaskan dari unit tersebut, maka ditambahkan tanggal selesai pada log karir mengenai pekerjaan anggota tersebut di unit terkait. Hal tersebut menandakan pekerjaan pengguna tersebut di unit tersebut telah usai. Dan tipe terakhir adalah anggota yang awalnya ada di unit tersebut, dan kemudian tetap ada di unit tersebut. Tipe ini dapat dibagi menjadi dua lagi, sub tipe pertama anggota tersebut tetap di unit tersebut dan pekerjaannya tidak berubah. Pada kasus sub tipe ini tidak ada log karir yang ditambahkan atau berubah. Pada sub tipe kedua, anggota tersebut tetap ada di unit tersebut namun pekerjaannya berubah. Pada sub tipe kedua ini akan ditambahkan tanggal selesai pada log karir mengenai pengguna tersebut untuk pekerjaan lama dan dibuat log karir baru untuk pengguna tersebut untuk pekerjaan baru di unit yang sama.

Laman *'unit-detail'* juga dilengkapi dengan perintah keamanan. Pertama diperiksa apakah pengguna sudah masuk (*login*). Bila belum pengguna akan dialihkan ke laman masuk. Kedua diperiksa apakah pengguna memiliki otoritas atas fitur 6 “Mengelola Organisasi”. Bila pengguna tidak memiliki otoritas, pengguna akan dialihkan ke laman *'home'*.



Gambar 3.33 Cuplikan layar laman *'unit-detail'* dengan petunjuk

Pada laman *'unit-detail'* ditunjukkan nama dari unit yang sedang diperlihatkan detailnya pada elemen yang ditunjuk nomor 1 pada gambar 3.33. Nama unit atasan dan kepala unit dapat dilihat dan diubah pada elemen yang ditunjuk oleh nomor 3 dan nomor 4 secara berurutan. Nama anggota dan pekerjaannya dapat dilihat dan diubah pada elemen yang ditunjuk nomor 6 dan nomor 7 secara berurutan. Bila ingin menambah anggota baru, dapat dipencet tombol yang ditunjuk nomor 5 dan akan muncul kolom isian kosong seperti elemen yang ditunjuk nomor 6, nomor 7, dan sebuah tombol seperti yang ditunjuk nomor 8. Tombol bertuliskan *'Remove'* seperti yang ditunjuk oleh nomor 8 dapat dipencet untuk menghapus baris mengenai anggota tertentu. Bila pengelolaan unit terkait telah selesai dilakukan, pengguna harus memencet tombol bertuliskan *"Save"* yang ditunjuk nomor 2 untuk menyimpan pengelolaan. Tanggal dibuatnya unit dapat dilihat dari elemen yang ditunjuk nomor 9.

Pada seksi konten (*content section*) di bawah, dapat terlihat posisi dari unit terkait di organisasi pada tombol yang ditunjuk nomor 11. Tombol unit atasan dari unit terkait akan terlihat di atas dan kiri tombol unit terkait seperti tombol yang ditunjuk nomor 10. Dan tombol unit bawahan dari unit terkait akan terlihat di bawah dan kanan tombol unit terkait. Sebagai contoh tombol yang ditunjuk nomor 12 adalah unit bawahan langsung dari unit terkait. Hal tersebut dapat terlihat dari posisi tombol unit bawahan. Semakin ke kanan, semakin rendah tingkatan unit tersebut.

Laman '*unit-detail*' dari suatu objek '*Unit*' dapat diakses melalui laman '*unit-list*'. Laman '*unit-list*' akan berisi tombol yang bertuliskan nama objek '*Unit*' beserta hierarkinya. Penjelasan mengenai pengembangan laman tersebut akan dijabarkan pada lampiran sub bab A.14.

3.4.12. Pengembangan Laman '*auth-detail*'

Laman bernama '*auth-detail*' ini dapat diakses dengan URL 127.0.0.1:8000/auth/. Laman ini ditujukan untuk mengakomodasi aktivitas pada fitur 7 "Mengelola Otorisasi" mengenai pengelolaan otorisasi dari suatu fitur. Laman ini menggunakan fungsi '*AuthDetail()*' sebagai *view*.

Fungsi '*AuthDetail()*' ditulis penulis di berkas *user/views.py*. Fungsi ini akan mengumpulkan semua objek dengan model '*Auth*' berdasarkan fiturnya dan berdasarkan jenisnya (objek '*Auth*' untuk pengguna tertentu, untuk unit tertentu, dan untuk tingkatan (*level*) tertentu). Fungsi '*AuthDetail()*' akan ditampilkan melalui *template* di berkas *user/auth_detail.html*.

Fungsi '*AuthDetail()*' menerima data dari pengguna dengan metode POST (*request.POST*). Secara garis besar data yang dikirim pengguna dapat dibagi menjadi 3 kelompok. Kelompok pertama adalah data mengenai objek '*Auth*' yang sebelumnya tidak ada, namun ada pada kiriman data. Kelompok kedua adalah data mengenai objek '*Auth*' yang semula ada, namun tidak ada pada kiriman data. Dan kelompok terakhir adalah data mengenai objek '*Auth*' yang semula ada, dan tetap ada pada kiriman data.

Data pada kelompok pertama menunjukkan objek '*Auth*' baru yang ditambahkan. Maka dari itu akan dibuat objek '*Auth*' yang baru. Data pada kelompok kedua menunjukkan objek '*Auth*' yang telah ada (*existing*) yang dihapus. Maka dari itu objek '*Auth*' terkait akan dihapus. Data pada kelompok ketiga menunjukkan objek '*Auth*' yang telah ada (*existing*) akan tetap ada. Tidak diberi perintah lebih lanjut pada data tersebut.

Laman *'auth-detail'* juga dilengkapi dengan perintah keamanan. Pertama diperiksa apakah pengguna sudah masuk (*login*). Bila belum pengguna akan dialihkan ke laman masuk. Kedua diperiksa apakah pengguna memiliki otoritas atas fitur 7 “Mengelola Otoritas”. Bila pengguna tidak memiliki otoritas, pengguna akan dialihkan ke laman *'home'*. Dengan memiliki otoritas atas fitur 7 pengguna dapat menambahkan otoritas fitur lainnya pada dirinya. Sehingga memiliki otoritas pada fitur 7 sama saja dengan memiliki kemampuan untuk memiliki otoritas atas seluruh fitur.

Pada laman *'auth-detail'* akan ditampilkan seluruh fitur. Nama dari fitur terkait ditunjukkan pada elemen yang ditunjuk nomor 2, nomor 2*, nomor 2**, nomor 2***, nomor 2**** pada gambar 3.34 dan seterusnya. Pengguna dapat menggulirkan (*scroll*) laman untuk melihat pengaturan otorisasi untuk fitur lainnya. Bila suatu fitur tidak diatur otoritasnya, pada contoh kasus ini seperti fitur *'Create a Report'* dan *'View Reports'* yang ditunjuk nomor 2 dan nomor 2* secara berurutan, maka semua pengguna dapat mengakses fitur tersebut. Bila hal itu terjadi akan muncul peringatan seperti elemen yang ditunjuk nomor 3 dan nomor 3*. Agar sebuah fitur tidak dapat diakses oleh semua pengguna, pengguna harus menambahkan paling tidak satu pengguna (*user*), unit, maupun tingkatan (*level*) dengan memencet tombol yang ditunjukkan nomor 4, nomor 5, atau nomor 6 secara berurutan dan memilih objek “*User*”, “*Unit*”, atau bilangan tingkatan yang ingin dipilih.

Seperti pada kasus fitur 3 yang ditunjuk nomor 2**, telah ditambahkan beberapa pengguna. Dapat terlihat dari kolom pilihan yang ditunjuk nomor 7. Sehingga hanya kedua pengguna tersebut yang dapat mengakses fitur 3. Bila ingin menghapus otoritas kedua pengguna tersebut dari fitur 3, pengguna saat memencet tombol bertuliskan *'Remove'* yang ditunjuk nomor 10.

Kasus pada fitur 4 yang ditunjuk nomor 2***, menunjukkan bahwa otoritas dapat diberikan pada suatu unit. Hal tersebut terlihat dari adanya kolom pilihan unit seperti yang ditunjuk oleh nomor 8. Sama halnya dengan kasus pada fitur 5 yang ditunjuk nomor 2****, menunjukkan bahwa otoritas dapat diberikan kepada suatu tingkatan. Hal tersebut terlihat dari adanya kolom pilihan tingkatan seperti yang ditunjuk nomor 9. Dan juga dapat terlihat kolom pilihan untuk unit dan level dapat dihapus dengan memencet tombol bertuliskan *'Remove'* yang ditunjuk nomor 10 dan nomor 10*.

Pemberian otoritas dapat diberikan kepada lebih dari satu jenis objek. Seperti pada kasus pada fitur 4 yang ditunjuk nomor 2***, terlihat kombinasi otoritas yang diberikan kepada pengguna tertentu dan unit tertentu. Sama halnya pada kasus pada fitur 5 yang ditunjuk nomor

2****, terlihat kombinasi otoritas yang diberikan kepada pengguna tertentu dan tingkatan tertentu. Dapat pula dibuat kombinasi otoritas unit tertentu dan tingkatan tertentu maupun kombinasi ketiganya. Semua dapat dikelola sesuai dengan kebutuhan perusahaan.

The screenshot shows a web application interface for 'Manage Authorization'. The top navigation bar includes 'Project TA', 'Subscribed', 'Progress', 'Dashboard', 'About Us', and a dropdown menu with 'Report', 'Tag', 'Progress', 'Organization', and 'Personal'. The main content area is titled 'Manage Authorization' and contains several sections, each with numbered annotations (1-10) indicating specific features or actions:

- Create A Report**: Includes fields for 'User', 'Unit', and 'Level'. Annotations 1, 2, 3, 4, 5, and 6 point to the 'Save' button and the respective fields. A note below states: '* Free to be accessed by anyone. Please add at least an auth to limit the access'.
- View Reports**: Includes fields for 'User', 'Unit', and 'Level'. Annotations 2* and 3 point to the fields. A note below states: '* Free to be accessed by anyone. Please add at least an auth to limit the access'.
- Create A Tag**: Includes fields for 'User', 'Unit', and 'Level'. Annotations 7 and 10 point to the 'Remove' and 'Add User(s)' buttons respectively. The 'User' field contains 'admin' and 'uji3'.
- Take A Report**: Includes fields for 'User', 'Unit', and 'Level'. Annotations 7*, 8, and 10* point to the 'Remove' button, the 'Unit' field, and the 'Add User(s)' button respectively. The 'User' field contains 'uji4' and the 'Level' field contains '2'.
- View User Activities**: Includes fields for 'User', 'Unit', and 'Level'. Annotations 2****, 7**, and 9 point to the 'Remove' button, the 'Unit' field, and the 'Level' field respectively. The 'User' field contains 'admin', the 'Unit' field contains 'Unit A1aii', and the 'Level' field contains 'Dept A1b'.

On the right side of the interface, there are two event posters:

- Townhall Meeting (Aug 8 2020)**: A yellow poster for a townhall meeting on Saturday 8th, 2020, at the Community Centre. The topic of discussion is 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sem mi, pellentesque eget diam non, gravida pretium leo. In pretium suscipit leo, eu hendrerit augue ullamcorper ac. Praesent ex nisi, viverra a est et, varius ultricies turpis. Ut vel nunc est.'
- INTERNAL BADMINTON SPARRING (Aug 17 2020)**: A blue poster for an internal badminton sparring event on August 17, 2020, titled 'THE FRESHMAN'. It features an illustration of a hand holding a racket and a shuttlecock. The text 'SINCE 2012' is at the bottom.

Gambar 3.34 Cuplikan layar laman 'auth-detail' dengan petunjuk

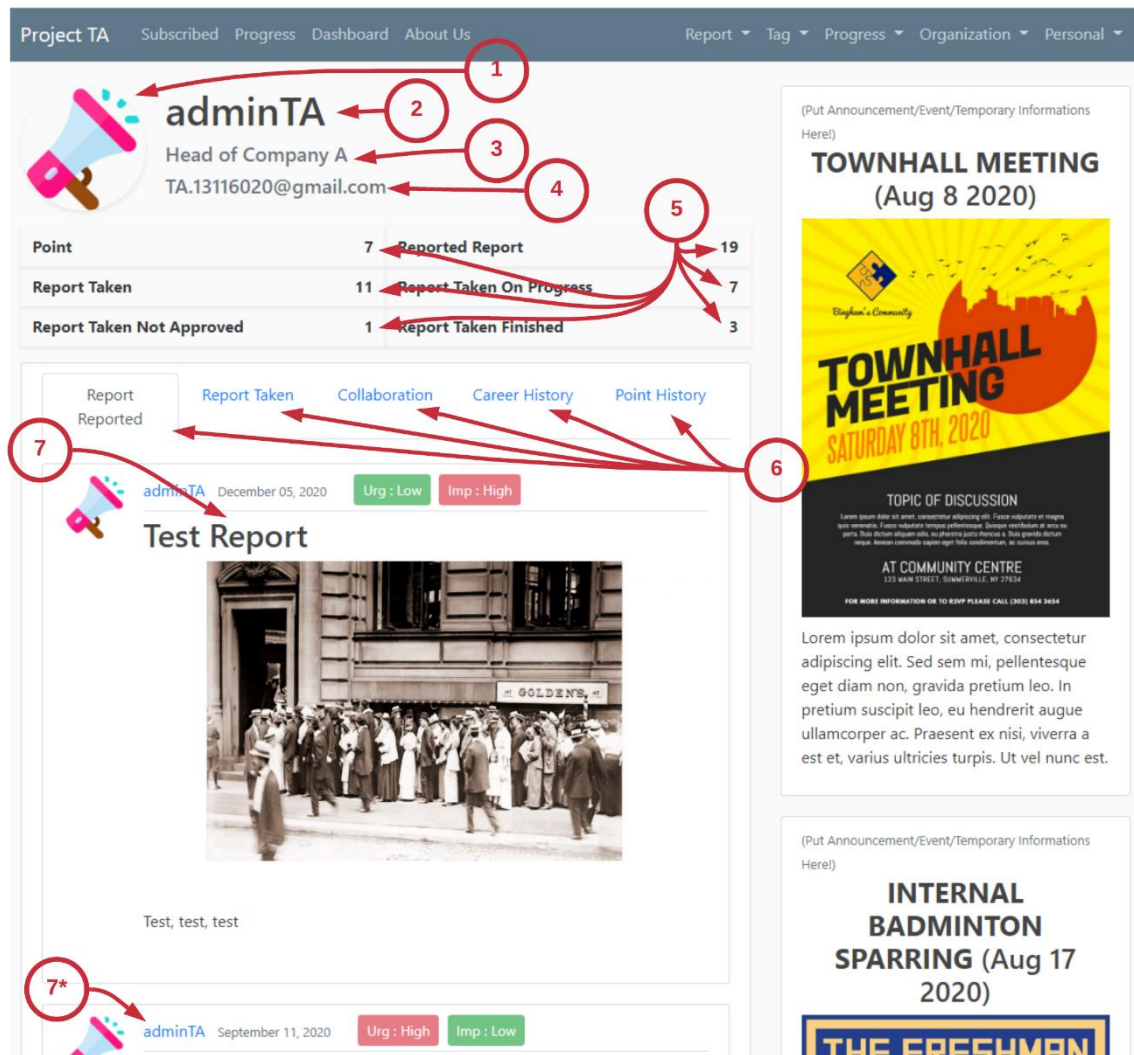
3.4.13. Pengembangan Laman ‘user-reports’

Laman bernama ‘user-reports’ ini dapat diakses dengan URL 127.0.0.1:8000/user/<str:username>/. Dimana <str:username> adalah nama pengguna (*username*) dari pengguna terkait dalam bentuk string. Laman ini menjadi laman yang dituju bila kita memencet tombol yang bertuliskan nama pengguna dari suatu pengguna. Laman ‘user-reports’ ditujukan untuk mengakomodasi aktivitas 6 fitur 2 “Melihat Laporan” dan aktivitas 1 fitur 5 “Melihat Aktivitas Pengguna” mengenai penglihatan laporan yang dibuat pengguna tertentu. Laman ini menggunakan kelas ‘UserReportListView’ sebagai *view*.

Kelas ‘UserReportListView’ ditulis penulis di berkas report/views.py. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) ListView sehingga menerima pewarisan (*inheritance*) dari kelas ListView. Kelas UserReportListView akan memunculkan semua objek ‘Report’ yang dilaporkan oleh pengguna tertentu. Selanjutnya sejumlah objek ‘Report’ akan disusun berdasarkan berdasarkan tanggal laporannya, kemudian kemendesakannya (*urgency*), dan yang terakhir berdasarkan kepentingannya (*importance*). Kelas ini akan ditampilkan melalui *template* di berkas report/user_reports.html.

Laman ‘user-reports’ juga dilengkapi dengan perintah keamanan. Digunakan LoginRequiredMixin untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga UserPassesTestMixin yang dijelaskan dengan metode test_func() untuk memastikan bahwa laman tersebut terkait pengguna yang akan mengakses atau pengguna memiliki otoritas atas fitur 2 atau fitur 5 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari LoginRequiredMixin pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari UserPassesTestMixin tidak dipenuhi, pengguna akan diarahkan ke laman ‘user-taken’ terkait pengguna yang laman ‘user-reports’-nya ingin diakses menggunakan metode handle_no_permission().

Tampilan laman ‘user-reports’ memiliki kemiripan dengan laman ‘all-reports’ karena kedua laman ini bertujuan untuk menunjukkan laporan. Pada laman ‘user-reports’ setiap laporan akan ditampilkan pada satu seksi konten (*content section*) dengan penjelasan yang sama dengan penjelasan isi seksi konten pada sub sub bab 3.4.8 mengenai laman ‘all-reports’. Contoh laporan yang dilaporkan pengguna terkait pada laman ini terlihat pada elemen yang ditunjuk oleh nomor 7 dan nomor 7* pada gambar 3.35 secara berurutan.



Gambar 3.35 Cuplikan layar laman 'user-reports' dengan petunjuk

Selain menampilkan laporan yang dilaporkan oleh pengguna terkait, laman 'user-reports' juga menampilkan data pribadi dan data statistik mengenai suatu pengguna. Data pribadi pengguna yang laman 'user-reports'-nya dibuka yang ditampilkan adalah gambar pengguna, nama pengguna (*username*), dan posisi pengguna melalui elemen yang ditunjuk nomor 1, nomor 2, dan nomor 3. Yang dimaksud dengan posisi adalah nama pekerjaan dan unit tempat pengguna bekerja.

Alamat surel pengguna terkait juga ditampilkan pada elemen yang ditunjuk nomor 4 bila pengguna yang melihat memiliki tingkatan yang lebih tinggi (ditunjukkan dengan bilangan tingkatan yang rendah) atau setara. Serupa halnya alamat surel, data statistik pengguna terkait seperti jumlah poin, jumlah laporan yang dilaporkan, jumlah laporan yang diambil, jumlah laporan yang sedang ditangani, jumlah laporan yang ditolak, dan jumlah laporan yang telah diselesaikan akan ditampilkan pada elemen-elemen yang ditunjuk nomor 5 bila pengguna yang

melihat memiliki tingkatan yang lebih tinggi (ditunjukkan dengan bilangan tingkatan yang rendah).

Selain informasi yang telah dijabarkan di atas, laman '*user-reports*' juga memiliki tombol-tombol yang dibuat untuk memudahkan pengguna saat mencari informasi yang berkaitan dengan pengguna yang ingin dipelajari. Tombol-tombol yang ditunjuk nomor 6 akan mengalihkan pengguna ke laman '*user-reports*', laman '*user-taken*', laman '*user-collab*', laman '*user-career*', dan laman '*user-point*' secara berurutan dari kiri ke kanan.

Terdapat juga empat laman serupa yang ditujukan untuk mengetahui informasi dari suatu pengguna, yaitu laman '*user-taken*', '*user-collab*', '*user-career*', dan '*user-point*'. Penjelasan mengenai pengembangan laman-laman tersebut akan dijabarkan pada lampiran sub bab A.15, A.16, A.17, dan A.18.

3.5. Perancangan dan pengaturan jaringan

Pada pengerjaan tugas akhir ini penulis memilih menggunakan jaringan WLAN (*Wireless LAN*). Penulis tidak membuat aplikasi web yang terhubung dengan internet karena alasan biaya dan keamanan. Bila ingin membuat aplikasi web terhubung dengan internet, salah satu solusinya adalah dengan menggunakan *virtual private server* (VPS) sebagai peladen (*server*). Terdapat penyedia jasa peladen VPS seperti Heroku yang menyediakan VPS gratis, namun dengan beberapa batasan tertentu. Salah satunya peladen ini tidak dapat menyimpan gambar, sehingga pengguna harus menyewa penyimpanan awan dari penyedia (*vendor*) lainnya seperti produk Amazon Simple Storage Service (Amazon S3) dari Amazon Web Services (AWS). Yang mana produk Amazon S3 adalah produk jasa berbayar.

Karena alasan yang dijabarkan di atas, penulis memilih menjalankan aplikasi web pada jaringan WLAN. Dengan menggunakan jaringan Wi-Fi di tempat penulis dan komputer lipat penulis berada, penulis dapat menghubungkan aplikasi web dengan perangkat yang terhubung dengan jaringan Wi-Fi tersebut. Pada kasus ini komputer lipat penulis akan menjadi peladen (*server*) dari aplikasi web ini.

Walaupun pada penelitian tugas akhir aplikasi web ini dihubungkan ke jaringan WLAN, pada pengimplementasiannya aplikasi web ini dapat dihubungkan ke jaringan internet, jaringan WLAN, maupun jaringan lainnya sesuai dengan keinginan, kebutuhan, dan kondisi dari perusahaan atau organisasi

Pada sub bab ini akan dijelaskan cara menghubungkan aplikasi web pada jaringan WLAN. Spesifiknya pada kasus ini penulis akan menghubungkan aplikasi web ini dengan

jaringan Wi-Fi di rumah penulis. Pada kasus ini komputer lipat penulis yang berbasis Windows akan menjadi peladen aplikasi web ini.

Pertama pastikan komputer telah terhubung dengan jaringan Wi-Fi yang ingin dihubungkan. Kemudian bukalah *terminal* atau *command prompt* pada komputer. Selanjutnya ketik ‘ipconfig’ (tanpa tanda petik) dan eksekusi perintah. Carilah bagian Wi-Fi seperti gambar 3.36 pada *terminal*, kita akan membutuhkan alamat IPv4.

```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : 
Link-local IPv6 Address . . . . . : fe80::1288:7a38:174d:8000%11
IPv4 Address. . . . . : 192.168.100.5
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.100.1
```

Gambar 3.36 Cuplikan layar bagian WLAN *adapter* Wi-Fi pada *terminal*

Alamat IPv4 yang didapatkan akan ditambahkan ke list `ALLOWED_HOSTS` pada `settings.py` di proyek Django terkait. Dengan menambahkan alamat IPv4 ke list tersebut, alamat tersebut dapat menjadi *host* untuk aplikasi web ini. Bila pengguna masih ingin mengakses aplikasi melalui *localhost*, pengguna dapat menambahkan alamat IP untuk *localhost*, yaitu 127.0.0.1 pada list `ALLOWED_HOSTS`. (Sehingga akan tertulis seperti ini: `ALLOWED_HOSTS = ['192.168.100.5', '127.0.0.1', 'localhost']`).

Setelah mendapatkan alamat IP dari jaringan Wi-Fi dan memasukkannya pada pengaturan aplikasi web, langkah terakhir adalah menjalankan aplikasi web pada jaringan. Pada umumnya penulis menulis perintah “python manage.py runserver” (tanpa tanda petik) untuk menjalankan aplikasi web pada *localhost*. Namun untuk menjalankannya pada jaringan WLAN perintah yang dijalankan akan menjadi “python manage.py runserver 192.168.100.5:8000” (tanpa tanda petik). Pada pengimplementasiannya 192.168.100.5 dapat diganti dengan alamat IPv4 Wi-Fi yang terhubung dengan komputer peladen dan 8000 dapat diganti dengan *port* yang diinginkan.

Aplikasi web telah terhubung dengan jaringan Wi-Fi. Perangkat lain dapat membuka aplikasi web dengan membuka alamat IPv4 dan *port* terkait pada peramban web (*web browser*) di perangkat tersebut. Pada kasus ini perangkat lain dapat masukkan alamat 192.168.100.5:8000. Layanan aplikasi web akan terus ada asalkan *terminal* tempat menjalankannya perintah runserver masih bekerja.

Bab 4

Pengujian dan Analisis

4.1. Skema Pengujian

Aplikasi web yang telah dirancang dan dikembangkan seperti yang dijabarkan pada bab 3 akan diuji pada bab ini. Secara garis besar akan terdapat dua jenis pengujian. Pengujian jenis pertama adalah pengujian dari setiap fitur pada aplikasi web. Pengujian fitur ini mengikuti metodologi yang tertulis pada diagram alir paling kanan pada gambar 3.2. Secara garis besar akan diuji aksesibilitas, interaksi komponen, dan data yang diterima dari pengguna melalui request.POST untuk setiap laman. Sedangkan pengujian jenis kedua adalah pengujian dari kinerja aplikasi web saat terhubung dengan jaringan WLAN. Akan diuji penggunaan dan interaksi aplikasi web pada perangkat selain komputer yang menjadi hos (*host*), seperti ponsel pintar maupun komputer lipat lainnya yang terhubung dengan jaringan Wi-Fi terkait.

4.2. Pengujian Fitur pada Aplikasi Web

Setiap fitur yang telah dirancang pada sub bab 3.3 akan diuji. Pengujian fitur dilakukan dengan menguji setiap laman yang mengakomodasi aktivitas pada fitur tersebut. Pada suatu laman akan dilakukan tiga pengujian bila memungkinkan.

Pertama, bila laman terkait dilengkapi perintah keamanan, akan diuji aksesibilitas laman tersebut. Secara umum akan aksesibilitas akan diuji dengan pengguna yang belum masuk (*login*), pengguna yang telah masuk tapi tidak memiliki otoritas atas fitur tersebut, dan pengguna yang telah masuk dengan otoritas pada fitur tersebut.

Kedua, bila laman terkait memiliki elemen yang muncul pada kondisi tertentu atau elemen yang dapat berinteraksi dengan pengguna, akan diuji kemunculan (muncul atau tidak) serta interaksi dari elemen terkait dengan pengguna. Elemen yang dimaksud dapat berupa tombol, kolom isian, kolom pilihan, ataupun elemen lainnya.

Dan yang terakhir, bila laman terkait dapat mengunggah informasi kepada pengguna melalui metode POST, akan diuji apakah informasi tersebut telah terunggah dan tersimpan pada basis data.

Laman dianggap dapat bekerja dengan baik bila berhasil melewati pengujian tersebut. Bila setiap laman pada suatu fitur telah berhasil melewati pengujian, maka fitur tersebut dianggap dapat bekerja dengan baik. Bila setiap fitur yang ada dapat bekerja dengan baik,

dianggap aplikasi web ini telah bekerja dengan baik pada jaringan lokal (*localhost* / 127.0.0.1). Pengujian mengenai jaringan akan dilaksanakan pada sub bab 4.3.

Teknis pengujian ini akan menggunakan 3 peramban web (*web browser*). Peramban web pertama akan menggunakan Google Chrome. Pada peramban web ini pengguna akan masuk menggunakan akun dengan nama pengguna adminTA. Akun dengan nama pengguna adminTA merupakan akun *superuser* sehingga dapat mengakses seluruh fitur. Pengujian pada peramban web ini menggambarkan perlakuan aplikasi web pada akun dengan otoritas atas fitur yang sedang diuji dan telah masuk.

Peramban web kedua akan menggunakan Google Chrome pada mode *incognito*. Pada peramban web ini pengguna akan masuk menggunakan akun dengan nama pengguna uji2. Akun dengan nama pengguna uji2 akan diatur otoritasnya sehingga tidak bisa memiliki otoritas fitur yang sedang diuji. Pengaturan otoritas akan dilakukan melalui akun adminTA pada peramban web pertama. Pengujian pada peramban web ini menggambarkan perlakuan aplikasi web pada akun tanpa otoritas atas fitur yang sedang diuji dan telah masuk.

Peramban web ketiga akan menggunakan Microsoft Edge. Pada peramban web ini pengguna tidak masuk (*login*) dengan akun apapun. Pengujian pada peramban web ini akan menggambarkan perlakuan aplikasi web pada pengguna yang tidak masuk (*login*).

Pengujian fitur akan dijabarkan secara rinci pada Lampiran B. Hasil dari pengujian menunjukkan seluruh laman dapat bekerja dengan baik pada kasus yang diberikan. Karena semua laman bekerja dengan baik, dianggap semua fitur bekerja dengan baik.

4.3. Pengujian Kinerja Aplikasi Web saat Terhubung dengan Jaringan WLAN

Setelah dilakukan pengujian pada sub bab B mengenai pengujian fitur, didapati bahwa setiap laman bekerja dengan baik. Baik aksesibilitas, interaksi elemennya, maupun pengunggahan datanya bekerja dengan baik. Setelah didapati setiap laman pada suatu fitur bekerja dengan baik, dapat dianggap suatu fitur bekerja dengan baik. Dan dari pengujian sebelumnya semua fitur bekerja dengan baik, dapat dianggap aplikasi web sudah dapat bekerja dengan baik.

Setelah didapati seluruh aplikasi web bekerja dengan baik, maka di sub bab ini akan diuji kinerja aplikasi web pada perangkat yang terhubung pada jaringan WLAN. Jaringan WLAN pada kasus ini adalah jaringan Wi-Fi rumah penulis yang telah diatur pada sub bab 3.5.

Pada pengujian jaringan ini, akan diuji kinerja aplikasi web pada perangkat yang terhubung dengan jaringan terkait. Perangkat yang akan dihubungkan dan diuji adalah sebuah

ponsel pintar dan sebuah komputer (selain hos). Sedangkan kinerja yang diuji akan berupa kinerja pada sampel beberapa laman yang memiliki interaksi elemen dan pengunggahan data.

Secara spesifik sampel laman tersebut berupa laman '*login*', '*report-create*', '*report-detail*', dan '*logout*'. Pada kedua laman tersebut akan diuji aktivitas masuk ke dalam aplikasi web (*login*), membuat laporan, mengambil laporan, mengubah status laporan, memberi kolaborasi pada laporan tersebut, dan keluar dari aplikasi web (*logout*). Pengujian aktivitas yang disebutkan tersebut menyimulasikan kegiatan antar pengguna pada aplikasi web.

Pada kasus ini pengguna yang menggunakan ponsel pintar akan masuk menggunakan akun dengan nama pengguna adminHP, sedangkan pengguna yang menggunakan komputer (selain hos) akan masuk menggunakan akun dengan nama pengguna staffTekProd. Selain itu untuk tujuan kemudahan pengujian, pada kasus pengujian ini akan diberikan otoritas atas seluruh fitur kepada kedua pengguna tersebut.

Pengujian kinerja aplikasi pada perangkat yang terhubung dengan jaringan WLAN akan dijabarkan secara rinci pada Lampiran C. Hasil dari pengujian menunjukkan aplikasi web dapat bekerja dengan baik pada perangkat-perangkat yang terhubung dengan jaringan WLAN terkait. Karena aplikasi web dapat bekerja dengan baik pada kedua perangkat tersebut, maka dapat dianggap aplikasi web dapat bekerja dengan baik bila dihubungkan dengan jaringan WLAN.

Bab 5

Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan perancangan, pengembangan, dan pengujian aplikasi web yang telah dijabarkan pada bab 3 dan bab 4, maka dapat disimpulkan bahwa:

1. Telah dibuat aplikasi web yang mengakomodasi yang mengakomodasi kegiatan pelaporan, pengarsipan, dan manajemen tugas untuk pihak yang terlibat pada lingkungan produksi dan pihak lain yang terkait.
2. Telah dibuat sekumpulan basis data yang menyimpan data secara terstruktur dan telah dibuat laman yang dapat mengakomodasi pengambilan (*retrieving*) data yang memudahkan pengguna.
3. Telah disetel dan dihubungkan aplikasi web dengan jaringan WLAN sehingga aplikasi web dapat diakses oleh perangkat yang terhubung dengan jaringan WLAN terkait.
4. Telah dibuat laman dan basis data yang menunjang skema gamifikasi untuk memberi insentif kepada pengguna aplikasi web yang berpartisipasi.
5. Telah dibuat skema dan laman seperti pada fitur 6 “Mengelola Organisasi” dan fitur 7 “Mengelola Otoritas” untuk menunjang fungsi skema utama (pelaporan, pengarsipan, dan manajemen tugas) dan skema insentif.

Bila pembaca ingin melihat dan mempelajari kode yang telah dikembangkan oleh penulis, pembaca dapat mengunjungi repositori kode mengenai projek penelitian tugas akhir ini di tautan <https://github.com/geoffrey-wen/TA>.

5.2. Saran

Bila ingin mengembangkan aplikasi web ini lebih lanjut, terdapat beberapa saran dari penulis seperti sebagai berikut:

1. Bila perusahaan ingin menggunakan aplikasi web dalam jangka panjang, dapat dikembangkan basis data lebih lanjut dengan memigrasi data dari basis data sekarang (yang menggunakan SQLite) ke basis data berbasis SQL yang dapat menampung lebih banyak data seperti PostgreSQL, My SQL, atau basis data lainnya.
2. Bila perusahaan memiliki beberapa *plant*, jaringan yang digunakan dapat diganti dari menggunakan WLAN menjadi menggunakan internet.

3. Dapat dilakukan penelitian pengalaman dan interaksi pengguna (*UI/UX research*) lebih dalam untuk menyesuaikan rancangan antarmuka aplikasi web.
4. Dapat dikembangkan versi aplikasi web yang menggunakan bahasa Indonesia atau bahasa yang diinginkan perusahaan.
5. Seperti yang kita ketahui terdapat cukup banyak topik tugas akhir mengenai *Internet of Things (IoT)* dari Laboratorium Teknik Produksi Teknik Mesin. Bila ingin mengintegrasikan perangkat lunak dari tugas akhir yang lain, dapat dikembangkan pengintegrasian dengan menggunakan *Open API*.

Daftar Pustaka

- [1] C. Giffi, A. V. Roth, B. Gangula , High-performing Manufacturers, Where They Play and How They Win, *Deloitte*, 29 November 2016 (<https://www2.deloitte.com/us/en/insights/industry/manufacturing/understanding-high-performance-manufacturing-competitiveness.html>, diakses 1 Desember 2020).
- [2] H. Takayama, *Panduan Peningkatan Produktivitas Industri Pendukung Otomotif dengan Lean Production System*, Direktorat Jenderal Industri Alat Transportasi dan Telematika Departemen Perindustrian, Jakarta, 2009.
- [3] M. Rosoff, Why is Tech Getting Cheaper?, *World Economic Forum*, 16 Oktober 2015 (<https://www.weforum.org/agenda/2015/10/why-is-tech-getting-cheaper/>, diakses 1 Desember 2020).
- [4] S. L. Cisco, K. V. Strong, The Value Added Information Chain. *Information Management Journal*, **33(1)**: 4-15, 1999 (<https://www.sciencedirect.com/science/article/pii/S1877042812034726>, diakses 1 Desember 2020).
- [5] K. Chantarasenar, 7 Abnormal with AM Step 1: Initial Cleaning (Cleaning is Inspection), *Lean TPM*, 16 Agustus 2018 (<https://leantpm.co/2018/08/16/7abnormal-with-am-step-1-initial-cleaning-cleaning-is-inspection/>, diakses 1 Desember 2020).
- [6] O. Kulcu, T. Cakmak, Convergence of the Records Management and Enterprise Content Management in the Digital Environment, *Procedia - Social and Behavioral Science*, **62**: 194-197, 2012 (<https://www.sciencedirect.com/science/article/pii/S1877042812034726>, diakses 1 Desember 2020).
- [7] N. Evans, J. Price, Development of a Holistic Model for the Management of an Enterprise's Information Assets, *International Journal of Information Management*, **54**, 2020 (<https://www.sciencedirect.com/science/article/pii/S0268401219300052>, diakses 1 Desember 2020).
- [8] K. Robsin, K. Plangger, et al., Is It All a Game? Understanding the Principles of Gamification, *Business Horizons*, **58(4)**: 411-420, 2015 (<https://www.sciencedirect.com/science/article/pii/S000768131500035X>, diakses 1 Desember 2020).

- [9] B. F. Skinner, *The Behavior of Organisms: An Experimental Analysis*, Appleton-Century, New York, 1938
(<https://www.sciencedirect.com/science/article/pii/S000768131500035X>, diakses 1 Desember 2020).
- [10] E. L. Thorndike, *The Elements of Psychology*, A.G. Seiler, New York, 1905
(<https://www.sciencedirect.com/science/article/pii/S000768131500035X>, diakses 1 Desember 2020).
- [11] E. T. Higgins, Value from Hedonic Experience and Engagement, *Psychological Review*, **113(3)**: 439—460, 2006
(<https://www.sciencedirect.com/science/article/pii/S000768131500035X>, diakses 1 Desember 2020).
- [12] R. Hunicke, M. LeBlanc, R. Zubek, (2004). MDA: A Formal Approach to Game Design and Game Research, dipresentasikan pada *AAAI Workshop on Challenges in Game AI*, San Jose - California, 25-26 Juli 2004
(<https://www.sciencedirect.com/science/article/pii/S000768131500035X>, diakses 1 Desember 2020).
- [13] M. LeBlanc, Game Design and Tuning Workshop Materials, dipresentasikan pada *the Game Developers Conference*, San Jose - California, 22-26 Maret 2004
(<https://www.sciencedirect.com/science/article/pii/S000768131500035X>, diakses 1 Desember 2020).
- [14] P. Sweetser, P. Wyeth, GameFlow: A Model for Evaluating Player Enjoyment in Games. *Computers in Entertainment (CIE)*, **3(3)**: 1—24, 2005
(<https://www.sciencedirect.com/science/article/pii/S000768131500035X>, diakses 1 Desember 2020).
- [15] M. Lutz, *Python Pocket Reference 4th Edition*, O'Reilly Media, Sebastopol - California, 2009.
- [16] M. Lutz, *Learning Python 4th Edition*, O'Reilly Media, Sebastopol - California, 2009.
- [17] P. Christensson, Web Application Definition, *TechTerms*, 17 Februari 2014
(https://techterms.com/definition/web_application, diakses 1 Desember 2020).
- [18] M. Georgiou, Web App vs. Website: What's the Difference & Which One Is Good For Your Business?, *Imaginovation*, 24 Oktober 2019

(<https://www.imaginnovation.net/blog/website-or-web-app-which-is-good-for-business/>, diakses 1 Desember 2020).

- [19] N. George, *Mastering Django*, Packt Publishing, Birmingham, 2016.
- [20] FAQ: General, *Django Documentation*, Django Software Foundation, Lawrence - Kansas, 2013 (<https://docs.djangoproject.com/en/dev/faq/general/>, diakses 1 Desember 2020).
- [21] Writing Your First Django App, Part 1, *Django Documentation*, Django Software Foundation, Lawrence - Kansas, 2013 (<https://docs.djangoproject.com/en/3.1/intro/tutorial01/>, diakses 1 Desember 2020).
- [22] Models, *Django Documentation*, Django Software Foundation, Lawrence - Kansas, 2013 (<https://docs.djangoproject.com/en/3.1/topics/db/models/>, diakses 1 Desember 2020).
- [23] Model Field Reference, *Django Documentation*, Django Software Foundation, Lawrence - Kansas, 2013 (<https://docs.djangoproject.com/en/3.1/ref/models/fields/>, diakses 1 Desember 2020).
- [24] Templates, *Django Documentation*, Django Software Foundation, Lawrence - Kansas, 2013 (<https://docs.djangoproject.com/en/3.1/topics/templates/>, diakses 1 Desember 2020).
- [25] The Django Template Language, *Django Documentation*, Django Software Foundation, Lawrence - Kansas, 2013 (<https://docs.djangoproject.com/en/3.1/ref/templates/language/>, diakses 1 Desember 2020).
- [26] Writing Views, *Django Documentation*, Django Software Foundation, Lawrence - Kansas, 2013 (<https://docs.djangoproject.com/en/3.1/topics/http/views/>, diakses 1 Desember 2020).
- [27] Class-based Views, *Django Documentation*, Django Software Foundation, Lawrence - Kansas, 2013 (<https://docs.djangoproject.com/en/3.1/topics/class-based-views/>, diakses 1 Desember 2020).
- [28] URL Dispatcher, *Django Documentation*, Django Software Foundation, Lawrence - Kansas, 2013 (<https://docs.djangoproject.com/en/3.1/topics/http/urls/>, diakses 1 Desember 2020).

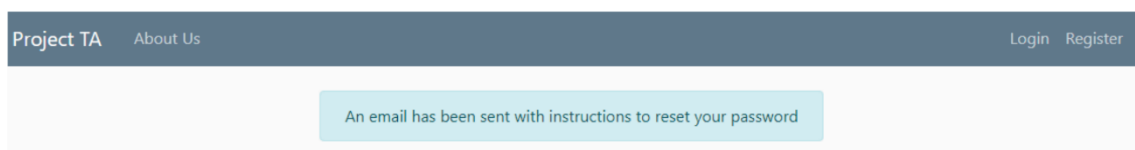
- [29] U. V. Riss, A. Tickayzen, et al., Challenges for Business Process and Task Management, *Journal of Universal Knowledge Management*, **0(2)**: 77-100, 2005 (https://www.researchgate.net/publication/228669449_Challenges_for_business_processes_and_task_management, diakses 1 Desember 2020).
- [30] State Transition Diagrams for To-Do Tasks, *IBM Business Automation Workflow V19.0.0.3 Documentation*, IBM, Armonk - New York (https://www.ibm.com/support/knowledgecenter/SS8JB4_19.x/com.ibm.wbpm.admin.doc/topics/ctasklifecycle_todo.html, diakses 1 Desember 2020).
- [31] Task Lifecycle, *BMC Helix Business Workflows 20.08 Documentation*, BMC, Houston - Texas, 2020 (<https://docs.bmc.com/docs/bwf/task-lifecycle-797321409.html>, diakses 1 Desember 2020).
- [32] Managing the Task Lifecycle, *Camunda Best Practices*, Camunda, Berlin (<https://camunda.com/best-practices/managing-the-task-lifecycle/>, diakses 1 Desember 2020).
- [33] D. J. Anderson, A. Carmichael, *Essential Kanban Condensed*, Blue Hole Press, Seattle - Washington, 2015.
- [34] A. Stellman, J. Greene, *Learning Agile*, O'Reilly Media, Sebastopol - California, 2014.
- [35] Expedite Work with Swimlanes, *Azure DevOps documentation*, Microsoft, Redmond - Washington, 2020 (<https://docs.microsoft.com/en-us/azure/devops/boards/boards/expedite-work?view=azure-devops>, diakses 1 Desember 2020).
- [36] G. A. Donahue, *Network Warrior*, O'Reilly Media, Sebastopol - California, 2011.
- [37] S. Shukla, D. Patel, Data Modelling and Entity Relationship Diagram (ERD) [Kelas CS215 - Web and Database Programming], *Dept of Computer Science, University of Regina*, Musin Gugur 2020 (<https://www.cs.uregina.ca/Links/class-info/215/erd/index.html>, diakses 1 Desember 2020).
- [38] J. Bernat, Crow's Foot Notation [Kelas CS215 - Web and Database Programming], *Dept of Computer Science, University of Regina* (<http://www2.cs.uregina.ca/~bernatja/crowsfoot.html>, diakses 1 Desember 2020).
- [39] A. Sichkarenko, Python [JetBrains Developer Ecosystem Survey 2020], *JetBrains*, 2020 (<https://www.jetbrains.com/lp/devecosystem-2020/python/>, diakses 1 Desember 2020).

Lampiran A

Pengembangan Perangkat Lunak (Lanjutan)

A.1. Pengembangan Laman *'password_reset_done'*

Laman bernama *'password_reset_done'* ini dapat diakses dengan URL `127.0.0.1:8000/password_reset_done/`. Laman ini menjadi respons umpan balik kepada pengguna yang telah melakukan aktivitas 2 fitur “Akun Pengguna” mengenai memasukkan surel bila lupa kata sandi. Laman ini menggunakan *view* `django.view.auth.PasswordResetDoneView`. `PasswordResetDoneView` merupakan kelas *view* bawaan (*built-in*) dari Django, jadi penulis tidak perlu membuat fungsi *view* khusus untuk menunjukkan bahwa tautan penyetelan ulang kata sandi telah dikirim ke kotak masuk surel pengguna. *Template* yang digunakan penulis terdapat di berkas `user/password_reset_done.html`.



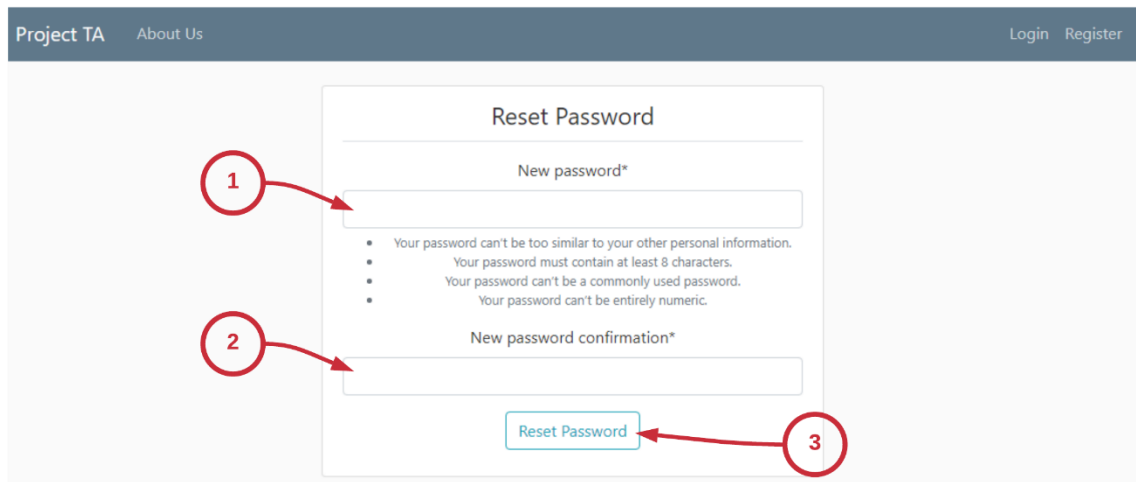
Gambar A.1. Cuplikan layar laman *'password_reset_done'*

A.2. Pengembangan Laman *'password_reset_confirm'*

Laman bernama *'password_reset_confirm'* ini dapat diakses dengan URL `127.0.0.1:8000/password-reset-confirm/<uidb64>/<token>/`. Dimana `<uidb64>` adalah id pengguna yang dikodekan dalam basis 64, sedangkan `<token>` adalah kode yang dibuat untuk memeriksa bahwa tautan ini valid. Laman ini ditujukan untuk mengakomodasi aktivitas 4 dan aktivitas 5 fitur “Akun Pengguna” mengenai masuk ke laman penyetelan kata sandi. Laman ini menggunakan *view* `django.view.auth.PasswordResetConfirmView`. `PasswordResetConfirmView` merupakan kelas *view* bawaan (*built-in*) dari Django, jadi penulis tidak perlu membuat fungsi *view* khusus untuk menyetel ulang kata sandi. *Template* yang digunakan penulis terdapat di berkas `user/password_reset_confirm.html`.

Tautan untuk menyetel ulang kata sandi pada surel yang dikirim akan mengarahkan pada laman ini. Pengguna dapat memasukkan kata sandi baru pada kolom isian yang ditunjuk nomor 1 pada gambar A.2. Selanjutnya pengguna mengkonfirmasi kata sandi tersebut dengan

memasukkan ulang pada kolom isian yang ditunjuk nomor 2. Dan bila pengguna telah yakin dengan kata sandi tersebut, pengguna dapat memencet elemen yang ditunjuk nomor 3.



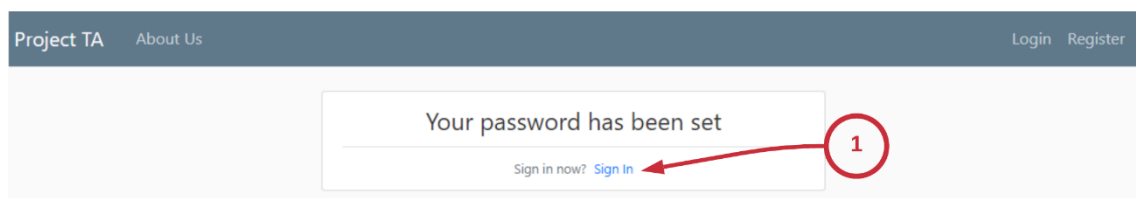
The screenshot shows a web application interface with a dark blue header containing 'Project TA' and 'About Us' on the left, and 'Login' and 'Register' on the right. The main content area is light gray and features a white 'Reset Password' form. The form has two input fields: 'New password*' and 'New password confirmation*'. Between these fields is a list of four password requirements. At the bottom of the form is a blue 'Reset Password' button. Three red circular callouts with arrows point to specific elements: callout '1' points to the 'New password*' input field, callout '2' points to the 'New password confirmation*' input field, and callout '3' points to the 'Reset Password' button.

Gambar A.2 Cuplikan layar laman *'password_reset_confirm'* dengan petunjuk

A.3. Pengembangan Laman *'password_reset_complete'*

Laman bernama *'password_reset_complete'* ini dapat diakses dengan URL `127.0.0.1:8000/password-reset-complete/`. Laman ini menjadi respons umpan balik kepada pengguna yang telah melakukan aktivitas 5 fitur “Akun Pengguna” mengenai masuk ke laman penyetelan kata sandi. Laman ini menggunakan *view* `django.view.auth.PasswordResetCompleteView`. `PasswordResetCompleteView` merupakan kelas *view* bawaan (*built-in*) dari Django, jadi penulis tidak perlu membuat fungsi *view* khusus untuk menunjukkan bahwa kata sandi telah diganti. *Template* yang digunakan penulis terdapat di berkas `user/password_reset_complete.html`.

Setelah tahu bahwa kata sandi telah berhasil diganti, bila pengguna ingin masuk pada aplikasi web, pengguna dapat memencet elemen yang ditunjuk nomor 1 pada gambar A.3.



The screenshot shows the same web application interface as before. The main content area is light gray and features a white box with the message 'Your password has been set'. Below this message is the text 'Sign in now?' followed by a blue 'Sign In' link. A red circular callout with the number '1' and an arrow points to the 'Sign In' link.

Gambar A.3 Cuplikan layar laman *'password_reset_complete'* dengan petunjuk

A.4. Pengembangan Laman *'point-list'*

Laman bernama *'point-list'* ini dapat diakses dengan URL 127.0.0.1:8000/point-list/. Laman ini ditujukan untuk mengakomodasi aktivitas 2 fitur 8 “Mengelola Poin” mengenai penglihatan daftar log poin. Laman ini menggunakan fungsi *'PointHistoryList()'* sebagai *view*.

Fungsi *'PointHistoryList()'* ditulis penulis di berkas *user/views.py*. Fungsi ini akan mengumpulkan semua log poin untuk ditampilkan kepada pengguna. Agar pengguna dapat mencari *tag* yang diinginkan dengan lebih mudah, penulis menambahkan kelas *PointHistoryFilter* yang merupakan subkelas dari kelas *FilterSet* yang berasal dari *package django_filters*. Kelas *PointHistoryFilter* ditulis di berkas *user/filters.py*. *PointHistoryFilter* dapat menyaring tanggal, besar, pemberi, penerima, dan catatan dari suatu objek *'PointHistory'*. Fungsi *'PointHistoryList()'* akan ditampilkan melalui *template* di berkas *user/pointhistory_list.html*.

Laman *'point-list'* juga dilengkapi dengan perintah keamanan. Pertama diperiksa apakah pengguna sudah masuk (*login*). Bila belum pengguna akan dialihkan ke laman masuk. Kedua diperiksa apakah pengguna memiliki otoritas atas fitur 8. Bila pengguna tidak memiliki otoritas, pengguna akan dialihkan ke laman *'home'*.

Pada laman *'point-list'* pengguna dapat mencari log poin yang diinginkan. Kolom yang ditunjuk nomor 11, nomor 12, nomor 13, nomor 14, dan nomor 15 pada gambar A.4 menunjukkan tanggal, besar, penerima, pemberi, dan catatan dari suatu log poin secara berurutan. Satu log poin ditampilkan pada satu baris. Untuk mempermudah pencarian pengguna dapat menggunakan filter yang disediakan pada bagian atas laman. Dengan memasukkan rentang tanggal pada kolom isian yang ditunjuk nomor 1 dan nomor 2, rentang jumlah poin pada kolom isian yang ditunjuk nomor 3 dan nomor 4, mengisi sebagian nama pengguna (*username*) penerima poin dan pemberi poin pada kolom isian yang ditunjuk nomor 5 dan nomor 6, ataupun mengisi sebagian dari catatan pada kolom nomor 7, pengguna dapat menyaring log poin yang ditampilkan. Jumlah dan hasil penjumlahan poin dari log poin yang telah disaring akan ditampilkan pada elemen yang ditunjuk nomor 9 dan nomor 10.

Project TA Subscribed Progress Dashboard About Us Report Tag Progress Organization Personal

Log(s) After

Log(s) Before

Point ≥

Point ≤

Receiver's Username

Writer's Username

Note

Number of Log(s) 7 Sum of Point(s) -11

Date	Point	Receiver	Writer	Note
07Dec20 22:12	-3	adminTA	adminTA	Ditukar dengan bonus bulan Desember
12Nov20 10:58	-5	uji4	adminTA	Terlambat
12Nov20 10:56	4	adminTA	adminTA	Point from report: "Urg-Imp Low-Low", pk = 28
12Nov20 09:35	-3	uji3	adminTA	Ditukar dengan gaji
17Oct20 14:00	-3	uji1	adminTA	penalty
17Oct20 13:58	-4	admin	adminTA	Tukar dengan gaji
17Oct20 13:30	3	adminTA	adminTA	Point from report: "Test Kanban 1", pk = 30

(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING
(Aug 8 2020)

TOWNHALL MEETING
SATURDAY 8TH, 2020

TOPIC OF DISCUSSION

AT COMMUNITY CENTRE
123 MAIN STREET, SUMMITVILLE, NY 27034

FOR MORE INFORMATION OR TO RSVP PLEASE CALL (303) 854 3654

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sem mi, pellentesque eget diam non, gravida pretium leo. In pretium suscipit leo, eu hendrerit augue ullamcorper ac. Praesent ex nisi, viverra a est et, varius ultricies turpis. Ut vel nunc est.

(Put Announcement/Event/Temporary Informations Here!)

INTERNAL BADMINTON SPARRING
(Aug 17 2020)

Gambar A.4 Cuplikan layar laman 'point-list' dengan petunjuk

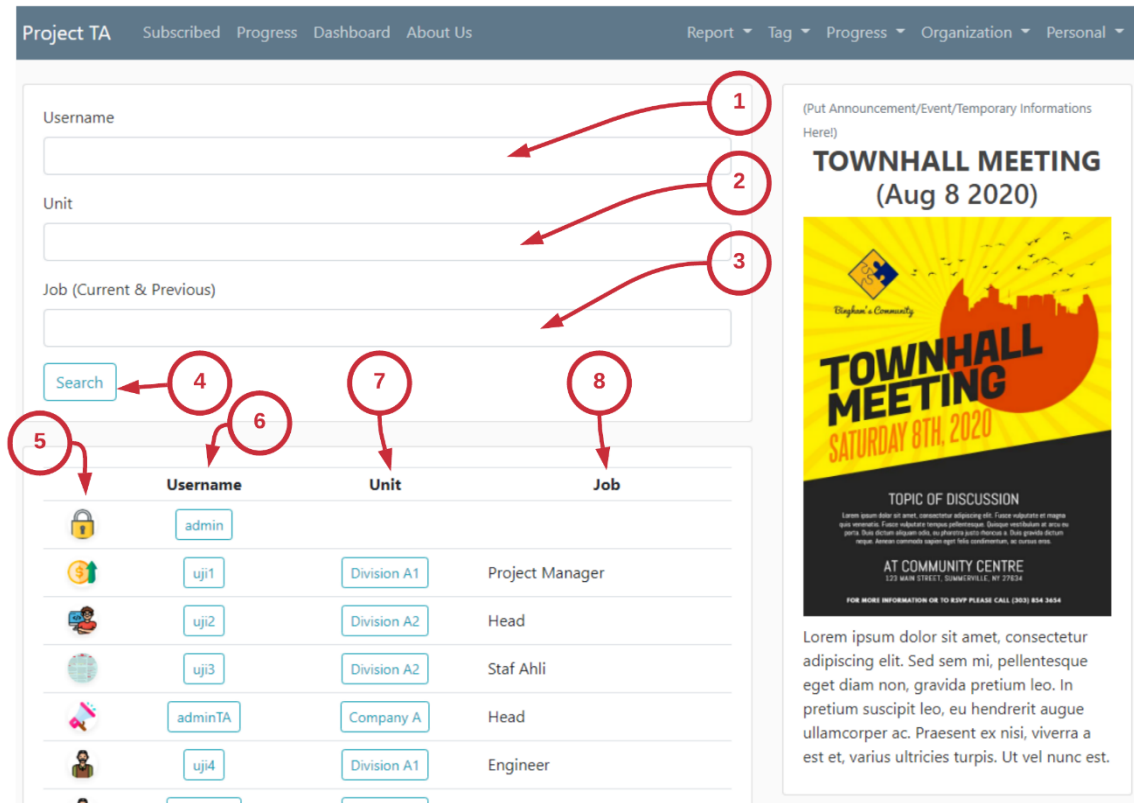
A.5. Pengembangan Laman 'user-list'

Laman bernama 'user-list' ini dapat diakses dengan URL 127.0.0.1:8000/user-list/. Laman ini ditujukan untuk mengakomodasi aktivitas 3 fitur 6 "Mengelola Organisasi" mengenai penglihatan daftar pengguna. Laman ini menggunakan fungsi 'UserList()' sebagai view.

Fungsi 'UserList()' ditulis penulis di berkas user/views.py. Fungsi ini akan mengumpulkan semua pengguna untuk ditampilkan. Agar pengguna dapat mencari pengguna yang diinginkan dengan lebih mudah, penulis menambahkan kelas UserFilter yang merupakan subkelas dari kelas FilterSet yang berasal dari package django_filters. Kelas UserFilter ditulis di berkas user/filters.py. UserFilter dapat menyaring nama pengguna, unit bekerja, dan rekam

jejak pekerjaan dari suatu objek 'User'. Fungsi 'UserList()' akan ditampilkan melalui *template* di berkas user/user_list.html.

Laman 'user-list' juga dilengkapi dengan perintah keamanan. Pertama diperiksa apakah pengguna sudah masuk (*login*). Bila belum pengguna akan dialihkan ke laman masuk. Kedua diperiksa apakah pengguna memiliki otoritas atas fitur 6. Bila pengguna tidak memiliki otoritas, pengguna akan dialihkan ke laman 'home'.



Gambar A.5 Cuplikan layar laman 'user-list' dengan petunjuk

Pada laman 'user-list' pengguna dapat mencari pengguna yang diinginkan. Kolom yang ditunjuk nomor 5, nomor 6, nomor 7, dan nomor 8 pada gambar A.5 menunjukkan gambar profil, tombol yang bertuliskan nama pengguna, tombol yang bertuliskan nama unit sekarang, dan pekerjaan sekarang dari suatu pengguna secara berurutan. Satu pengguna ditampilkan pada satu baris. Bila suatu pengguna tidak bekerja pada unit apapun pada saat ini, bagian kolom yang ditunjuk nomor 7 dan nomor 8 akan kosong seperti pada contoh kasus di gambar A.5. Pada gambar tersebut, pada baris teratas, terdapat pengguna dengan nama pengguna 'admin' yang tidak sedang berada pada unit apapun, sehingga tidak memiliki pekerjaan apapun.

Untuk mempermudah pencarian pengguna dapat menggunakan filter yang disediakan pada bagian atas laman. Dengan memasukkan sebagian nama pengguna, sebagian nama unit, ataupun sebagian nama pekerjaan pada kolom isian yang ditunjuk nomor 1, nomor 2, dan nomor 3 secara berurutan, pengguna dapat menyaring pengguna yang ditampilkan.

A.6. Pengembangan Laman ‘tag-create’

Laman bernama ‘tag-create’ ini dapat diakses dengan URL 127.0.0.1:8000/tag-new/. Laman ini ditujukan untuk mengakomodasi aktivitas 3 fitur 3 “Membuat *Tag*” mengenai pembuatan *tag* baru. Laman ini menggunakan kelas ‘TagCreateView’ sebagai *view*.

Kelas ‘TagCreateView’ ditulis penulis di berkas report/views.py. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) CreateView sehingga menerima pewarisan (*inheritance*) dari kelas CreateView. Kelas CreateView telah menangani pembuatan formulir untuk membuat objek ‘*Tag*’ baru dan fungsi untuk menyimpan objek tersebut. Pada form ini diminta bidang name dan description pada objek ‘*Tag*’. Untuk bidang creator akan dihubungkan dengan objek ‘*User*’ yang terasosiasi dengan pengguna yang sedang membuat tag melalui metode form_valid(). Kelas ini akan ditampilkan melalui *template* di berkas report/tag_form.html.

Laman ‘tag-create’ juga dilengkapi dengan perintah keamanan. Digunakan LoginRequiredMixin untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga UserPassesTestMixin yang dijelaskan dengan metode test_func() untuk memastikan pengguna memiliki otoritas atas fitur 3 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari LoginRequiredMixin pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari UserPassesTestMixin tidak dipenuhi, pengguna akan diarahkan ke laman ‘home’ menggunakan metode handle_no_permission().

Pada laman ‘tag-create’ pengguna dapat memasukkan nama dan deskripsi dari *tag* baru pada kolom isian yang ditunjukkan nomor 1 dan nomor 2 pada gambar A.6 secara berurutan. Bila pengguna telah yakin, pengguna dapat memencet elemen yang ditunjuk nomor 3.

Gambar A.6 Cuplikan layar laman ‘tag-create’ dengan petunjuk

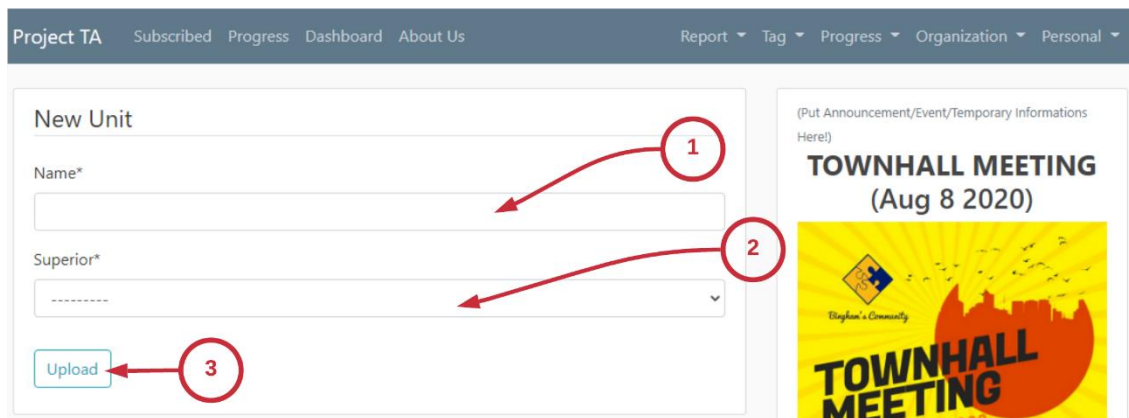
A.7. Pengembangan Laman ‘unit-create’

Laman bernama ‘unit-create’ ini dapat diakses dengan URL 127.0.0.1:8000/unit-new/. Laman ini ditujukan untuk mengakomodasi aktivitas 1 fitur 6 “Mengelola Organisasi” mengenai pembuatan unit baru. Laman ini menggunakan kelas ‘UnitCreateView’ sebagai *view*.

Kelas ‘UnitCreateView’ ditulis penulis di berkas user/views.py. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) *CreateView* sehingga menerima pewarisan (*inheritance*) dari kelas *CreateView*. Kelas *CreateView* telah menangani pembuatan formulir untuk membuat objek ‘Unit’ baru dan fungsi untuk menyimpan objek tersebut. Pada form ini diminta bidang name dan superior pada objek ‘Report’. Kelas ini akan ditampilkan melalui *template* di berkas user/unit_form.html.

Laman ‘unit-create’ juga dilengkapi dengan perintah keamanan. Digunakan *LoginRequiredMixin* untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga *UserPassesTestMixin* yang dijelaskan dengan metode *test_func()* untuk memastikan pengguna memiliki otoritas atas fitur 6 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari *LoginRequiredMixin* pengguna akan dialihkan ke laman

masuk. Sedangkan bila syarat dari `UserPassesTestMixin` tidak dipenuhi, pengguna akan diarahkan ke laman *'home'* menggunakan metode `handle_no_permission()`.



Gambar A.7 Cuplikan layar laman *'unit-create'* dengan petunjuk

Pada laman *'unit-create'* pengguna dapat memasukkan nama dan unit atasan dari unit baru pada kolom isian yang ditunjuk nomor 1 dan 2 pada gambar A.7 secara berurutan. Bila pengguna telah yakin untuk membuat unit baru, pengguna dapat memencet tombol yang ditunjuk nomor 3 untuk membuat objek *'Unit'* baru. Bila objek *'Unit'* baru telah terbuat, pengguna akan diarahkan ke laman *'unit-detail'* dari unit baru tersebut dan pengguna dapat mengatur anggota dari unit tersebut.

A.8. Pengembangan Laman *'point-create'*

Laman bernama *'point-create'* ini dapat diakses dengan URL `127.0.0.1:8000/point-new/`. Laman ini ditujukan untuk mengakomodasi aktivitas 1 fitur 8 “Mengelola Poin” mengenai pembuatan log poin baru. Laman ini menggunakan kelas *'PointHistoryCreateView'* sebagai *view*.

Kelas *'PointHistoryCreateView'* ditulis penulis di berkas `user/views.py`. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) *CreateView* sehingga menerima pewarisan (*inheritance*) dari kelas *CreateView*. Kelas *CreateView* telah menangani pembuatan formulir untuk membuat objek *'PointHistory'* baru dan fungsi untuk menyimpan objek tersebut. Pada form ini diminta bidang *user*, *point*, dan *note* pada objek *'PointHistory'*. Untuk bidang *writer* akan dihubungkan dengan objek *'User'* yang terasosiasi dengan pengguna yang sedang membuat log poin melalui metode `form_valid()`. Kelas ini akan ditampilkan melalui *template* di berkas `user/pointhistory_form.html`.

Laman *'point-create'* juga dilengkapi dengan perintah keamanan. Digunakan `LoginRequiredMixin` untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga `UserPassesTestMixin` yang dijelaskan dengan metode `test_func()` untuk memastikan pengguna memiliki otoritas atas fitur 8 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari `LoginRequiredMixin` pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari `UserPassesTestMixin` tidak dipenuhi, pengguna akan diarahkan ke laman *'home'* menggunakan metode `handle_no_permission()`.

The screenshot shows a web interface for 'Project TA'. The top navigation bar includes links for 'Subscribed', 'Progress', 'Dashboard', and 'About Us', along with dropdown menus for 'Report', 'Tag', 'Progress', 'Organization', and 'Personal'. The main content area is titled 'Write Point History'. It contains three input fields: 'User*' (a dropdown menu), 'Point*' (a text input), and 'Note*' (a large text area). A red circle with the number '1' points to the 'User*' dropdown. A red circle with the number '2' points to the 'Point*' input field. A red circle with the number '3' points to the 'Note*' text area. A red circle with the number '4' points to the 'Upload' button at the bottom left of the form. To the right of the form is a sidebar with a header '(Put Announcement/Event/Temporary Informations Here!)' and a poster for a 'TOWNHALL MEETING (Aug 8 2020)'. The poster includes details about the meeting date, location (Community Centre), and a topic of discussion.

Gambar A.8 Cuplikan layar laman *'point-create'* dengan petunjuk

Pada laman *'point-create'* pengguna dapat memilih nama tujuan diberikan pada kolom pilihan yang ditunjuk nomor 1 pada gambar A.8. Setelah itu pengguna dapat mengisi jumlah poin yang diberikan (atau dikurangi bila angka negatif) dan catatan keterangan pada kolom isian yang ditunjuk nomor 2 dan nomor 3 secara berurutan. Bila pengguna telah yakin untuk memberikan (atau mengurangi) poin kepada suatu pengguna lainnya, pengguna dapat memencet tombol yang ditunjuk nomor 4 untuk membuat objek *'PointHistory'* baru.

A.9. Pengembangan Laman ‘tag-reports’

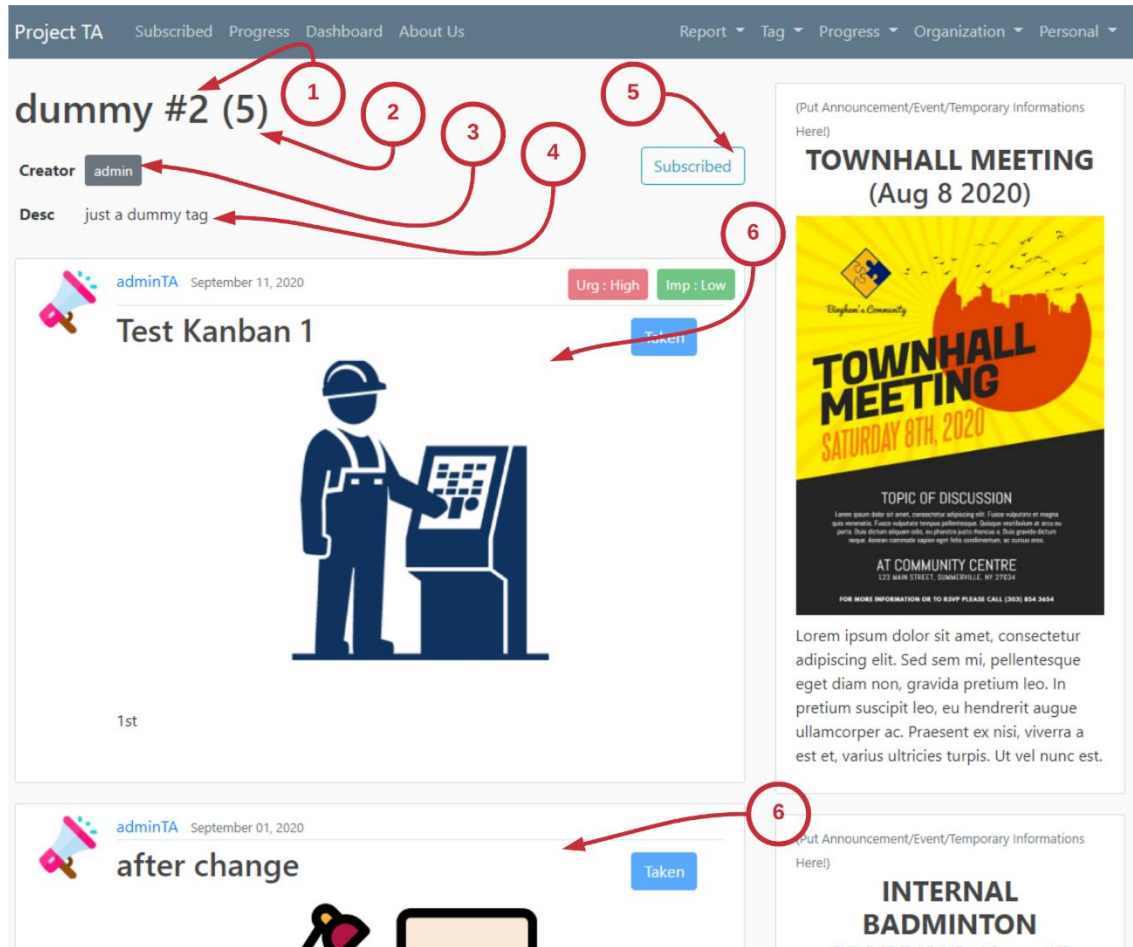
Laman bernama ‘tag-reports’ ini dapat diakses dengan URL 127.0.0.1:8000/tag/<str:tagname>/. Dimana <str:tagname> adalah nama dari *tag* yang ingin diakses dalam bentuk string. Laman ini ditujukan untuk mengakomodasi aktivitas 5 fitur 2 “Melihat Laporan” dan aktivitas 2 fitur 3 “Membuat *Tag*” mengenai penglihatan laman *tag* terkait dan aktivitas 2 fitur 2 “Melihat Laporan” mengenai berlangganan *tag*. Laman ini menggunakan kelas ‘TagReportListView’ sebagai *view*.

Kelas ‘TagReportListView’ ditulis penulis di berkas report/views.py. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) ListView sehingga menerima pewarisan (*inheritance*) dari kelas ListView. Kelas TagReportListView akan memunculkan sejumlah objek ‘Report’ yang terkait dengan *tag* tertentu. Selanjutnya sejumlah objek ‘Report’ akan disusun berdasarkan tanggal laporannya, kemudian berdasarkan kemendesakannya (*urgency*), dan yang terakhir berdasarkan kepentingannya (*importance*). Kelas ini akan ditampilkan melalui *template* di berkas report/tag_reports.html.

Laman ini menerima tindakan (*action*) dari pengguna dengan metode POST (request.POST). Tindakan tersebut berupa mulai melanggan atau berhenti melanggan *tag* terkait.

Laman ‘tag-reports’ juga dilengkapi dengan perintah keamanan. Digunakan LoginRequiredMixin untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga UserPassesTestMixin yang dijelaskan dengan metode test_func() untuk memastikan pengguna memiliki otoritas atas salah satu dari fitur 2 atau fitur 3 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari LoginRequiredMixin pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari UserPassesTestMixin tidak dipenuhi, pengguna akan diarahkan ke laman ‘home’ menggunakan metode handle_no_permission().

Pada laman ‘tag-reports’ pengguna dapat melihat detail informasi dari suatu *tag*. Pengguna dapat mengetahui nama, jumlah objek ‘Report’ yang terkait, pembuat, dan deskripsi dari suatu *tag* dengan melihat elemen yang ditunjuk oleh nomor 1, nomor 2, nomor 3, dan nomor 4 pada gambar A.9 secara berurutan. Pengguna juga dapat melihat objek ‘Report’ apa saja yang terkait dengan suatu *tag* dengan melihat elemen yang ditunjuk nomor 6. Penjelasan lebih detail mengenai penglihatan objek ‘Report’ akan dijabarkan pada sub sub bab 3.4.8.



Gambar A.9 Cuplikan layar laman 'tag-reports' pada kasus 1 dengan petunjuk



Gambar A.10 Cuplikan layar laman 'tag-reports' pada kasus 2 dengan petunjuk

Bila pengguna belum berlangganan *tag* tersebut, akan terlihat elemen yang ditunjuk nomor 5* pada gambar A.10. Sedangkan bila pengguna sudah berlangganan, akan terlihat elemen yang ditunjuk nomor 5 pada gambar A.9. Bila pengguna ingin memulai berlangganan, pengguna dapat memencet tombol tersebut. Memencet tombol tersebut sekali lagi akan menyebabkan pengguna berhenti berlangganan *tag* terkait.

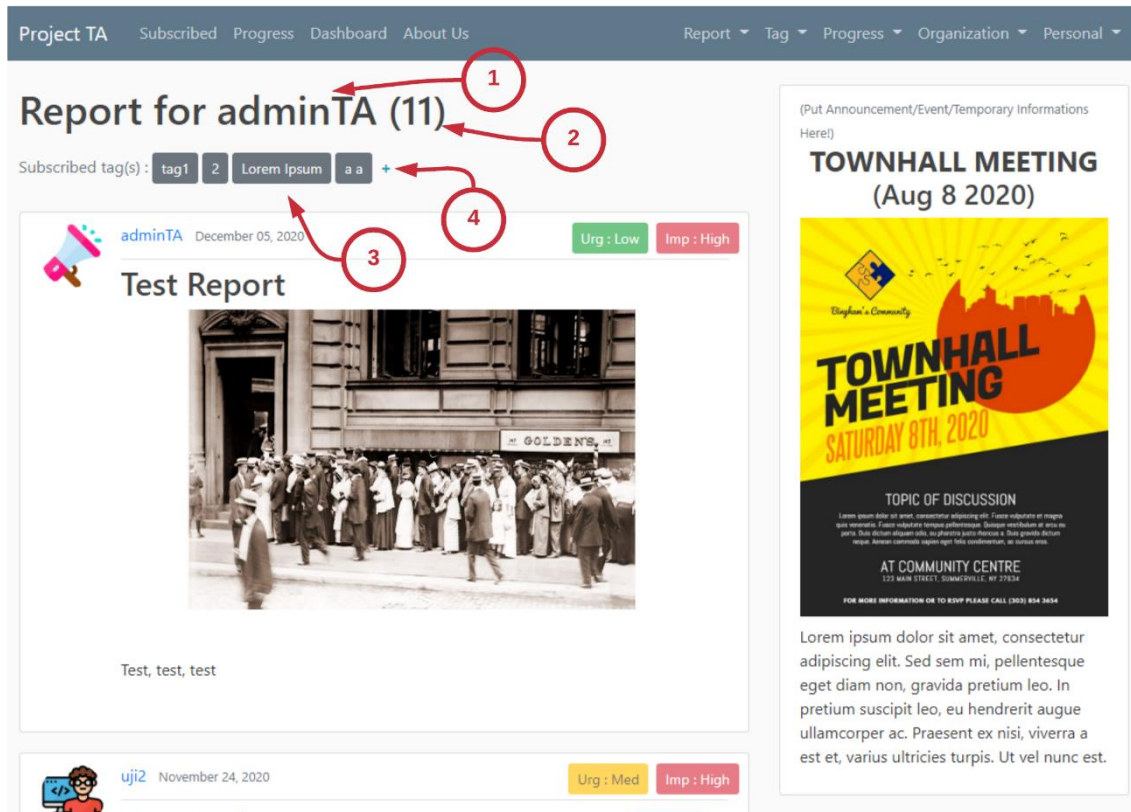
A.10. Pengembangan Laman ‘home’

Laman bernama ‘home’ ini dapat diakses dengan URL 127.0.0.1:8000. Laman ini menjadi laman dasar (*homepage*) yang akan menjadi halaman tujuan bila pengguna berhasil masuk melalui laman masuk (*login page*) atau bila pengguna dialihkan dari suatu laman karena tidak memiliki otoritas mengakses laman tersebut. Laman ‘home’ ditujukan untuk mengakomodasi aktivitas 3 fitur 2 “Melihat Laporan” mengenai penglihatan laporan langganan. Laman ini menggunakan kelas ‘SubscribedReportListView’ sebagai *view*.

Kelas ‘SubscribedReportListView’ ditulis penulis di berkas `report/views.py`. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) `ListView` sehingga menerima pewarisan (*inheritance*) dari kelas `ListView`. Kelas `SubscribedReportListView` akan memunculkan semua objek ‘*Report*’ yang terkait dengan *tag* yang dilanggan oleh pengguna. Selanjutnya sejumlah objek ‘*Report*’ akan disusun berdasarkan apakah laporan tersebut telah diambil (*taken*), kemudian berdasarkan tanggal laporannya, kemudian kemendesakannya (*urgency*), dan yang terakhir berdasarkan kepentingannya (*importance*). Kelas ini akan ditampilkan melalui *template* di berkas `report/subscribed_report.html`.

Laman ‘home’ juga dilengkapi dengan perintah keamanan. Digunakan `LoginRequiredMixin` untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga `UserPassesTestMixin` yang dijelaskan dengan metode `test_func()` untuk memastikan pengguna memiliki otoritas atas fitur 2 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari `LoginRequiredMixin` pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari `UserPassesTestMixin` tidak dipenuhi, pengguna akan diarahkan ke laman ‘*user-reports*’ terkait pengguna tersebut menggunakan metode `handle_no_permission()`.

Tampilan laman ‘home’ memiliki kemiripan dengan laman ‘*all-reports*’ karena kedua laman ini bertujuan menunjukkan laporan. Pada laman ‘home’ setiap laporan akan ditampilkan pada satu seksi konten (*content section*) dengan penjelasan yang sama dengan penjabaran isi seksi konten pada sub bab 3.4.8 mengenai laman ‘*all-reports*’. Selain menampilkan laporan yang dilanggan, laman ‘home’ juga menampilkan nama pengguna, jumlah laporan yang dilanggan, dan *tag* yang dilanggan pengguna pada elemen yang ditunjuk nomor 1, nomor 2, dan nomor 3 pada gambar A.11. Bila pengguna ingin menambah *tag* yang dilanggan, pengguna dapat memencet tombol dengan tulisan ‘+’ yang ditunjuk nomor 4 untuk diarahkan ke laman ‘*tag-list*’.



Gambar A.11 Cuplikan layar laman 'home' dengan petunjuk

A.11. Pengembangan Laman 'report-update'

Laman bernama 'report-update' ini dapat diakses dengan URL `127.0.0.1:8000/report/<int:pk>/update/`. Dimana `<int:pk>` adalah kunci primer (*primary key*) dari laporan yang ingin diubah dalam bentuk bilangan bulat. Laman ini ditujukan untuk mengakomodasi aktivitas 4 fitur 1 "Membuat Laporan" mengenai pengubahan laporan. Laman ini menggunakan kelas 'ReportUpdateView' sebagai *view*.

Kelas 'ReportUpdateView' ditulis penulis di berkas `report/views.py`. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) `UpdateView` sehingga menerima pewarisan (*inheritance*) dari kelas `UpdateView`. Kelas `ReportUpdateView` akan menangani pembuatan formulir untuk mengubah nilai dari bidang pada objek 'Report' yang dibuat. Pada form ini diminta bidang `title`, `content`, `image`, `tag`, `urgency`, dan `importance` pada objek 'Report'. Untuk bidang `reporter` akan dihubungkan dengan objek 'User' yang terasosiasi dengan pengguna yang mengubah laporan melalui metode `form_valid()`. Kelas ini akan ditampilkan melalui *template* di berkas `report/report_form.html`, sama seperti *template* untuk laman 'report-create'.

Laman *'report-update'* juga dilengkapi dengan perintah keamanan. Digunakan `LoginRequiredMixin` untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga `UserPassesTestMixin` yang dijelaskan dengan metode `test_func()` untuk memastikan pengguna adalah pelapor laporan terkait dan laporan terkait juga belum diambil untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari `LoginRequiredMixin` pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari `UserPassesTestMixin` tidak dipenuhi, pengguna akan diarahkan ke laman *'home'* menggunakan metode `handle_no_permission()`.

The screenshot shows a web application interface for creating a new report. The top navigation bar includes links for Project TA, Subscribed, Progress, Dashboard, and About Us. The right side of the navigation bar has dropdown menus for Report, Tag, Progress, Organization, and Personal.

The main content area is titled "New Report". It contains the following fields and controls:

- Title***: A text input field containing "Test Report".
- Content***: A large text area containing "Test, test, test".
- Image***: A section showing the current image path as `report_images/newyork-1392x557.jpg`. Below it is a "Change:" label and a "Choose File" button. The status "No file chosen" is displayed.
- Tag***: A dropdown menu showing a list of tags: "tag1", "dummy #2", "from url", and "1".
- Urgency***: A dropdown menu set to "Low".
- Importance***: A dropdown menu set to "High".
- Upload**: A blue button to submit the report.

On the right side of the form, there are two promotional banners:

- Townhall Meeting (Aug 8 2020)**: A yellow banner with a city skyline graphic. It mentions "The Freshman's Community" and "Topic of Discussion".
- Internal Badminton Sparring (Aug 17 2020)**: A blue and white banner with the text "THE FRESHMAN".

Gambar A.12 Cuplikan layar laman *'report-update'*

Pada laman *'report-update'* akan terlihat form yang serupa dengan form yang ditunjukkan pada laman *'report-create'*. Namun perbedaannya pada laman *'report-update'*

kolom isian dari form tersebut telah terisi dengan data lama, sehingga kolom isian yang tidak ingin diganti tidak perlu diisi ulang. Penjabaran mengenai setiap kolom isian sama dengan penjabaran pada sub sub bab 3.4.7.

A.12. Pengembangan Laman ‘report-delete’

Laman bernama ‘report-delete’ ini dapat diakses dengan URL 127.0.0.1:8000/report/<int:pk>/delete/. Dimana <int:pk> adalah kunci primer (*primary key*) dari laporan yang ingin dihapus dalam bentuk bilangan bulat. Laman ini ditujukan untuk mengakomodasi aktivitas 5 fitur 1 “Membuat Laporan” mengenai penghapusan laporan. Laman ini menggunakan kelas ‘ReportDeleteView’ sebagai *view*.

Kelas ‘ReportDeleteView’ ditulis penulis di berkas report/views.py. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) DeleteView sehingga menerima pewarisan (*inheritance*) dari kelas DeleteView. Kelas ReportDeletView akan menangani penghapusan pada objek ‘Report’ terkait. Kelas ini akan ditampilkan melalui *template* di berkas report/report_confirm_delete.html.

Laman ‘report-delete’ juga dilengkapi dengan perintah keamanan. Digunakan LoginRequiredMixin untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga UserPassesTestMixin yang dijelaskan dengan metode test_func() untuk memastikan pengguna adalah pelapor laporan bila laporan belum diambil atau pengguna adalah pengambil laporan bila laporan telah diambil untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari LoginRequiredMixin pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari UserPassesTestMixin tidak dipenuhi, pengguna akan diarahkan ke laman ‘home’ menggunakan metode handle_no_permission().



Gambar A.13 Cuplikan layar laman ‘report-delete’ dengan petunjuk

Pada laman *'report-delete'* akan ditunjukkan nama laporan yang akan dihapus pada elemen yang ditunjuk nomor 1 pada gambar A.13. Bila pengguna yakin untuk menghapus laporan tersebut, pengguna dapat memencet tombol yang ditunjuk nomor 2. Sedangkan bila pengguna ingin menguntungkan keinginan untuk menghapus, pengguna dapat memencet tombol yang ditunjuk nomor 3.

A.13. Pengembangan Laman *'progress-subscribed'*

Laman bernama *'progress-subscribed'* ini dapat diakses dengan URL 127.0.0.1:8000/progress-subscribed/. Laman ini ditujukan untuk mengakomodasi aktivitas 5 fitur 4 “Mengambil Laporan” mengenai penglihatan papan kanban dari laporan yang dilanggan. Laman ini menggunakan fungsi *'ProgressSubscribed()'* sebagai *view*.

Fungsi *'ProgressSubscribed()'* ditulis penulis di berkas *report/views.py*. Fungsi ini akan mengumpulkan semua laporan yang terkait dengan *tag* yang dilanggan pengguna dan memisahkannya berdasarkan status laporan tersebut serta diurut berdasarkan keterdesakan, kemudian berdasarkan kepentingan, dan yang terakhir berdasarkan tanggal dilaporkan. Secara garis besar laporan yang diambil akan dipisahkan menjadi 4 kelompok, laporan yang belum diambil, laporan yang masih berjalan, laporan yang sudah selesai, dan laporan yang ditolak. Fungsi *'ProgressSubscribed()'* akan ditampilkan melalui *template* di berkas *report/progress_subscribed.html*.

Laman *'progress-subscribed'* juga dilengkapi dengan perintah keamanan. Pertama diperiksa apakah pengguna sudah masuk (*login*). Bila belum pengguna akan dialihkan ke laman masuk. Kedua diperiksa apakah pengguna memiliki otoritas atas fitur 4 “Mengambil Laporan”. Bila pengguna tidak memiliki otoritas, pengguna akan dialihkan ke laman *'home'*.

Laman *'progress-subscribed'* ditujukan untuk menunjukkan status dari laporan yang dilanggan. Secara garis besar laman *'progress-subscribed'* dibagi menjadi 4 bagian, bagian laporan yang belum diambil, bagian laporan yang sedang ditangani, bagian laporan yang telah selesai ditangani, dan bagian laporan yang ditolak. Setiap bagian ditampilkan pada satu seksi konten (*content section*). Karena cara membacanya serupa, penjelasan mengenai bagian laporan yang sedang ditangani, bagian laporan yang telah selesai ditangani, dan bagian laporan yang ditolak secara mendetail dapat dilihat pada sub bab 3.4.10.

Pada bagian laporan yang belum diambil akan menampung laporan yang dilanggan yang belum diambil. Pada bagian ini setiap laporan akan ditunjukkan pada satu baris, dan

informasinya terletak pada kolom. Elemen yang ditunjuk nomor 1 pada gambar A.14 menampilkan jumlah laporan yang dilanggan yang belum diambil siapapun. Kolom yang ditunjuk nomor 2, nomor 3, dan nomor 4 menunjukkan tombol yang bertuliskan judul laporan, tombol yang bertuliskan nama pengguna (*username*) pelapor, dan cuplikan isi laporan secara berurutan. Tombol pada kolom 2 dapat mengalihkan pengguna ke laman ‘*report-detail*’ dari laporan terkait, sedangkan tombol pada kolom 3 dapat mengalihkan pengguna ke laman ‘*user-reports*’ dari pelapor terkait.

Project TA Subscribed Progress Dashboard About Us Report ▾ Tag ▾ Progress ▾ Organization ▾ Personal ▾

Not Taken Yet (1)

Report	Reporter	Content
Test Report	adminTA	Test, test, test

Ongoing (7)

Taker	Report	Field Checking	Approved	Responding	Note
07Dec20 10:35 adminTA	Urg-imp Hi-Hi	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Telah disetujui
13Sep20 14:17 Ujihp2	PC - LAN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Telah dikirim 5 orang
07Dec20 10:34 adminTA	Test Kanban 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Baru diambil
07Dec20 10:35 adminTA	Test Auth	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sedang diperiksa
13Sep20 14:18 admin	Made by Phone	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sedang dikerjakan
27Oct20 22:31 adminTA	test progress-susbscribed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
07Dec20 10:36 adminTA	Urg-imp Low-Low	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sedang diselesaikan

Finished (1)

Taker	Report	Resolved	Postponed	Failed	Note
13Sep20 13:27 adminTA	Ini gelas	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	gelas telah pecah

Not Approved (1)

Taker	Report	Reporter	Note
13Sep20 13:44 adminTA	a	adminTA	Bohong

(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING (Aug 8 2020)

TOPIC OF DISCUSSION

AT COMMUNITY CENTRE
123 MAIN STREET, SUMMITVILLE, NY 27234

FOR MORE INFORMATION OR TO RSVP PLEASE CALL (303) 804 3454

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sem mi, pellentesque eget diam non, gravida pretium leo. In pretium suscipit leo, eu hendrerit augue ullamcorper ac. Praesent ex nisi, viverra a est et, varius ultricies turpis. Ut vel nunc est.

(Put Announcement/Event/Temporary Informations Here!)

INTERNAL BADMINTON SPARRING (Aug 17 2020)

Gambar A.14 Cuplikan layar laman ‘*progress-subscribed*’ dengan petunjuk

Pada bagian laporan yang sedang ditangani, bagian laporan yang telah terselesaikan, dan bagian laporan yang ditolak memiliki struktur yang serupa dengan papan kanban pada laman *'progress-taken'*. Perbedaannya yaitu di isinya, dimana laman *'progress-taken'* menunjukkan laporan yang diambil oleh pengguna itu sendiri, sedangkan pada laman *'progress-subscribed'* menunjukkan laporan dengan *tag* yang dilanggan. Hal itu berarti pengguna yang mengambil laporan yang ditunjukkan pada laman *'progress-subscribed'* beragam. Sehingga perlu ditambahkan kolom baru yang menunjukkan siapa pengguna yang mengambil laporan tersebut. Kolom pengguna pengambil dapat terlihat pada kolom yang ditunjuk nomor 5, nomor 6, dan nomor 7 untuk bagian laporan yang sedang ditangani, bagian laporan yang telah terselesaikan, dan bagian laporan yang ditolak secara berurutan.

A.14. Pengembangan Laman *'unit-list'*

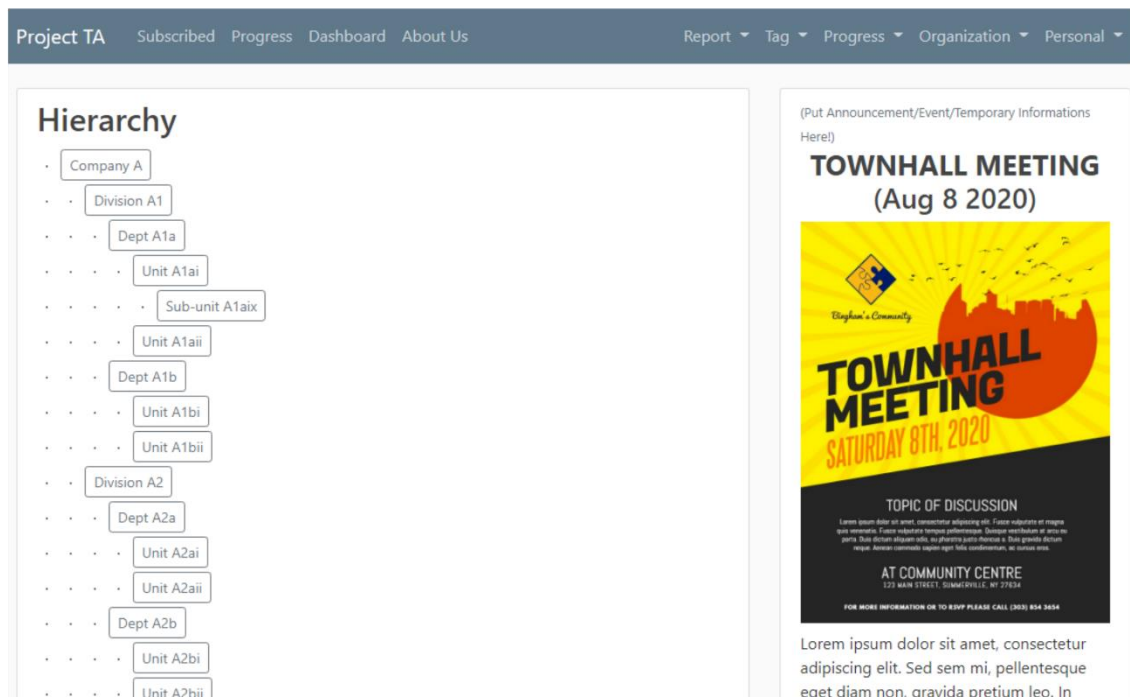
Laman bernama *'unit-list'* ini dapat diakses dengan URL 127.0.0.1:8000/unit-hierarchy/. Laman ini ditujukan untuk mengakomodasi aktivitas 4 fitur 6 “Mengelola Organisasi” mengenai pengelolaan anggota pada unit terkait. Laman ini menggunakan fungsi *'UnitHierarchy()'* sebagai *view*.

Fungsi *'UnitHierarchy()'* ditulis penulis di berkas *user/views.py*. Fungsi ini akan mengumpulkan semua unit dengan tingkatan (*level*) 1, atau unit dengan tingkatan tertinggi. Bila satu sistem hanya digunakan oleh satu organisasi, maka hanya ada satu unit dengan tingkatan 1. Namun bila pada sistem ini ditempati oleh lebih dari satu organisasi, maka akan ada lebih dari satu unit dengan tingkatan 1 (tingkatan tertinggi). Fungsi *'UnitHierarchy()'* akan ditampilkan melalui *template* di berkas *user/unit_list.html*.

Laman *'unit-list'* juga dilengkapi dengan perintah keamanan. Pertama diperiksa apakah pengguna sudah masuk (*login*). Bila belum pengguna akan dialihkan ke laman masuk. Kedua diperiksa apakah pengguna memiliki otoritas atas fitur 6 “Mengelola Organisasi”. Bila pengguna tidak memiliki otoritas, pengguna akan dialihkan ke laman *'home'*.

Laman *'unit-list'* akan menunjukkan semua unit dan disusun berdasarkan hierarkinya. Unit dengan hierarki tertinggi berada di paling kiri. Dan bila unit dibawahnya berposisi lebih ke kanan, hal tersebut berarti unit yang tepat di bawahnya adalah unit bawahan dari unit tersebut. Pada kasus pada cuplikan layar pada gambar A.15 unit dengan tingkatan tertinggi bernama *'Company A'*. *'Company A'* memiliki 2 unit bawahan langsung, yaitu *'Division A1'*

dan ‘Division A2’. Selanjutnya ‘Division A1’ memiliki 2 unit bawahan langsung, yaitu ‘Dept A1a’ dan ‘Dept A1b’. Begitu juga dengan ‘Division A2’ memiliki 2 unit bawahan langsung, yaitu ‘Dept A2a’ dan ‘Dept A2b’. Dan mengikuti logika seperti itu sampai seterusnya. Nama unit yang ditampilkan pada laman ini ditampilkan pada suatu tombol, dimana tombolnya dapat dipencet bila pengguna ingin dialihkan ke laman ‘*unit-detail*’ dari unit terkait.



Gambar A.15 Cuplikan layar laman ‘*unit-list*’

A.15. Pengembangan Laman ‘*user-taken*’

Laman bernama ‘*user-taken*’ ini dapat diakses dengan URL `127.0.0.1:8000/user/<str:username>/taken/`. Dimana `<str:username>` adalah nama pengguna (*username*) dari pengguna terkait dalam bentuk string. Laman ‘*user-taken*’ ditujukan untuk mengakomodasi aktivitas 6 fitur 4 “Mengambil Laporan” dan aktivitas 2 fitur 5 “Melihat Aktivitas Pengguna” mengenai penglihatan laporan yang diambil pengguna tertentu. Laman ini menggunakan kelas ‘*UserTakenListView*’ sebagai *view*.

Kelas ‘*UserTakenListView*’ ditulis penulis di berkas `report/views.py`. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) *ListView* sehingga menerima pewarisan (*inheritance*) dari kelas *ListView*. Kelas *UserTakenListView* akan memunculkan semua objek ‘*Report*’ yang diambil oleh pengguna tertentu. Selanjutnya sejumlah objek ‘*Report*’ akan disusun berdasarkan berdasarkan tanggal laporannya, kemudian kemendesakannya (*urgency*),

dan yang terakhir berdasarkan kepentingannya (*importance*). Kelas ini akan ditampilkan melalui *template* di berkas `report/user_taken.html`.

Laman '*user-taken*' juga dilengkapi dengan perintah keamanan. Digunakan `LoginRequiredMixin` untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga `UserPassesTestMixin` yang dijelaskan dengan metode `test_func()` untuk memastikan bahwa laman tersebut terkait pengguna yang akan mengakses atau pengguna memiliki otoritas atas fitur 4 atau fitur 5 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari `LoginRequiredMixin` pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari `UserPassesTestMixin` tidak dipenuhi, pengguna akan diarahkan ke laman '*user-collab*' terkait pengguna yang laman '*user-takens*'-nya ingin diakses menggunakan metode `handle_no_permission()`.

The screenshot displays the 'adminTA' user interface. At the top, there is a navigation bar with links: Project TA, Subscribed, Progress, Dashboard, About Us, Report, Tag, Progress, Organization, and Personal. Below the navigation bar, the user profile for 'adminTA' is shown, including the role 'Head of Company A' and email 'TA.13116020@gmail.com'. A statistics table is present:

Point	7	Reported Report	19
Report Taken	11	Report Taken On Progress	7
Report Taken Not Approved	1	Report Taken Finished	3

Below the table, there are tabs for 'Report Reported', 'Report Taken', 'Collaboration', 'Career History', and 'Point History'. The 'Report Taken' tab is active, showing a report card for 'Urg-Imp Hi-Hi' with a 'Taken' button. The report card includes a lock icon, the user 'admin', the date 'September 04, 2020', and status indicators 'Urg : High' and 'Imp : High'. A large graphic of a dollar coin with a green arrow pointing up is displayed. Below the graphic, there is a line of placeholder text: 'critical critical critical critical critical critical critical critical critical critical critical critical critical critical critical criti ...'. On the right side, there are two event announcements. The first is for a 'TOWNHALL MEETING (Aug 8 2020)' with a poster image. The second is for 'INTERNAL BADMINTON SPARRING (Aug 17 2020)'.

Gambar A.16 Cuplikan layar laman '*user-taken*'

Tampilan laman *'user-taken'* memiliki kemiripan dengan laman *'user-reports'* karena kedua laman ini bertujuan untuk menunjukkan data pribadi dan data statistik dari pengguna terkait serta menunjukkan laporan. Pada laman *'user-taken'* data pribadi dan data statistik pengguna terkait ditampilkan pada bagian atas laman dengan penjelasan yang serupa dengan penjelasan data pribadi dan data statistik pada sub sub bab 3.4.13 mengenai laman *'user-reports'*. Seperti halnya dengan laporan yang ditampilkan juga memiliki penjelasan yang sama dengan penjelasan isi seksi konten pada sub sub bab 3.4.13 mengenai laman *'user-reports'*. Perbedaan kedua laman ini hanya jenis laporan yang ditampilkan, pada laman *'user-reports'* menunjukkan laporan yang pengguna terkait laporkan. Sedangkan pada laman *'user-taken'* menunjukkan laporan yang pengguna terkait ambil seperti yang terlihat pada gambar A.16.

A.16. Pengembangan Laman *'user-collab'*

Laman bernama *'user-collab'* ini dapat diakses dengan URL `127.0.0.1:8000/user/<str:username>/collaboration/`. Dimana `<str:username>` adalah nama pengguna (*username*) dari pengguna terkait dalam bentuk string. Laman *'user-collab'* ditujukan untuk mengakomodasi aktivitas 7 fitur 2 “Melihat Laporan”, aktivitas 3 fitur 5 “Melihat Aktivitas Pengguna”, dan aktivitas 3 fitur 9 “Berkolaborasi pada Laporan” mengenai penglihatan kolaborasi dari pengguna tertentu. Laman ini menggunakan kelas *'UserCollaborationListView'* sebagai *view*.

Kelas *'UserCollaborationListView'* ditulis penulis di berkas `report/views.py`. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) *ListView* sehingga menerima pewarisan (*inheritance*) dari kelas *ListView*. Kelas *UserCollaborationListView* akan memunculkan semua objek *'Collaboration'* dari pengguna tertentu. Selanjutnya sejumlah objek *'Collaboration'* akan disusun berdasarkan berdasarkan tanggal kolaborasi. Kelas ini akan ditampilkan melalui *template* di berkas `report/user_collab.html`.

Laman *'user-collab'* juga dilengkapi dengan perintah keamanan. Digunakan *LoginRequiredMixin* untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga *UserPassesTestMixin* yang dijelaskan dengan metode `test_func()` untuk memastikan bahwa laman tersebut terkait pengguna yang akan mengakses atau pengguna memiliki otoritas atas fitur 2, fitur 5, atau fitur 9 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari *LoginRequiredMixin* pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari *UserPassesTestMixin* tidak dipenuhi, pengguna akan diarahkan ke

laman 'user-career' terkait pengguna yang laman 'user-collab'-nya ingin diakses menggunakan metode `handle_no_permission()`.

Tampilan laman 'user-collab' memiliki kemiripan dengan laman 'user-reports' karena salah satu tujuan laman ini adalah untuk menunjukkan data pribadi dan data statistik dari pengguna terkait. Pada laman 'user-collab' data pribadi dan data statistik pengguna terkait ditampilkan pada bagian atas laman dengan penjelasan yang serupa dengan penjelasan data pribadi dan data statistik pada sub sub bab 3.4.13 mengenai laman 'user-reports'.

Pada bagian bawah laman ini ditujukan untuk menampilkan kolaborasi dari pengguna terkait. Penampilan kolaborasi juga tidak jauh berbeda dengan penampilan laporan pada umumnya. Pada penampilan kolaborasi ditampilkan judul dari kolaborasi dan judul dari laporan yang dikolaborasi pada elemen yang ditunjuk nomor 2 dan nomor 1 pada gambar A.17 secara berturut-turut. Gambar bukti kolaborasi serta isi kolaborasi juga ditampilkan pada elemen yang ditunjuk nomor 3 dan nomor 4.

The screenshot shows the 'user-collab' page for a user named 'adminTA', who is the 'Head of Company A' with email 'TA.13116020@gmail.com'. The page has a top navigation bar with links like 'Project TA', 'Subscribed', 'Progress', 'Dashboard', and 'About Us'. Below the user profile, there is a table with statistics:

Point	7	Reported Report	19
Report Taken	11	Report Taken On Progress	7
Report Taken Not Approved	1	Report Taken Finished	3

Below the table are tabs for 'Report Reported', 'Report Taken', 'Collaboration', 'Career History', and 'Point History'. The main content area shows a post by 'adminTA' from December 05, 2020, titled 'PC - LAN' with a sub-title '↳ Test Collab'. Below the title is a large image of a Japanese wave (The Great Wave off Kanagawa). At the bottom of the post is the text 'test collaboration'. Four red circles with numbers 1 through 4 point to the following elements: 1. The user profile icon and name 'adminTA'. 2. The title 'PC - LAN'. 3. The image of the wave. 4. The text 'test collaboration'.

On the right side of the page, there are two announcement boxes. The top one is for a 'TOWNHALL MEETING (Aug 8 2020)' with a yellow and red poster. The bottom one is for 'INTERNAL BADMINTON SPARRING (Aug 17)' with a black and white poster.

Gambar A.17 Cuplikan layar laman 'user-collab' dengan petunjuk

A.17. Pengembangan Laman ‘*user-career*’

Laman bernama ‘*user-career*’ ini dapat diakses dengan URL 127.0.0.1:8000/user/<str:username>/career/. Dimana <str:username> adalah nama pengguna (*username*) dari pengguna terkait dalam bentuk string. Laman ‘*user-career*’ ditujukan untuk mengakomodasi aktivitas 4 fitur 5 “Melihat Aktivitas Pengguna” dan aktivitas 5 fitur 6 “Mengelola Organisasi” mengenai penglihatan rekam jejak karir dari pengguna tertentu. Laman ini menggunakan kelas ‘*UserCareerListView*’ sebagai *view*.

Kelas ‘*UserCareerListView*’ ditulis penulis di berkas report/views.py. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) *ListView* sehingga menerima pewarisan (*inheritance*) dari kelas *ListView*. Kelas *UserCareerListView* akan memunculkan semua objek ‘*CareerHistory*’ dari pengguna tertentu. Selanjutnya sejumlah objek ‘*CareerHistory*’ akan disusun berdasarkan berdasarkan tanggal tercatatnya rekam jejak karir. Kelas ini akan ditampilkan melalui *template* di berkas report/user_career.html.

Laman ‘*user-career*’ juga dilengkapi dengan perintah keamanan. Digunakan *LoginRequiredMixin* untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga *UserPassesTestMixin* yang dijelaskan dengan metode *test_func()* untuk memastikan bahwa laman tersebut terkait pengguna yang akan mengakses atau pengguna memiliki otoritas atas fitur 5 atau fitur 6 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari *LoginRequiredMixin* pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari *UserPassesTestMixin* tidak dipenuhi, pengguna akan diarahkan ke laman ‘*user-point*’ terkait pengguna yang laman ‘*user-career*’-nya ingin diakses menggunakan metode *handle_no_permission()*.

Tampilan laman ‘*user-career*’ memiliki kemiripan dengan laman ‘*user-reports*’ karena salah satu tujuan laman ini adalah untuk menunjukkan data pribadi dan data statistik dari pengguna terkait. Pada laman ‘*user-career*’ data pribadi dan data statistik pengguna terkait ditampilkan pada bagian atas laman dengan penjelasan yang serupa dengan penjelasan data pribadi dan data statistik pada sub bab 3.4.13 mengenai laman ‘*user-reports*’.

Pada bagian bawah laman ini ditujukan untuk menampilkan rekam jejak karir dari pengguna terkait. Setiap catatan karir ditampilkan pada satu baris dan informasi terkait ditampilkan pada kolom yang terstruktur. Ditampilkan kunci primer (*primary key*), tanggal dimulainya pekerjaan, tanggal berakhirnya pekerjaan, unit tempat bekerja, dan pekerjaan yang

dikerjakan pada suatu catatan karir pada kolom yang ditunjuk nomor 1, nomor 2, nomor 3, nomor 4, dan nomor 5 pada gambar A.18 secara berurutan.

The screenshot shows the 'user-career' page in the adminTA system. The page header includes navigation links: Project TA, Subscribed, Progress, Dashboard, About Us, Report, Tag, Progress, Organization, and Personal. The user profile for 'adminTA' (Head of Company A, TA.13116020@gmail.com) is displayed. Below the profile is a table with the following data:

Point	7	Reported Report	19
Report Taken	11	Report Taken On Progress	7
Report Taken Not Approved	1	Report Taken Finished	3

Below the table is a section with five tabs: Report Reported, Report Taken, Collaboration, Career History, and Point History. The 'Career History' tab is active, showing a table with the following data:

Reg-ID	Date Started	Date Ended	Unit	Job
32	Oct 31, 2020	Incumbent	Company A	Head
35	Sep 18, 2020	Oct 31, 2020	Division A2	Head

Five red circles with numbers 1 through 5 are placed over the table headers: 1 over 'Reg-ID', 2 over 'Date Started', 3 over 'Date Ended', 4 over 'Unit', and 5 over 'Job'. To the right of the table is a sidebar with a 'TOWNHALL MEETING' announcement for August 8, 2020.

Gambar A.18 Cuplikan layar laman 'user-career' dengan petunjuk

A.18. Pengembangan Laman 'user-point'

Laman bernama 'user-point' ini dapat diakses dengan URL `127.0.0.1:8000/user/<str:username>/point/`. Dimana `<str:username>` adalah nama pengguna (*username*) dari pengguna terkait dalam bentuk string. Laman 'user-point' ditujukan untuk mengakomodasi aktivitas 5 fitur 5 "Melihat Aktivitas Pengguna" dan aktivitas 3 fitur 8 "Mengelola Poin" mengenai penglihatan log poin kepada pengguna tertentu. Laman ini menggunakan kelas 'UserPointListView' sebagai *view*.

Kelas 'UserPointListView' ditulis penulis di berkas `report/views.py`. Kelas ini merupakan subkelas dari kelas bawaan (*built-in*) `ListView` sehingga menerima pewarisan (*inheritance*) dari kelas `ListView`. Kelas `UserPointListView` akan memunculkan semua objek 'PointHistory' dari pengguna tertentu. Selanjutnya sejumlah objek 'PointHistory' akan disusun berdasarkan berdasarkan tanggal tercatatnya log poin. Kelas ini akan ditampilkan melalui *template* di berkas `report/user_point.html`.

Laman 'user-point' juga dilengkapi dengan perintah keamanan. Digunakan `LoginRequiredMixin` untuk mencegah pengguna yang belum masuk (*login*) untuk mengakses laman ini. Digunakan juga `UserPassesTestMixin` yang dijelaskan dengan metode `test_func()`

untuk memastikan bahwa laman tersebut terkait pengguna yang akan mengakses atau pengguna memiliki otoritas atas fitur 5 atau fitur 8 untuk mengakses laman ini. Bila pengguna tidak memenuhi syarat dari `LoginRequiredMixin` pengguna akan dialihkan ke laman masuk. Sedangkan bila syarat dari `UserPassesTestMixin` tidak dipenuhi, pengguna akan diarahkan ke laman 'home' menggunakan metode `handle_no_permission()`.

Tampilan laman 'user-point' memiliki kemiripan dengan laman 'user-reports' karena salah satu tujuan laman ini adalah untuk menunjukkan data pribadi dan data statistik dari pengguna terkait. Pada laman 'user-career' data pribadi dan data statistik pengguna terkait ditampilkan pada bagian atas laman dengan penjelasan yang serupa dengan penjelasan data pribadi dan data statistik pada sub sub bab 3.4.13 mengenai laman 'user-reports'.

The screenshot shows the 'adminTA' user profile page. At the top, there's a navigation bar with 'Project TA', 'Subscribed', 'Progress', 'Dashboard', and 'About Us'. Below this, the user's profile is displayed with a megaphone icon, the name 'adminTA', and the role 'Head of Company A' with email 'TA.13116020@gmail.com'. A statistics table shows: Point (4), Reported Report (19), Report Taken (11), Report Taken On Progress (7), Report Taken Not Approved (1), and Report Taken Finished (3). Below the statistics are five navigation tabs: 'Report Reported' (1), 'Report Taken' (2), 'Collaboration' (3), 'Career History' (4), and 'Point History' (5). The 'Point History' table is shown below, with columns: Reg-ID, Date, Point, Writer, and Note. The table contains three rows of data.

Reg-ID	Date	Point	Writer	Note
7	Dec 07, 2020	-3	adminTA	Ditukar dengan bonus bulan Desember
5	Nov 12, 2020	4	adminTA	Point from report: "Urg-Imp Low-Low", pk = 28
1	Oct 17, 2020	3	adminTA	Point from report: "Test Kanban 1", pk = 30

On the right side of the page, there is a 'TOWNHALL MEETING' announcement for August 8, 2020, at the Community Centre.

Gambar A.19 Cuplikan layar laman 'user-point' dengan petunjuk

Pada bagian bawah laman ini ditujukan untuk menampilkan log poin kepada pengguna terkait. Setiap log poin ditampilkan pada satu baris dan informasi terkait ditampilkan pada kolom yang terstruktur. Ditampilkan kunci primer (*primary key*), tanggal tanggal diberikan, jumlah, pemberi, dan catatan pada suatu log poin pada kolom yang ditunjuk nomor 1, nomor 2, nomor 3, nomor 4, dan nomor 5 pada gambar A.19 secara berurutan.

Pada contoh kasus pada cuplikan layar ini, terlihat log poin dengan pk=7 yang tertampil pada baris pertama dibuat dengan laman 'point-create'. Hal tersebut dapat dikira karena catatan dari log poin tersebut tidak mengikuti format tertentu. Sedangkan log poin dengan pk=5 dan

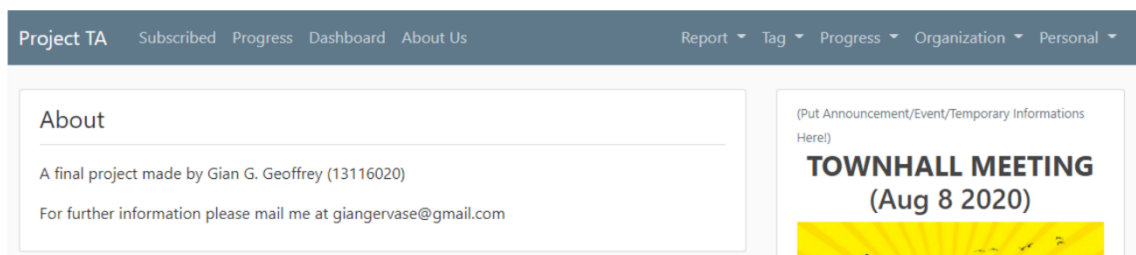
pk=1 pada baris tengah dan baris paling bawah didapat dari laporan yang diberi poin. Hal tersebut dapat diketahui karena catatan log poin tersebut mengikuti format yang menunjukkan dari mana poin tersebut berasal.

A.19. Pengembangan Laman ‘about’

Laman bernama ‘about’ ini dapat diakses dengan URL 127.0.0.1:8000/about/. Laman ini ditujukan untuk mengakomodasi aktivitas 3 fitur “Tambahan” mengenai penglihatan laman tentang kami. Laman ini menggunakan fungsi ‘About()’ sebagai *view*.

Fungsi ‘About()’ ditulis penulis di berkas report/views.py. Fungsi ini akan diisi pesan dari pengguna atau dari perusahaan yang akan ditampilkan. Fungsi ‘About()’ akan ditampilkan melalui *template* di berkas report/about.html.

Laman ‘about’ tidak dilengkapi dengan perintah keamanan. Siapa saja dapat mengakses laman ini. Maka dari itu informasi pada laman ini dirancang untuk diakses pihak dalam maupun luar organisasi. Cuplikan laman ini dapat dilihat pada gambar A.20.



Gambar A.20 Cuplikan layar laman ‘about’

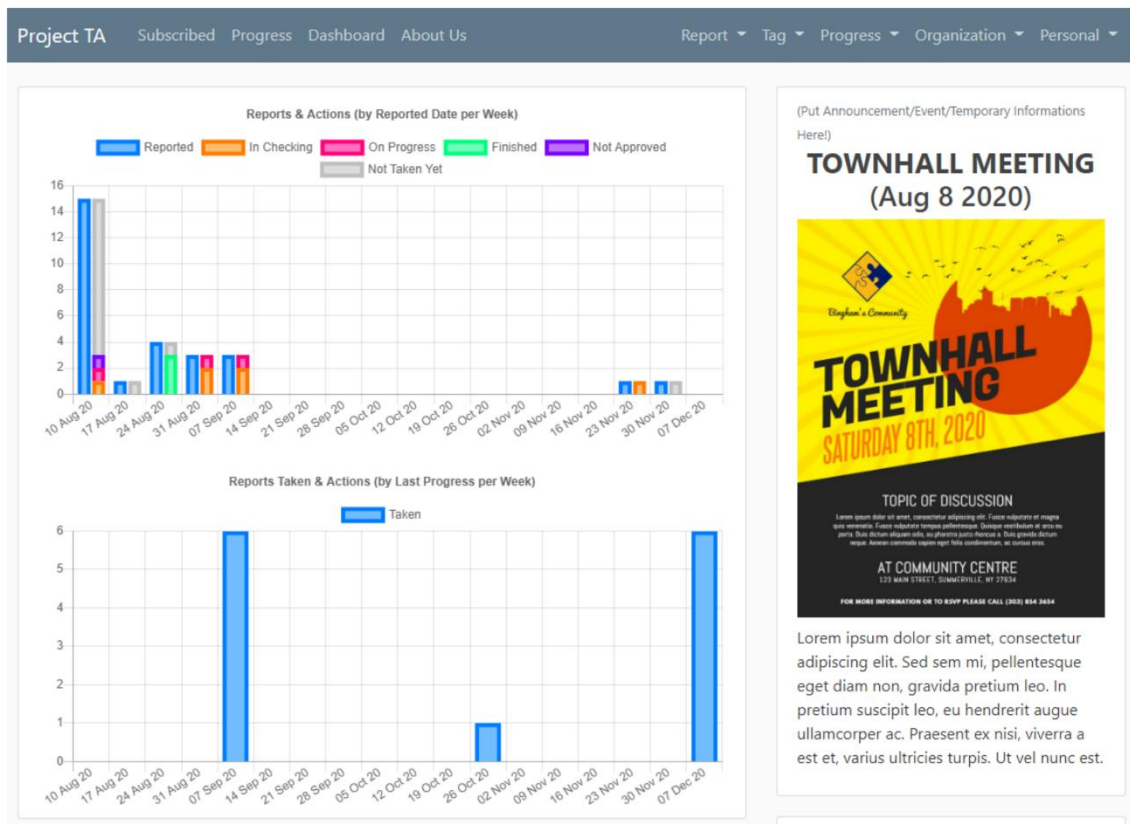
A.20. Pengembangan Laman ‘dashboard’

Laman bernama ‘dashboard’ ini dapat diakses dengan URL 127.0.0.1:8000/dashboard/. Laman ini ditujukan untuk mengakomodasi aktivitas 1 fitur “Tambahan” mengenai penglihatan laman dasbor. Laman ini menggunakan fungsi ‘Dashboard()’ sebagai *view*.

Fungsi ‘Dashboard()’ ditulis penulis di berkas report/views.py. Fungsi ini mengumpulkan segala data yang dibutuhkan untuk ditampilkan pada dasbor. Fungsi ‘Dashboard()’ akan ditampilkan melalui *template* di berkas report/dashboard.html.

Laman ‘dashboard’ juga dilengkapi dengan perintah keamanan. Diperiksa apakah pengguna sudah masuk (*login*). Bila belum pengguna akan dialihkan ke laman masuk. Bila pengguna telah masuk, pengguna dapat mengakses laman ini tanpa prasyarat otoritas. Cuplikan

laman ‘*dashboard*’ dapat dilihat pada gambar A.21. Bila perusahaan ingin visualisasi statistik lainnya, dapat ditambah data dan visualisasi yang diinginkan kedepannya.

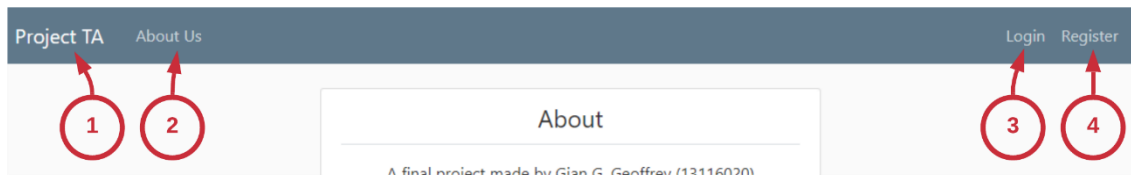


Gambar A.21 Cuplikan layar laman ‘*dashboard*’

A.21. Pengembangan *Template* Dasar

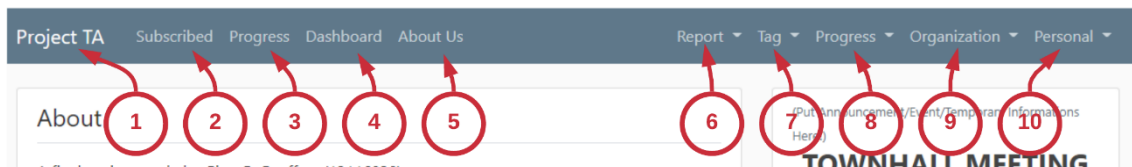
Bila diperhatikan, semua gambar cuplikan laman pada sub bab sebelumnya memiliki kesamaan. Hal tersebut karena semua laman memiliki *template* dasar yang sama, namun memuat konten yang berbeda. *Template* dasar dibuat penulis di berkas `templates/base.html`. Tampilan pada `base.html` dimunculkan di laman spesifik menggunakan fungsi `{% extends 'base.html' %}` berbahasa DTL (*Django Template Language*).

Pada kasus aplikasi web ini, secara umum *template* dasar menampilkan batang navigasi (*navigation bar / navbar*) dan batang samping (*sidebar*). Batang navigasi akan ditampilkan baik saat pengguna masuk (*login*) maupun tidak. Sedangkan batang samping hanya akan muncul saat pengguna telah masuk (*login*). Walaupun begitu tombol-tombol pada batang navigasi akan berubah bergantung pada masuk atau tidaknya pengguna dan otoritas fitur yang pengguna itu miliki.



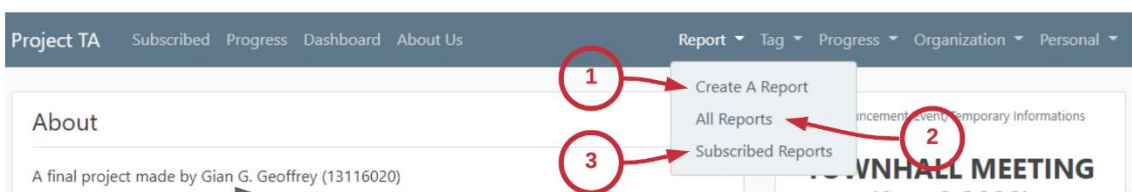
Gambar A.22 Cuplikan layar bagian batang navigasi pada kasus 1 di laman 'about'

Pada saat pengguna belum masuk, batang navigasi hanya menampilkan tombol seperti pada gambar A.22. Tombol yang ditunjuk nomor 1 bertuliskan nama dari proyek ini. Tombol yang ditunjuk nomor 1 dan nomor 3 bertujuan sama, yaitu mengalihkan pengguna ke laman 'login'. Tombol yang ditunjuk nomor 2 akan mengalihkan pengguna ke laman 'about'. Sedangkan tombol yang ditunjuk nomor 4 akan mengalihkan pengguna ke laman 'register'.



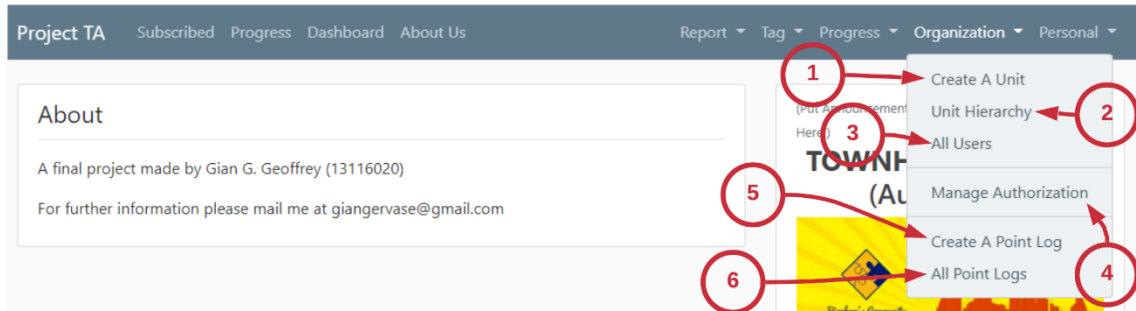
Gambar A.23 Cuplikan layar bagian batang navigasi pada kasus 2 di laman 'about'

Bila pengguna telah masuk (*login*), akan terlihat lebih banyak tombol pada batang navigasi. Namun tampilannya pun beragam, bergantung dengan otoritas pengguna tersebut. Pada gambar A.23 ditampilkan batang navigasi, dimana pada kasus ini pengguna yang telah masuk dan memiliki kesembilan (seluruh) otoritas yang ada. Tombol yang ditunjuk nomor 1 bertuliskan nama proyek ini. Tombol yang ditunjuk nomor 1 dan nomor 2 akan mengalihkan pengguna ke laman 'home'. Tombol yang ditunjuk nomor 3, nomor 4 dan nomor 5 akan mengalihkan pengguna ke laman 'progress-taken', laman 'dashboard', dan laman 'about' secara berurutan. Elemen yang ditunjuk nomor 6 hingga nomor 10 akan memunculkan menu *dropdown* bila dipencet.



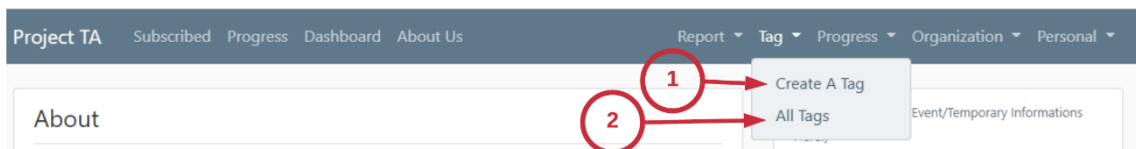
Gambar A.24 Cuplikan layar bagian batang navigasi pada kasus 3 di laman 'about'

Bila pengguna memencet elemen nomor 6 pada gambar A.23, akan muncul menu *dropdown* yang berisi tombol yang berkaitan dengan laporan seperti pada gambar A.24. Tombol yang ditunjuk nomor 1, nomor 2, dan nomor 3 pada gambar A.24 akan mengarahkan pengguna ke laman '*report-create*', laman '*all-reports*', dan laman '*home*'.



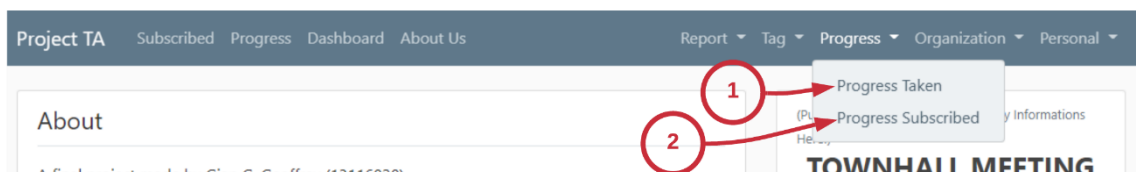
Gambar A.25 Cuplikan layar bagian batang navigasi pada kasus 4 di laman '*about*'

Bila pengguna memencet elemen nomor 7 pada gambar A.23, akan muncul menu *dropdown* yang berisi tombol yang berkaitan dengan tag seperti pada gambar A.25. Tombol yang ditunjuk nomor 1 dan nomor 2 pada gambar A.25 akan mengarahkan pengguna ke laman '*tag-create*' dan laman '*tag-list*'.



Gambar A.26 Cuplikan layar bagian batang navigasi pada kasus 5 di laman '*about*'

Bila pengguna memencet elemen nomor 8 pada gambar A.23, akan muncul menu *dropdown* yang berisi tombol yang berkaitan dengan penanganan laporan seperti pada gambar A.26. Tombol yang ditunjuk nomor 1 dan nomor 2 pada gambar A.26 akan mengarahkan pengguna ke laman '*progress-taken*' dan laman '*progress-subscribed*'.



Gambar A.27 Cuplikan layar bagian batang navigasi pada kasus 6 di laman '*about*'

Bila pengguna memencet elemen nomor 9 pada gambar A.23, akan muncul menu *dropdown* yang berisi tombol yang berkaitan dengan keorganisasian seperti pada gambar A.27. Tombol yang ditunjuk nomor 1, nomor 2, nomor 3, nomor 4, nomor 5, dan nomor 6 pada gambar A.27 akan mengarahkan pengguna ke laman ‘*unit-create*’, laman ‘*unit-list*’, laman ‘*user-list*’, laman ‘*auth-detail*’, laman ‘*point-create*’, dan laman ‘*point-list*’.

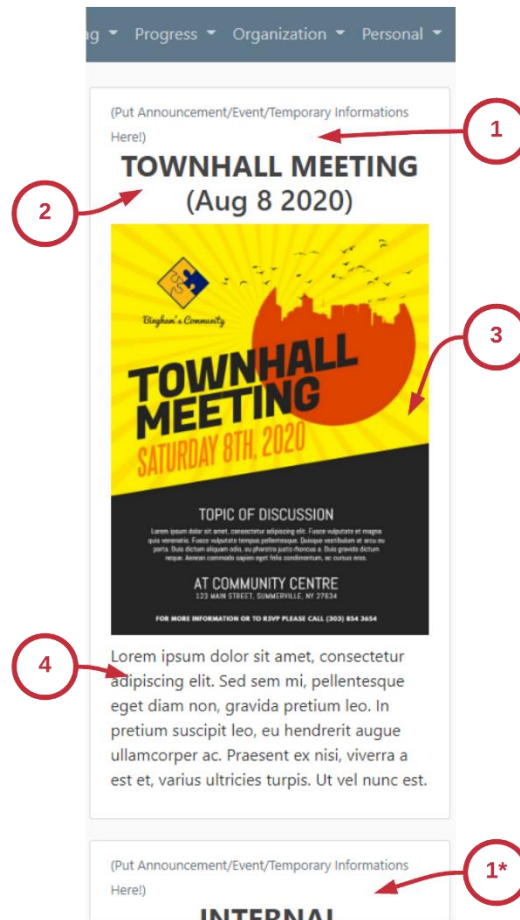


Gambar A.28 Cuplikan layar bagian batang navigasi pada kasus 7 di laman ‘*about*’

Dan apabila pengguna memencet elemen nomor 10 pada gambar A.23, akan muncul menu *dropdown* yang berisi tombol yang berkaitan dengan pribadi pengguna seperti pada gambar A.28. Tombol yang ditunjuk nomor 1 dan nomor 2 pada gambar A.28 akan mengarahkan pengguna ke laman ‘*user-reports*’ dari pengguna itu sendiri dan laman ‘*profile*’. Sedangkan tombol yang ditunjuk nomor 3 akan mengeluarkan akun pengguna (*logout*) dan pengguna akan dialihkan ke laman ‘*logout*’.

Seperti yang telah disebutkan sebelumnya, tombol yang ditampilkan di batang navigasi bergantung pada otoritas pengguna tertentu. Bila pengguna tersebut memiliki otoritas untuk mengakses laman yang dijabarkan di atas, akan muncul tombol terkait. Namun apabila pengguna tersebut tidak memiliki otoritas untuk mengakses laman yang dijabarkan di atas, tidak akan muncul tombol terkait. Untuk elemen yang dapat mengaktifkan *dropdown* menu, akan muncul bila paling tidak ada satu tombol di dalamnya yang muncul, Bila tidak ada sama sekali tombol di dalamnya yang muncul, elemen yang dapat mengaktifkan *dropdown* menu tersebut tidak akan ditampilkan.

Selain batang navigasi, *template* dasar juga memunculkan batang samping bila pengguna telah masuk (*login*). Karena bagian dari *template* dasar, batang samping ini akan muncul di semua halaman bila pengguna telah masuk. Penampilan batang samping ini ditujukan untuk mengakomodasi aktivitas 2 fitur ‘Tambahan’ mengenai melihat batang samping (*sidebar*). Batang samping akan berisi pengumuman atau informasi yang ingin disampaikan perusahaan kepada pengguna yang telah masuk. Seharusnya informasi tersebut dapat terlihat dengan mudah karena batang samping akan muncul di sepertiga bagian kanan setiap laman.



Gambar A.29 Cuplikan layar bagian batang samping pada laman 'about'

Lampiran B

Pengujian Fitur pada Aplikasi Web

B.1. Pengujian Fitur 1 “Membuat Laporan” (*Create a Report*)

Pengujian pada fitur 1 “Membuat Laporan” akan meliputi pengujian laman ‘*tag-list*’, ‘*report-create*’, ‘*all-reports*’, ‘*report-detail*’, ‘*report-update*’, dan ‘*report-delete*’. Akan dilakukan tiga jenis pengujian (bila memungkinkan) pada setiap laman di atas melalui tiga peramban web. Sebelumnya dilakukan pengujian, otoritas akun uji2 atas fitur 1 akan dimatikan, sedangkan otoritas akun uji2 atas fitur lain tetap diberikan.

Pada laman ‘*tag-list*’ akan diuji aksesibilitasnya dan interaksinya. Namun tidak diuji pengunggahan datanya karena tidak ada data yang pengguna dapat unggah melalui metode POST. Dan sebelum pengujian laman ini dimulai, otoritas akun uji2 atas fitur 2 dan fitur 3 akan dimatikan karena mempengaruhi laman ini.

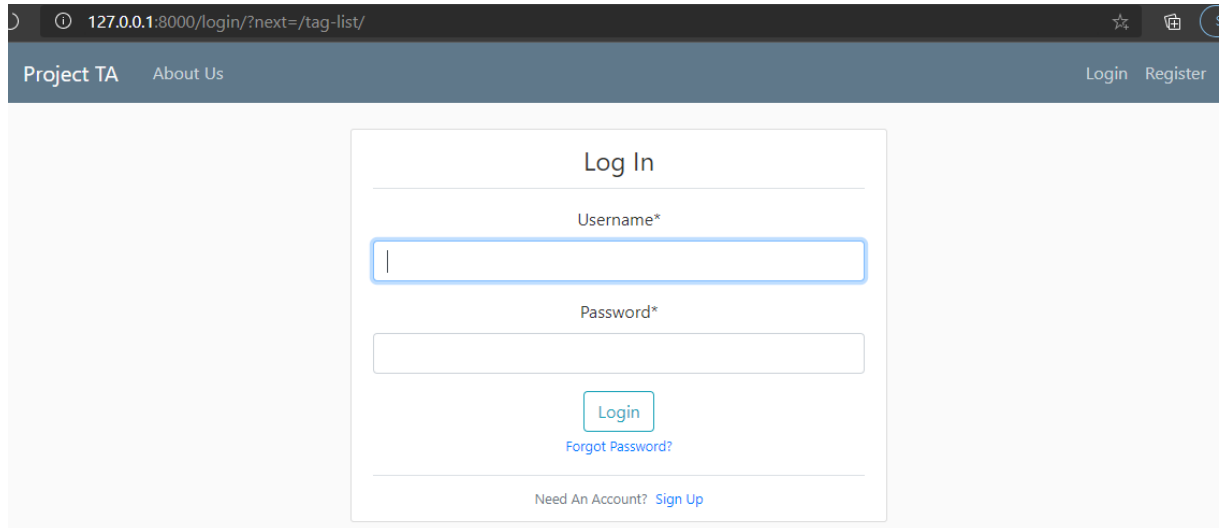
Pada peramban pertama, laman ‘*tag-list*’ dapat diakses dan semua elemen dapat berinteraksi dengan baik. Sebagai contoh, penulis menggunakan filter untuk mencari *tag* yang memiliki deskripsi yang mengandung tulisan “tag”. Setelah memencet tombol yang bertuliskan “Search”, laman ini hanya menampilkan *tag* yang bidang (*field*) description yang mengandung tulisan “tag”. Cuplikan pengujian ini dapat terlihat pada gambar B.1.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/tag-list/?name=&description=tag&creator=`. The page has a navigation bar with links: Project TA, Subscribed, Progress, Dashboard, About Us, Report, Tag, Progress, Organization, and Personal. The main content area is divided into two sections. The left section is a form for creating or filtering tags, with fields for Name, Description (containing 'tag'), and Creator, and a Search button. The right section is an announcement for a 'TOWNHALL MEETING (Aug 8 2020)' at the Community Centre. Below the form, there is a table of tags.

	Name	Creator	Description
<input checked="" type="checkbox"/>	tag1	adminHP	tag1 first
<input checked="" type="checkbox"/>	dummy #2	admin	just a dummy tag

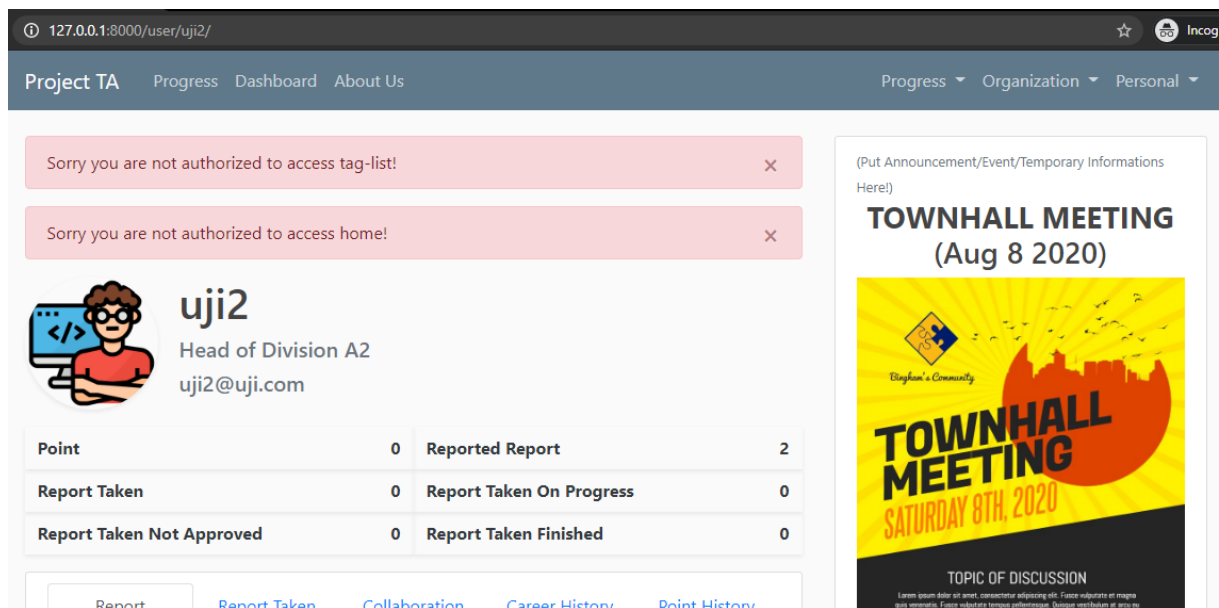
Gambar B.1 Cuplikan layar laman ‘*tag-list*’ pada pengujian pada peramban web pertama

Pada peramban kedua, laman *'tag-list'* tidak dapat diakses dan dialihkan ke laman *'home'*. Namun karena juga tidak memiliki otoritas atas fitur 2, pengguna dialihkan lagi ke laman *'user-reports'* milik dirinya sendiri. Cuplikan pengujian ini dapat terlihat pada gambar B.2.



Gambar B.2 Cuplikan layar laman *'tag-list'* pada pengujian pada peramban web kedua

Pada peramban ketiga, laman *'tag-list'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'tag-list'*. Cuplikan pengujian ini dapat terlihat pada gambar B.3.



Gambar B.3 Cuplikan layar laman *'tag-list'* pada pengujian pada peramban web ketiga

Pada laman *'report-create'* akan diuji aksesibilitasnya, interaksinya, dan pengunggahan datanya. Interaksinya akan diuji dengan mengisi form dan memasukkan berkas gambar. Sedangkan data yang diunggah akan berupa objek *'Report'* baru. Otoritas akun uji2 atas fitur 2 dan fitur 3 akan diberikan lagi karena tidak berhubungan dengan laman ini.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/report-new/'. The page has a navigation bar with links: Project TA, Subscribed, Progress, Dashboard, About Us, Report, Tag, Progress, Organization, and Personal. The main content area is titled 'New Report' and contains the following form elements:

- Title***: A text input field containing 'Partikel Berbahaya di Sektor XX'.
- Content***: A text area containing the text: 'Terdapat kebocoran partikel berbahaya di sektor XX bagian pembakaran bahan YY. Dilampirkan hasil X-Ray pada gambar.'
- Image***: A file upload section with a 'Choose File' button and the filename 'c81e728d9d4c2f636f067f89cc14862c3.jpg'.
- Tag***: A dropdown menu showing 'tag1', 'dummy #2', 'from url', and '1'.
- Urgency***: A dropdown menu set to 'Medium'.
- Importance***: A dropdown menu set to 'High'.
- Upload**: A blue button to submit the report.

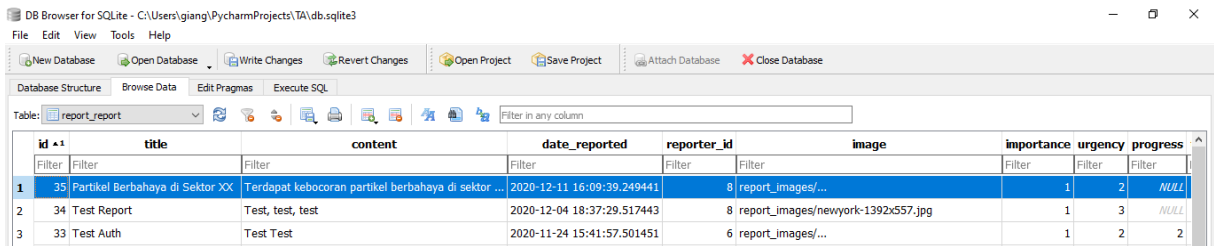
On the right side of the form, there are two promotional banners:

- Townhall Meeting (Aug 8 2020)**: A yellow banner with a city skyline and the text 'TOWNHALL MEETING SATURDAY 8TH, 2020'. Below it, it says 'TOPIC OF DISCUSSION' and 'AT COMMUNITY CENTRE'.
- INTERNAL BADMINTON SPARRING (Aug 17 2020)**: A white banner with the text 'INTERNAL BADMINTON SPARRING (Aug 17 2020)'.

Gambar B.4 Cuplikan layar laman *'report-create'* pada pengujian pada peramban web pertama

Pada peramban pertama, laman *'report-create'* dapat diakses. Pada pengujian ini dicoba untuk berinteraksi dengan elemen terkait dengan mengisi form. Semua kolom isian, kolom pilihan, dan elemen untuk mengunggah gambar dapat bekerja dengan baik. Cuplikan pengujian interaksi dapat terlihat pada gambar B.4. Setelah dapat menampung informasi yang akan diunggah, pengguna memencet tombol bertuliskan "Upload" untuk membuat laporan baru.

Tombol bekerja dengan baik dan objek *'Report'* baru telah tercatat di basis data. Cuplikan basis data pada gambar B.5 menyortir data yang berhasil diunggah.



DB Browser for SQLite - C:\Users\giang\P\pycharmProjects\TA\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

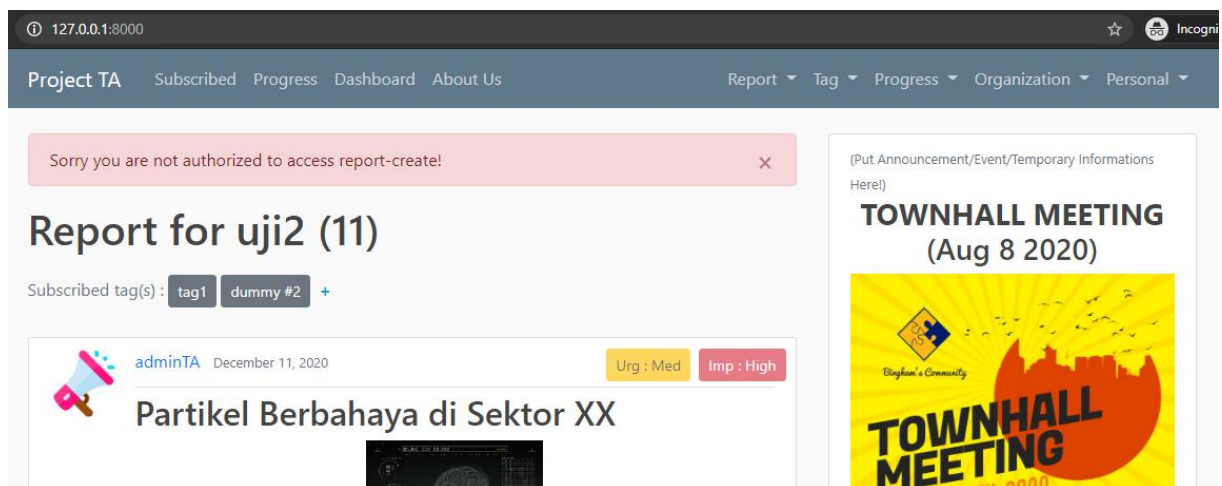
Database Structure Browse Data Edit Pragma Execute SQL

Table: report_report

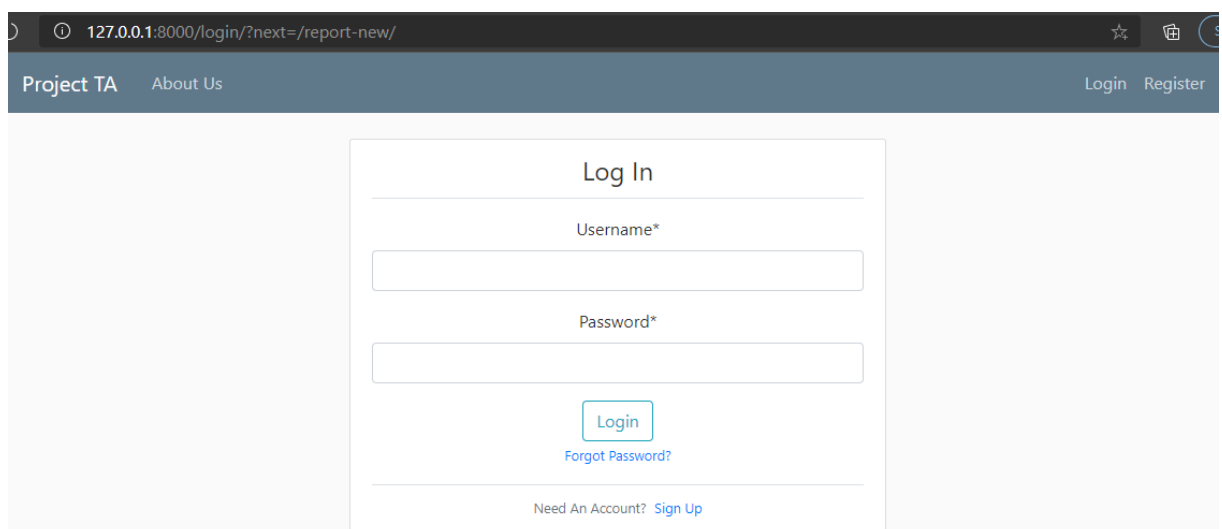
id	title	content	date_reported	reporter_id	image	importance	urgency	progress
1	35 Partikel Berbahaya di Sektor XX	Terdapat kebocoran partikel berbahaya di sektor ...	2020-12-11 16:09:39.249441	8	report_images/...	1	2	NULL
2	34 Test Report	Test, test, test	2020-12-04 18:37:29.517443	8	report_images/newyork-1392x557.jpg	1	3	NULL
3	33 Test Auth	Test Test	2020-11-24 15:41:57.501451	6	report_images/...	1	2	2

Gambar B.5 Cuplikan layar basis data untuk model *'Report'* setelah pengujian laman *'report-create'*

Pada peramban kedua, laman *'report-create'* tidak dapat diakses dan dialihkan ke laman *'home'*. Cuplikan pengujian ini dapat terlihat pada gambar B.6.



Gambar B.5 Cuplikan layar laman *'report-create'* pada pengujian pada peramban web kedua

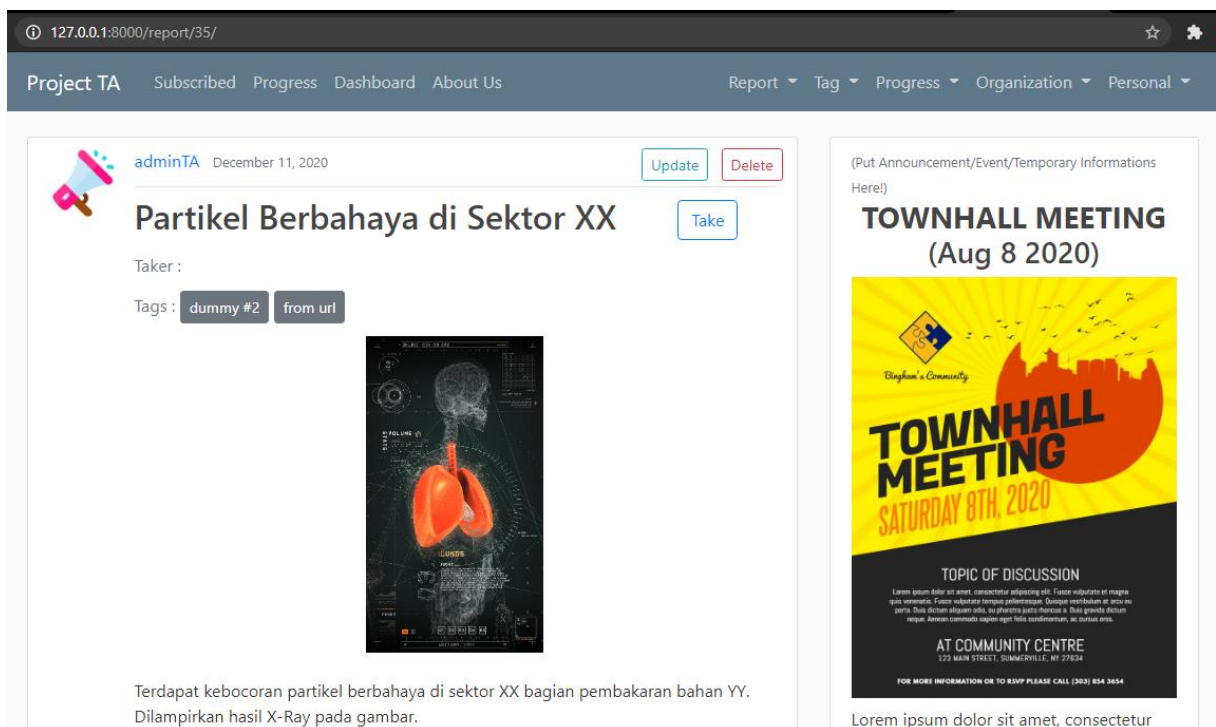


.Gambar B.6 Cuplikan layar laman *'report-create'* pada pengujian pada peramban web ketiga

Pada peramban ketiga, laman *'report-create'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'report-create'*. Cuplikan pengujian ini dapat terlihat pada gambar B.7.

Pada laman *'report-detail'* akan diuji akseibilitasnya dan pengujian interaksi sebagian elemen. Untuk pengujian pengunggahan data tidak akan dilakukan pada sub sub bab ini. Pengunggahan data pada laman ini berkaitan pada fitur 4 “Mengambil Laporan” dan fitur 9 “Berkolaborasi pada Laporan”, maka dari itu aktivitas mengenai interaksi dan pengunggahan data akan dibahas pada sub sub bab yang berkaitan. Interaksi elemen yang diuji juga hanya tombol bertuliskan “Update” dan “Delete”. Interaksi tombol bertuliskan “Take” akan dibahas pada pengujian fitur 4, sedangkan interaksi seksi konten kolaborasi akan dibahas pada pengujian fitur 9.

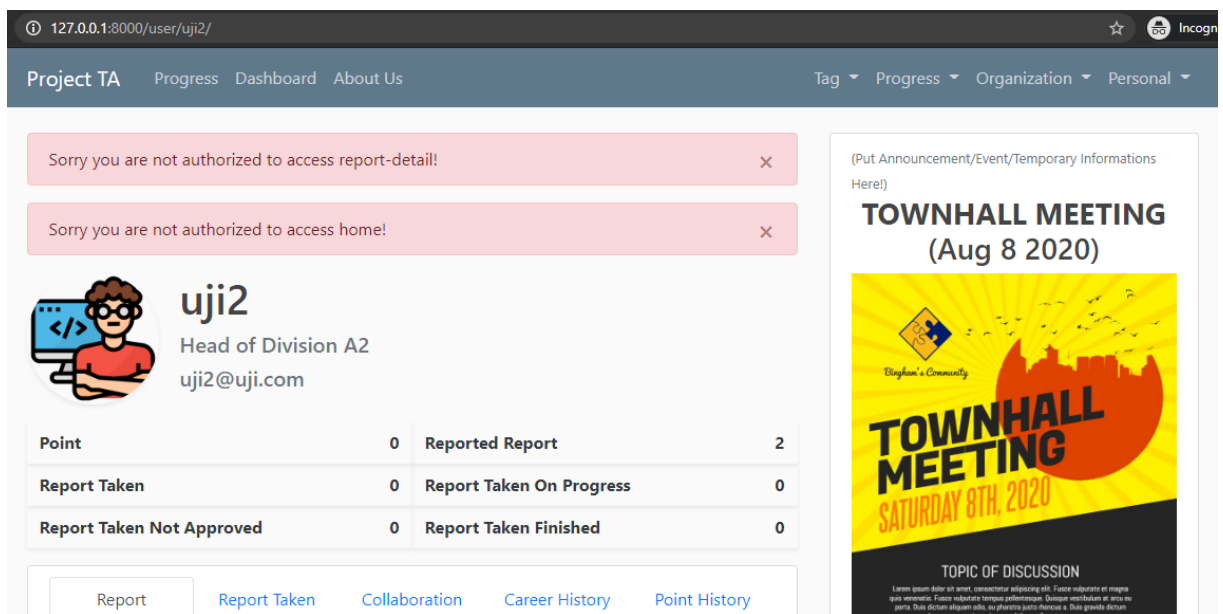
Menurut rancangan, laman *'report-detail'* hanya dapat diakses oleh pengguna yang memiliki otoritas atas fitur 2, pengguna yang membuat laporan tersebut, dan pengguna yang mengambil laporan tersebut. Maka dari itu sebelum pengujian laman ini dimulai, otoritas akun uji2 atas fitur 2 akan dimatikan. Dan sebagai catatan, detail laporan yang diuji adalah laporan yang baru dibuat sebelumnya oleh akun adminTA pada peramban pertama.



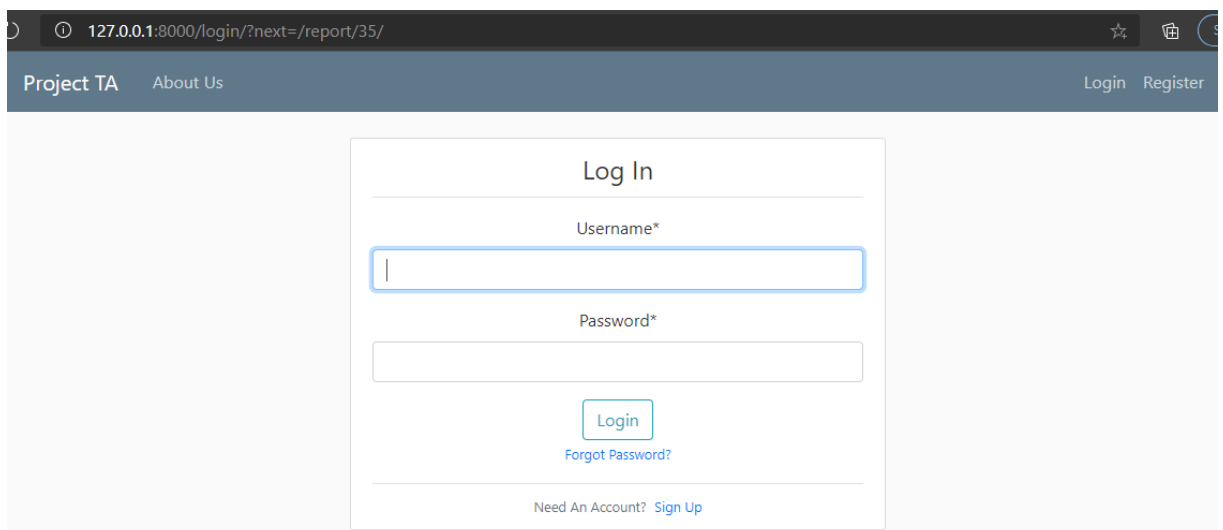
Gambar B.7 Cuplikan layar laman *'report-detail'* pada pengujian pada peramban web pertama

Pada peramban pertama, laman *'report-detail'* dapat diakses. Karena laporan terkait dibuat oleh akun adminTA dan belum diambil, akun adminTA dapat melihat keberadaan tombol bertuliskan “Update” dan “Delete”. Tombol bertuliskan “Update” akan mengalihkan pengguna ke laman *'report-update'*, sedangkan tombol bertuliskan “Delete” akan mengalihkan pengguna ke laman *'report-delete'*. Cuplikan pengujian ini dapat terlihat pada gambar B.8.

Pada peramban kedua, laman *'report-detail'* tidak dapat diakses dan dialihkan ke laman *'home'*. Namun karena otoritas atas fitur 2 juga dicabut pada pengujian ini, pengguna dialihkan lagi ke laman *'user-reports'* milik dirinya sendiri. Cuplikan pengujian ini dapat terlihat pada gambar B.9



Gambar B.8 Cuplikan layar laman *'report-detail'* pada pengujian pada peramban web kedua



Gambar B.9 Cuplikan layar laman *'report-detail'* pada pengujian pada peramban web ketiga

Pada peramban ketiga, laman *'report-detail'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'report-detail'* dari laporan terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.10.

Pada laman *'report-update'* akan diuji aksesibilitas, interaksi elemennya, dan pengunggahan datanya. Pada pengujian interaksi elemen, akan dicoba mengganti data salah satu kolom. Dan data yang diunggah seharusnya mengubah bidang terkait pada objek *'Report'* terkait. Otoritas akun uji2 atas fitur 2 akan diberikan lagi karena tidak berhubungan dengan laman ini.

The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/report/35/update/`. The navigation bar includes links for Project TA, Subscribed, Progress, Dashboard, and About Us, along with dropdown menus for Report, Tag, Progress, Organization, and Personal.

The main content area is titled "New Report" and contains the following form fields:

- Title***: A text input field containing "Partikel Berbahaya di Sektor ABC".
- Content***: A text area containing the text "Terdapat kebocoran partikel berbahaya di sektor XX bagian pembakaran bahan YY. Dilampirkan hasil X-Ray pada gambar." with a green circular icon at the bottom right.
- Image***: A section showing the current image URL (`report_images/c81e728d9d4c2f636f067f89cc14862c3.jpg`) and a "Change:" section with a "Choose File" button and the text "No file chosen".
- Tag***: A dropdown menu showing a list of tags: "tag1", "dummy #2", "from url", and "1".
- Urgency***: A dropdown menu set to "Medium".
- Importance***: A dropdown menu set to "High".
- Upload**: A blue button at the bottom of the form.

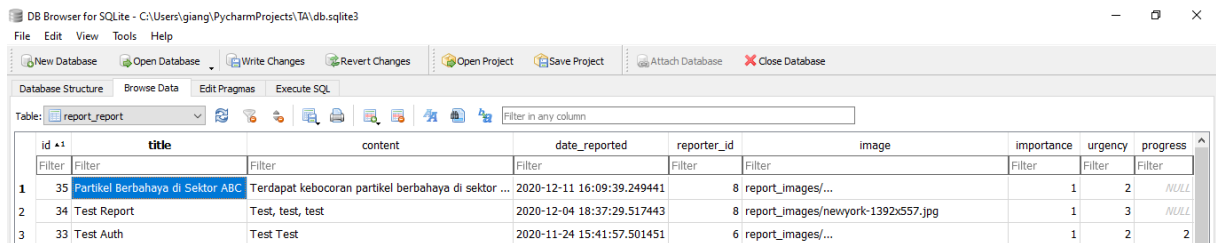
To the right of the form, there are two preview cards for community announcements:

- TOWNHALL MEETING (Aug 8 2020)**: A card with a yellow background and a red sun icon. It includes the text "TOPIC OF DISCUSSION" and "AT COMMUNITY CENTRE 123 MAIN STREET, SUMMERSVILLE, NY 77624".
- INTERNAL BADMINTON SPARRING (Aug 17 2020)**: A card with a blue background and a yellow sun icon. It includes the text "THE FRESHMAN" at the bottom.

Gambar B.10 Cuplikan layar laman *'report-update'* pada pengujian pada peramban web pertama

Menurut rancangan, laman *'report-update'* hanya dapat diakses oleh pengguna yang membuat laporan tersebut bila laporan tersebut belum diambil. Sebagai catatan, detail laporan yang diuji adalah laporan yang baru dibuka detailnya. Dimana pada kasus ini akun adminTA adalah pembuat laporan dan laporan tersebut belum diambil.

Pada peramban pertama, laman *'report-update'* dapat diakses. Elemen yang ada berupa kolom isian, kolom pilihan, dan elemen untuk mengunggah gambar telah terisi dengan data sebelumnya. Namun data tersebut dapat diubah dengan data yang baru. Pada kasus ini pengguna mengubah judul laporan dari “Partikel Berbahaya di Sektor XX” menjadi “Partikel Berbahaya di Sektor ABC”. Cuplikan pengujian ini dapat terlihat pada gambar B.11. Setelah mengubah data yang diperlukan, pengguna memencet tombol bertuliskan “Upload” untuk mengubah data pada laporan terkait. Tombol bekerja dengan baik dan objek *'Report'* baru telah tercatat di basis data. Cuplikan basis data pada gambar B.12 menyoroti data yang berhasil diubah.



DB Browser for SQLite - C:\Users\giang\P\chamProjects\TA\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

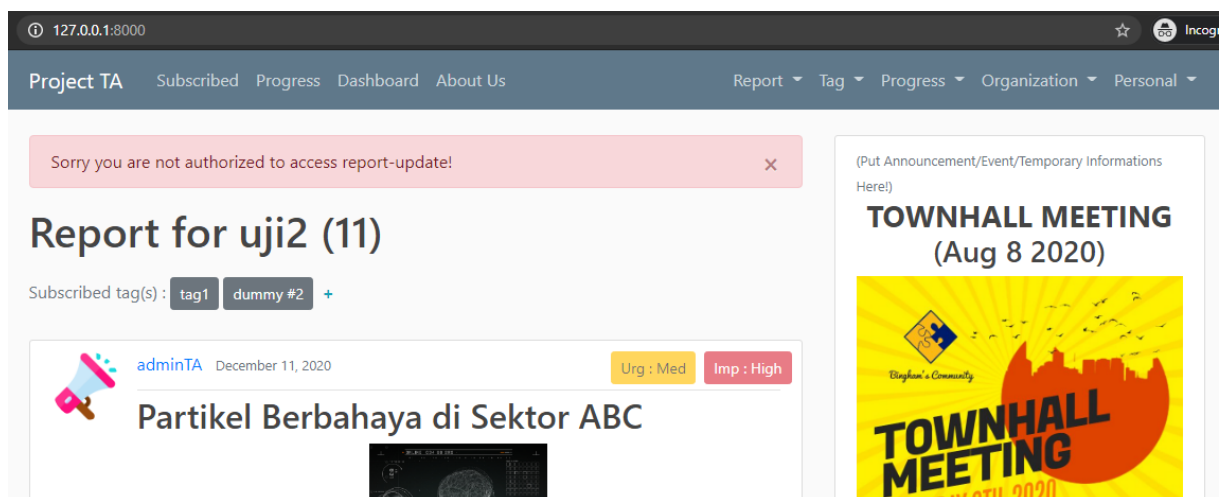
Database Structure Browse Data Edit Pragma Execute SQL

Table: report_report

id	title	content	date_reported	reporter_id	image	importance	urgency	progress
1	35 Partikel Berbahaya di Sektor ABC	Terdapat kebocoran partikel berbahaya di sektor ...	2020-12-11 16:09:39.249441	8	report_images/...	1	2	NULL
2	34 Test Report	Test, test, test	2020-12-04 18:37:29.517443	8	report_images/newyork-1392x557.jpg	1	3	NULL
3	33 Test Auth	Test Test	2020-11-24 15:41:57.501451	6	report_images/...	1	2	2

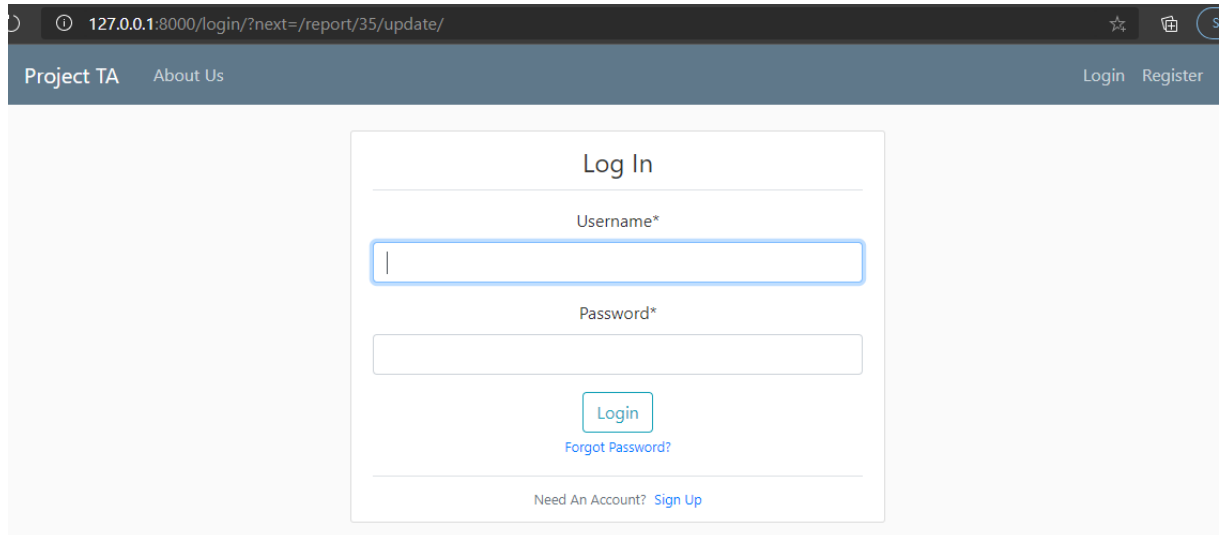
Gambar B.11 Cuplikan layar basis data untuk model *'Report'* setelah pengujian laman *'report-update'*

Pada peramban kedua, laman *'report-update'* tidak dapat diakses dan dialihkan ke laman *'home'*. Cuplikan pengujian ini dapat terlihat pada gambar B.13



Gambar B.12 Cuplikan layar laman *'report-update'* pada pengujian pada peramban web kedua

Pada peramban ketiga, laman *'report-update'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'report-update'* dari laporan terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.14.

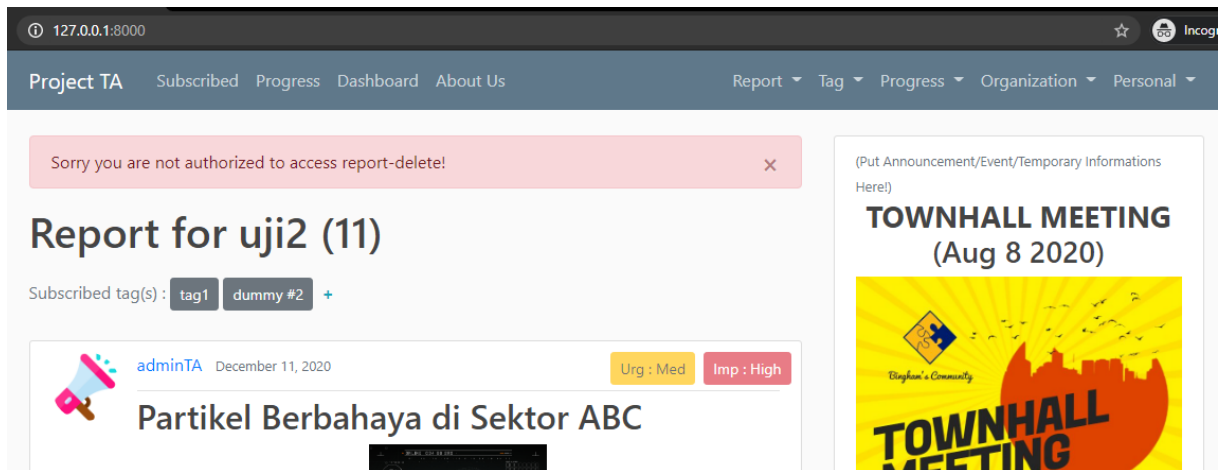


Gambar B.13 Cuplikan layar laman *'report-update'* pada pengujian pada peramban web ketiga

Pada laman *'report-delete'* akan diuji aksesibilitas, interaksi elemennya, dan pengunggahan datanya. Pada pengujian interaksi elemen, dapat dipencet tombol untuk menghapus laporan. Dan data yang diunggah adalah data mengenai penghapusan objek *'Report'* terkait.

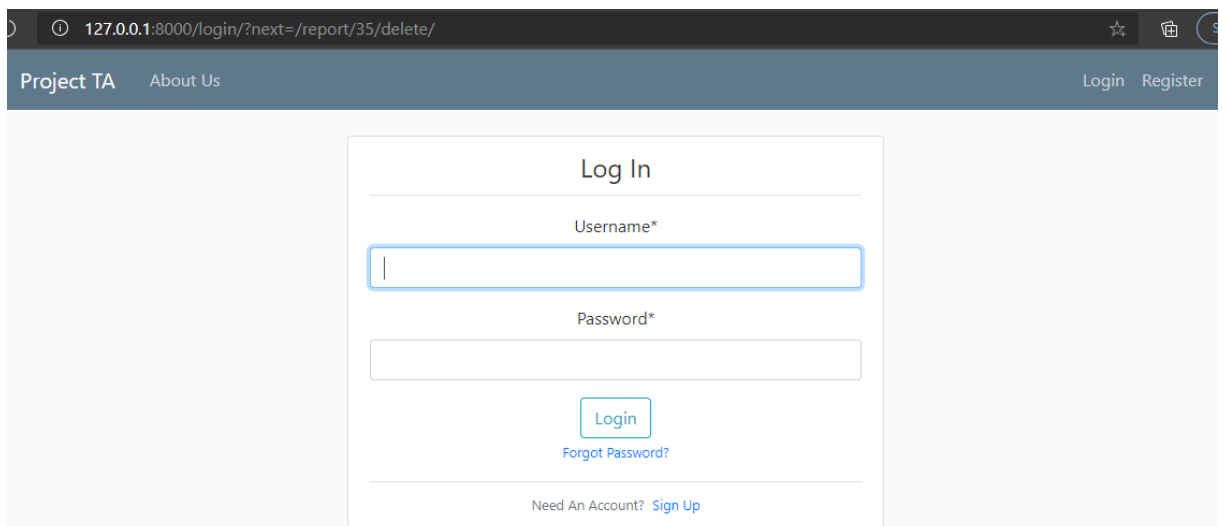
Menurut rancangan, laman *'report-delete'* hanya dapat diakses oleh pengguna yang membuat laporan tersebut bila belum diambil atau pengguna yang mengambil laporan tersebut bila telah diambil. Sebagai catatan, detail laporan yang diuji untuk dihapus adalah laporan yang baru diubah datanya. Dimana pada kasus ini akun adminTA adalah pembuat laporan dan laporan tersebut belum diambil. Pada kasus ini, menurut rancangan pengguna dengan akun adminTA pada peramban pertama dapat menghapus laporan, sehingga laman *'report-delete'* akan diuji pada peramban kedua dan peramban ketiga terlebih dahulu.

Pada peramban kedua, laman *'report-delete'* tidak dapat diakses dan dialihkan ke laman *'home'*. Cuplikan pengujian ini dapat terlihat pada laman pada gambar B.15.



Gambar B.14 Cuplikan layar laman *'report-delete'* pada pengujian pada peramban web kedua

Pada peramban ketiga, laman *'report-delete'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'report-delete'* dari laporan terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.16.



Gambar B.15 Cuplikan layar laman *'report-delete'* pada pengujian pada peramban web ketiga

Pada peramban pertama, laman *'report-delete'* dapat diakses. Elemen berupa tombol untuk menghapus atau mengurungkan niat menghapus akan diuji. Tombol untuk mengurungkan niat menghapus dapat bekerja dengan baik, pengguna dialihkan ke laman *'report-detail'* dari laporan terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.17. Setelah itu diuji tombol untuk menghapus laporan terkait. Laporan terkait telah terhapus dari basis data dan pengguna dialihkan ke laman *'home'*. Cuplikan basis data pada gambar B.18 memperlihatkan laporan dengan id=35 berjudul “Partikel Berbahaya di Sektor ABC” telah terhapus dari basis data.

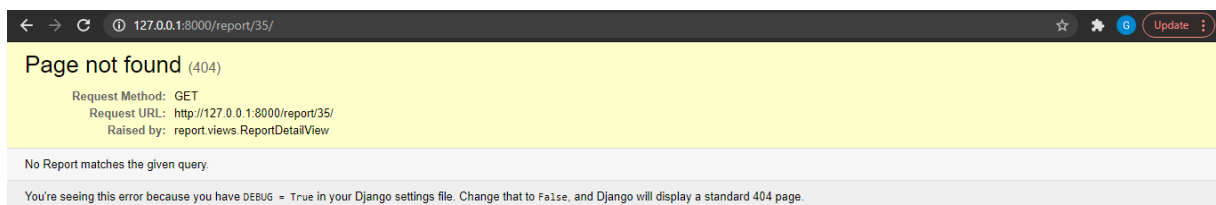


Gambar B.16 Cuplikan layar laman *'report-delete'* pada pengujian pada peramban web pertama

id	title	content	date_reported	reporter_id	image	importance	urgency	progress	taker
34	Test Report	Test, test, test	2020-12-04 18:37:29.517443	8	report_images/newyork-1392x557.jpg	1	3	NULL	NULL
33	Test Auth	Test Test	2020-11-24 15:41:57.501451	6	report_images/...	1	2	2	

Gambar B.17 Cuplikan layar basis data untuk model *'Report'* setelah pengujian laman *'report-delete'*

Laporan dengan id=35 berjudul “Partikel Berbahaya di Sektor ABC” telah dihapus sehingga datanya tidak terdapat di basis data. Laporan tersebut tidak dapat diakses lagi. Namun bila ada pengguna yang berusaha mengakses laporan tersebut akan muncul peringatan “*Page not found (404)*” seperti pada gambar B.19.



Gambar B.18 Pesan galat 404 untuk objek *'Report'* yang tidak ada

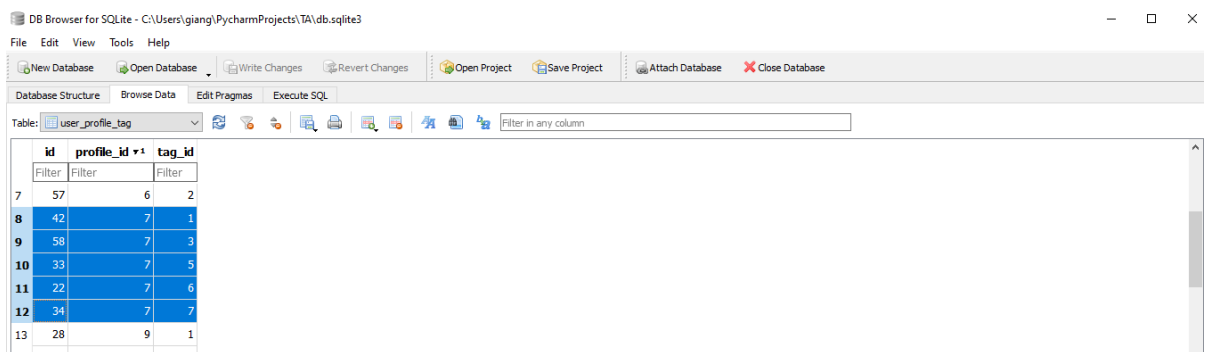
B.2. Pengujian Fitur 2 “Melihat Laporan” (View Reports)

Pengujian pada fitur 2 “Melihat Laporan” akan meliputi pengujian laman *'home'*, *'all-reports'*, *'tag-reports'*, *'user-reports'*, dan *'user-collab'*. Sedangkan laman *'tag-list'*, dan *'report-detail'* yang memang merupakan bagian dari fitur 2 tidak akan dibahas lagi karena sudah diuji pada pengujian fitur 1 yang dijabarkan di sub sub bab B.1. Seperti pada sub bab B.1 pengujian pada fitur 2 akan terdiri dari tiga jenis pengujian (bila memungkinkan) pada setiap laman seperti yang disebutkan di atas melalui tiga peramban web. Sebelum dilakukan pengujian, otoritas akun uji2 atas fitur 2 akan dimatikan, sedangkan otoritas akun uji2 atas fitur lain tetap diberikan.

Pada laman ‘home’ akan diuji aksesibilitas dan interaksi elemennya. Interaksi elemen yang diuji adalah kemunculan elemen bertuliskan “Taken” hanya pada laporan yang sudah diambil dan muncul tombol bertuliskan *tag* yang telah dilanggan pengguna. Tidak ada pengujian pengunggahan data karena tidak ada data yang dapat diunggah dari pengguna pada laman ‘home’ untuk mengubah data pada basis data.

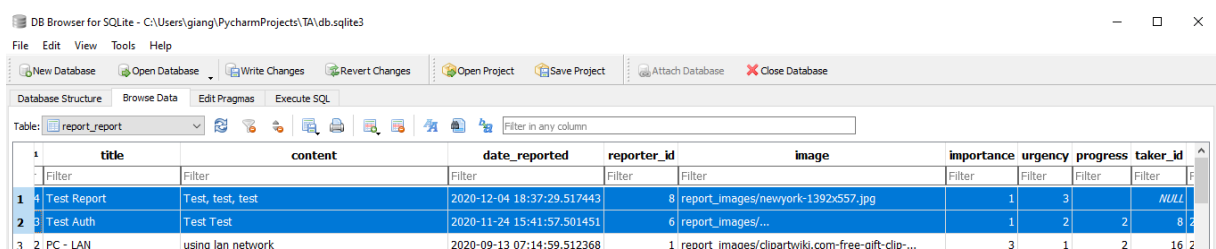
Pada peramban pertama, laman ‘home’ dapat diakses. Pada pengujian ini dapat terlihat bahwa terdapat lima *tag* yang dilanggan akun adminTA. Pada laman ini juga terlihat bahwa laporan dengan judul ‘TestReport’ belum diambil karena tidak ada elemen bertuliskan ‘Taken’, sedangkan laporan yang berjudul ‘TestAuth’ terlihat telah diambil. Cuplikan pengujian interaksi dapat terlihat pada gambar B.22.

Sebagai bukti mengenai tombol *tag*, akun adminTA memiliki profil dengan pk=7. Pada basis data hubungan banyak-dengan-banyak (*many to many*) objek ‘Profile’ dengan objek ‘Tag’ seperti pada gambar B.20 terlihat profile dengan id=7 memiliki 5 pasangan tag. Dan sebagai bukti mengenai elemen bertuliskan “Taken”, pada basis data objek ‘Report’ seperti pada gambar B.21. Terlihat pada baris laporan dengan judul “Test Report” (baris 1), kolom *taker_id* nya kosong (null). Sedangkan pada baris laporan dengan judul “Test Auth” (baris 2), kolom *taker_id*-nya telah terisi oleh kunci primer (*primary key*) dari objek ‘User’ dari akun yang mengambil.



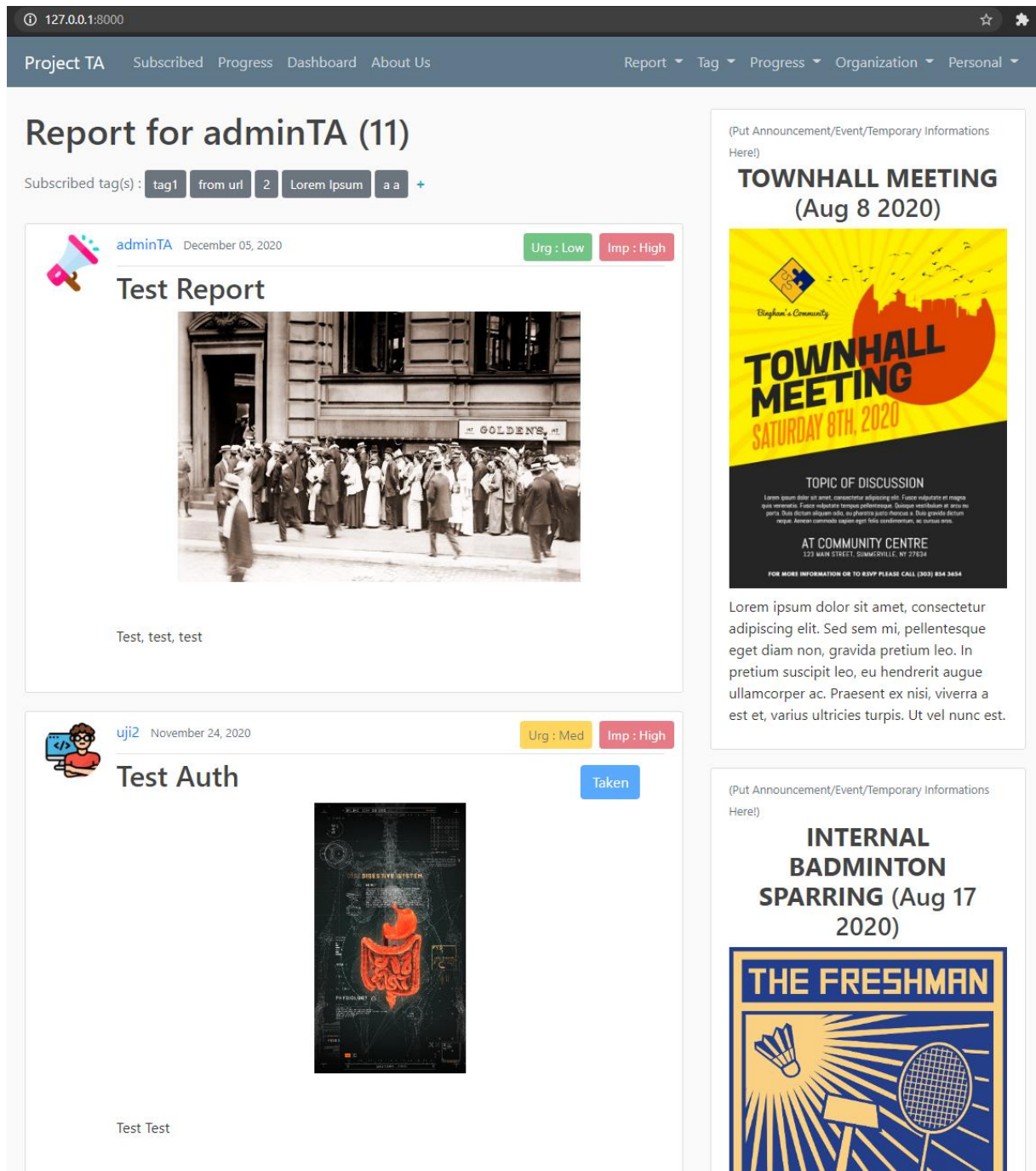
id	profile_id	tag_id
7	57	6
8	42	7
9	58	7
10	33	7
11	22	7
12	34	7
13	28	9

Gambar B.19 Cuplikan layar basis data untuk hubungan model ‘Profile’ dengan model ‘Tag’



id	title	content	date_reported	reporter_id	image	importance	urgency	progress	taker_id
1	Test Report	Test, test, test	2020-12-04 18:37:29.517443	8	report_images/newyork-1392x557.jpg	1	3		NULL
2	Test Auth	Test Test	2020-11-24 15:41:57.501451	6	report_images/...	1	2	2	8

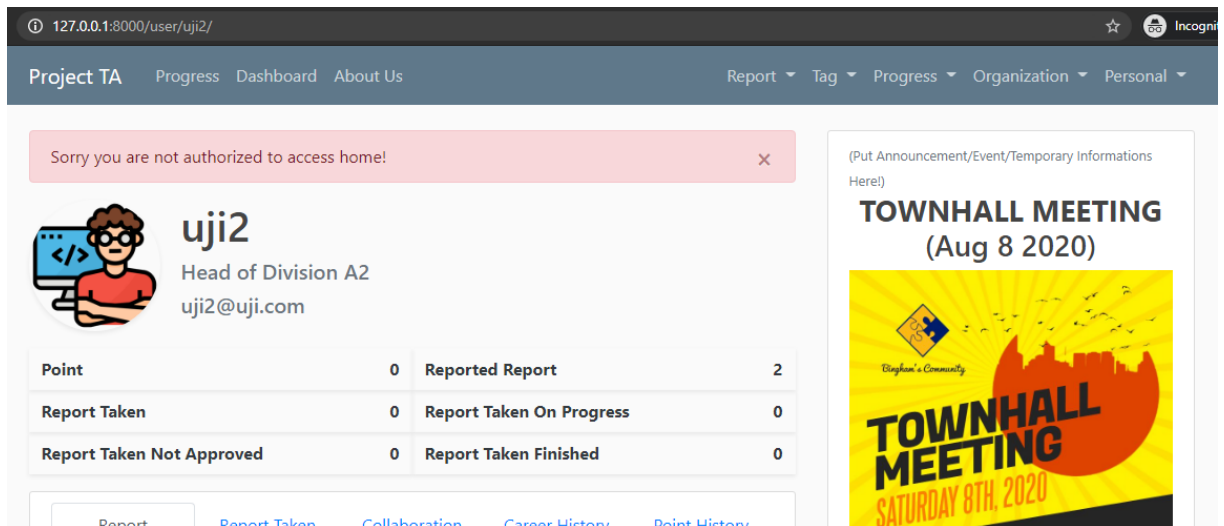
Gambar B.20 Cuplikan layar basis data untuk model ‘Report’



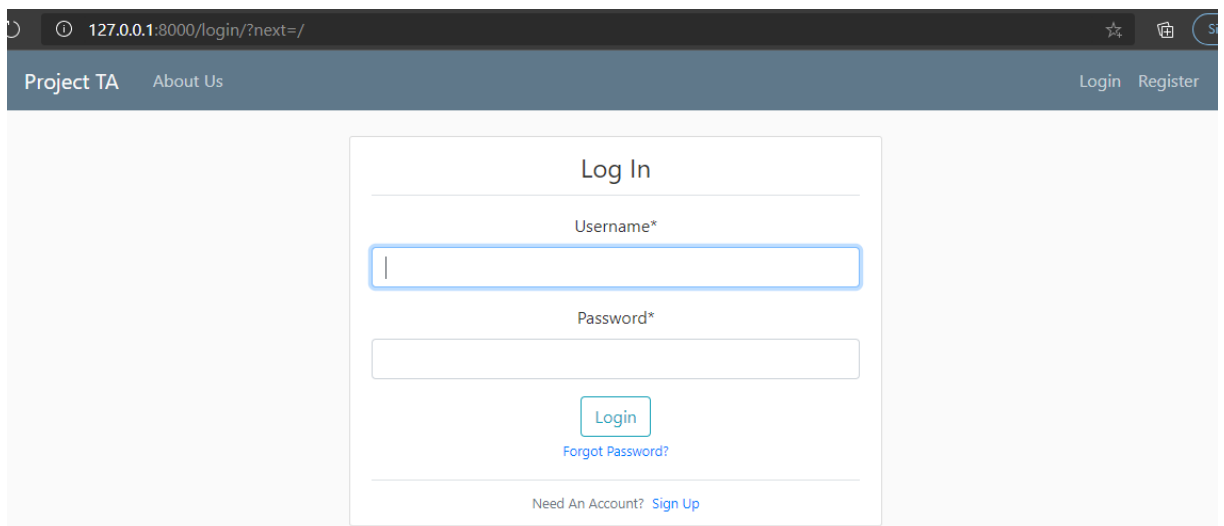
Gambar B.21 Cuplikan layar laman 'home' pada pengujian pada peramban web pertama

Pada peramban kedua, laman 'home' tidak dapat diakses dan dialihkan ke laman 'user-reports' terkait pengguna itu sendiri. Cuplikan pengujian ini dapat terlihat pada gambar B.23.

Pada peramban ketiga, laman 'home' tidak dapat diakses dan dialihkan ke laman 'login'. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman 'home'. Cuplikan pengujian ini dapat terlihat pada gambar B.24.



Gambar B.22 Cuplikan layar laman 'home' pada pengujian pada peramban web kedua

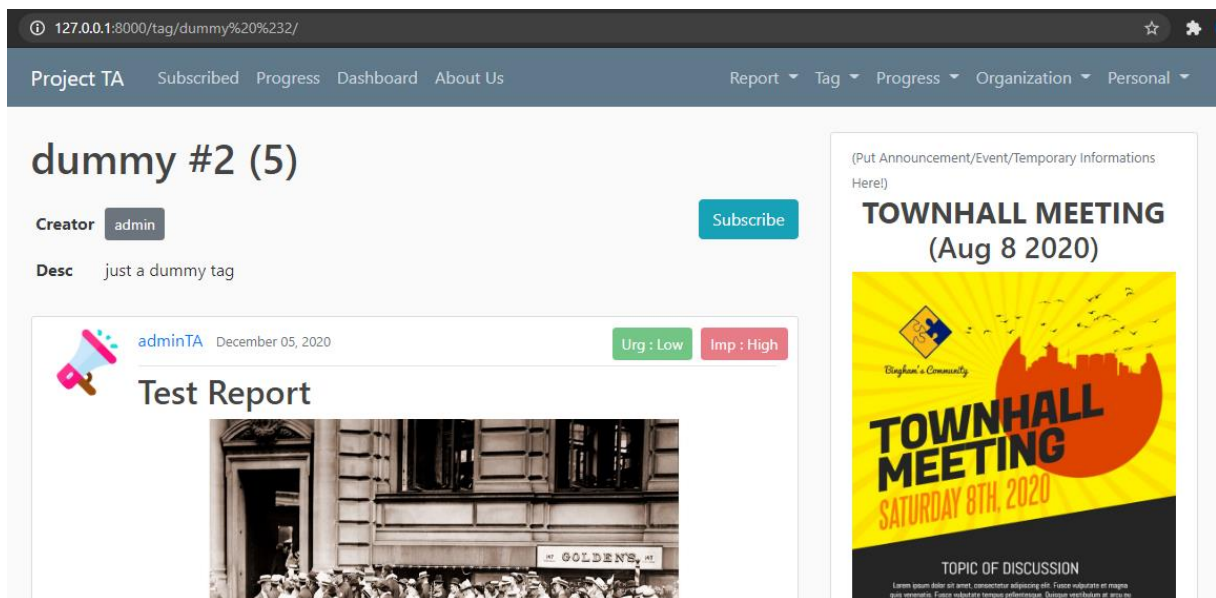


Gambar B.23 Cuplikan layar laman 'home' pada pengujian pada peramban web ketiga

Pada laman 'tag-reports' akan diuji aksesibilitasnya, interaksi elemennya, dan pengunggahan datanya. Elemen yang diuji adalah tombol untuk melanggan atau berhenti melanggan tag terkait. Sedangkan data yang diunggah akan berupa perintah untuk melanggan yang kemudian akan diproses dan masuk pada basis data hubungan banyak-dengan-banyak (*many to many*) objek 'Profile' dengan objek 'Tag'.

Pada kasus pengujian laman 'tag-reports' ini, akan diuji laman untuk tag dengan nama "dummy #2". Dimana pada kasus ini akun adminTA belum melanggan tag ini, hal tersebut dapat terlihat pada gambar B.22. Selain itu dicabut otoritas akun uji2 atas fitur 3 karena laman ini terkait dengan fitur 3.

Pada peramban pertama, laman ‘tag-reports’ dapat diakses. Pada kasus ini juga terlihat tombol berwarna gelap bertuliskan “Subscribe”, hal tersebut menunjukkan bahwa akun adminTA belum melanggan tag ini. Cuplikan pengujian ini dapat terlihat pada gambar B.25. Akan diuji interaksi elemen tersebut dengan memencet tombol tersebut. Karena tombol telah dipencet, berarti akun adminTA telah melanggan tag bernama “dummy #2”. Tombol bekerja dengan baik karena berubah warna menjadi terang dan menjadi bertuliskan “Subscribed”. Dan tag telah berhasil dilanggan akun adminTA, terlihat dengan bertambahnya tag yang terhubung dengan profil akun adminTA dengan pk=7 seperti pada gambar B.26. Total tag yang dilanggan profil dengan pk=7 (akun adminTA) menjadi 6 buah, yang sebelumnya 5 buah pada pengujian laman ‘home’.



Gambar B.24 Cuplikan layar laman ‘tag-reports’ pada pengujian pada peramban web pertama

DB Browser for SQLite - C:\Users\giangl\PycharmProjects\TA\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

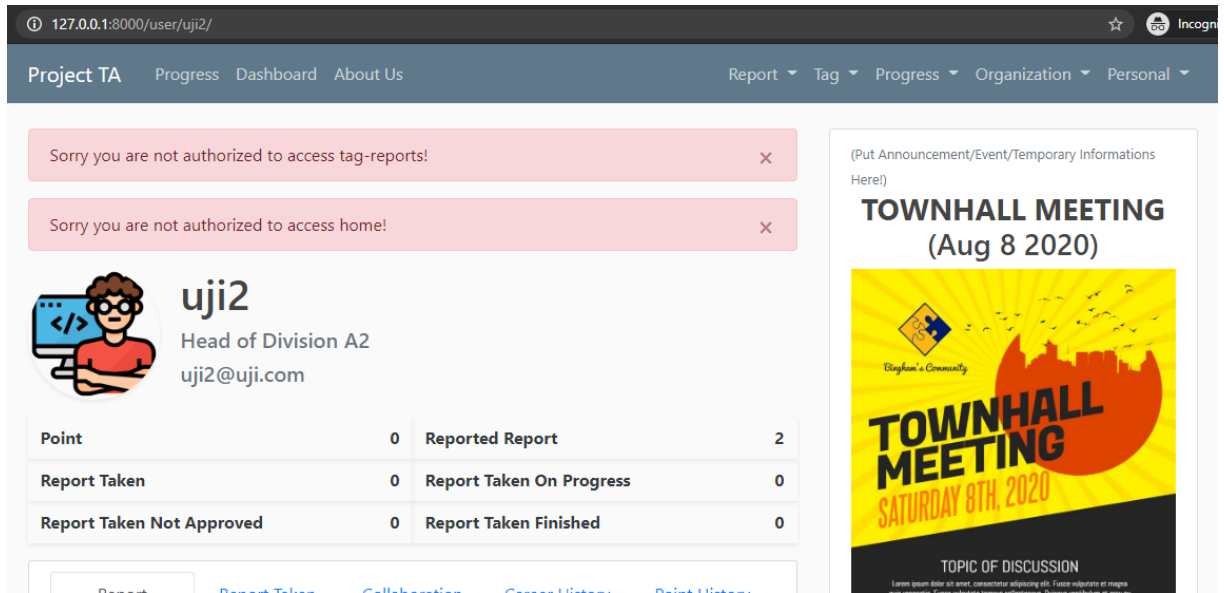
Database Structure Browse Data Edit Pragma Execute SQL

Table: user_profile_tag

	id	profile_id	tag_id
9	28	9	1
10	34	7	7
11	22	7	6
12	33	7	5
13	58	7	3
14	59	7	2
15	42	7	1
16	57	6	2

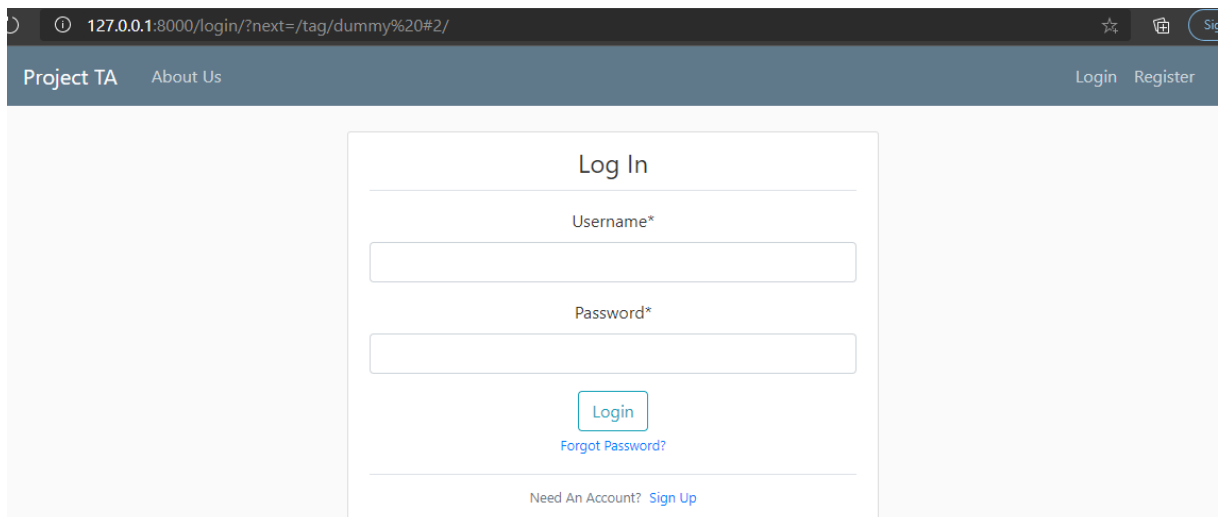
Gambar B.25 Cuplikan layar basis data untuk hubungan model ‘Profile’ dengan model ‘Tag’ setelah pengujian laman ‘tag-reports’

Pada peramban kedua, laman *'tag-reports'* tidak dapat diakses dan dialihkan ke laman *'home'*. Namun karena juga tidak memiliki otoritas atas fitur 2, pengguna dialihkan lagi ke laman *'user-reports'* milik dirinya sendiri. Cuplikan pengujian ini dapat terlihat pada gambar B.27.



Gambar B.26 Cuplikan layar laman *'tag-reports'* pada pengujian pada peramban web kedua

Pada peramban ketiga, laman *'tag-reports'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'tag-reports'* dari *tag* terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.28.

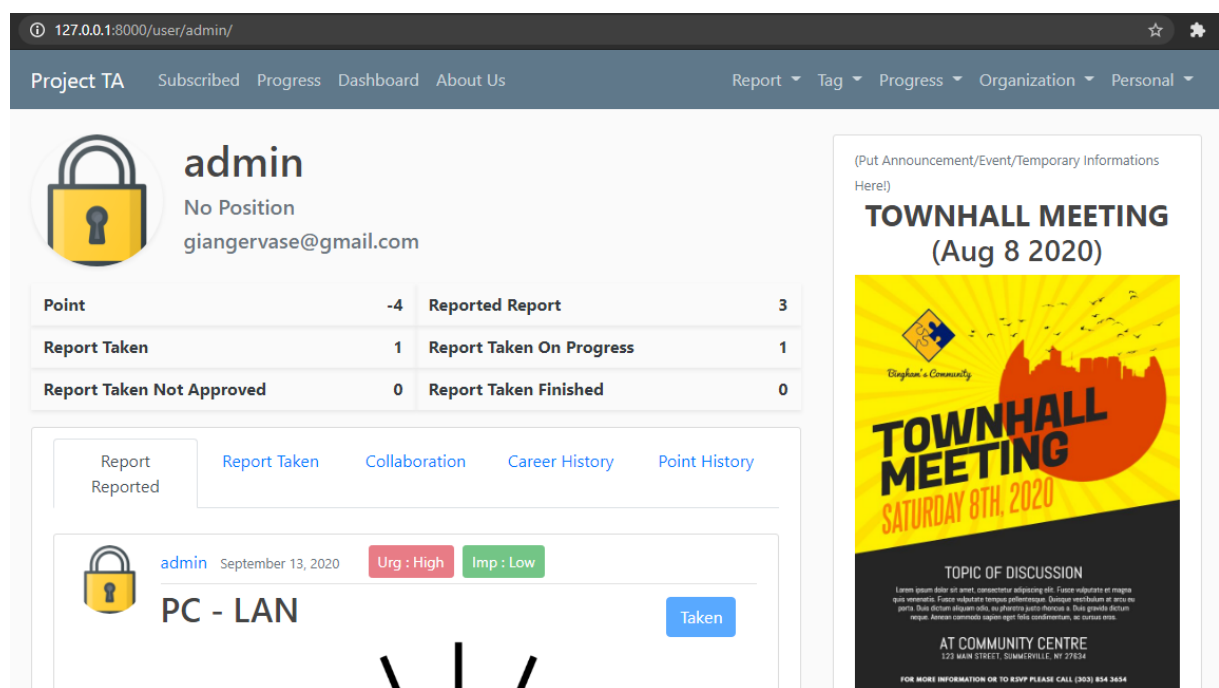


Gambar B.27 Cuplikan layar laman *'tag-reports'* pada pengujian pada peramban web ketiga

Pada laman *'user-reports'* akan diuji aksesibilitas dan interaksi elemennya. Pengunggahan data tidak diuji karena tidak ada data yang dapat diunggah pengguna untuk mengubah data pada basis data. Sedangkan elemen yang diuji interaksinya adalah tombol yang mengarahkan ke laman *'user-taken'*, *'user-collab'*, *'user-career'*, dan *'user-point'*.

Pada kasus pengujian ini akan digunakan laman *'user-reports'* untuk akun bernama pengguna admin. Selain itu, karena laman ini tidak berhubungan lagi dengan fitur 3, maka otoritas akun uji2 atas fitur 3 diberikan lagi. Namun otoritas akun uji2 atas fitur 5 akan dimatikan karena laman ini berhubungan dengan fitur 5. Hal tersebut berarti akun uji2 masih memiliki otoritas atas fitur 1, fitur 3, fitur 4, fitur 6, fitur 7, fitur 8, dan fitur 9.

Pada peramban pertama, laman *'user-reports'* dapat diakses. Pada pengujian interaksi elemen terlihat terdapat 4 tombol bertuliskan "Report Taken", "Collaboration", "Career History" dan "Point History" karena akun adminTA memiliki otoritas atas semua fitur, terlihat keempat tombol tersebut. Dan keempat tombol bekerja dengan baik untuk mengalihkan ke laman lain mengenai pengguna terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.29.



Gambar B.28 Cuplikan layar laman *'user-reports'* pada pengujian pada peramban web pertama

Pada peramban kedua, laman *'user-reports'* tidak dapat diakses dan dialihkan ke laman *'user-taken'* dari pengguna terkait. Pengalihan berhenti di laman *'user-taken'* karena akun uji2 memiliki otoritas atas fitur 4. Bila akun uji2 tidak memiliki otoritas atas fitur 4, akun uji2 akan dialihkan lagi ke laman lain. Pada laman *'user-taken'* dapat terlihat terdapat 3 tombol

bertuliskan “Collaboration”, “Career History” dan “Point History”, namun tidak ada tombol bertuliskan “Report Reported”. Ketiadaan tombol “Report Reported” dikarenakan pengguna tidak dapat mengakses laman ‘*user-reports*’, jadi tidak perlu tampilkan tombol yang dapat mengalihkan pengguna ke laman ‘*user-reports*’. Cuplikan pengujian ini dapat terlihat pada gambar B.30.

The screenshot shows a web application interface. At the top, there is a navigation bar with links: Project TA, Progress, Dashboard, About Us, Report, Tag, Progress, Organization, and Personal. Below the navigation bar, the user profile for 'admin' is displayed, showing a lock icon, the name 'admin', 'No Position', and the email 'giangervase@gmail.com'. A table shows statistics for reports: Point (-4), Reported Report (3), Report Taken (1), Report Taken On Progress (1), Report Taken Not Approved (0), and Report Taken Finished (0). Below the table, there are tabs for Report Taken, Collaboration, Career History, and Point History. The 'Report Taken' tab is active, showing a report titled 'Socialism ??' by 'Ujihp2' on September 13, 2020, with 'Urg : Med' and 'Imp : Low' status. A 'Taken' button is visible next to the report title. To the right of the report, there is a 'TOWNHALL MEETING' announcement for August 8, 2020, at the Community Centre, with a poster image.

Gambar B.29 Cuplikan layar laman ‘*user-reports*’ pada pengujian pada peramban web kedua

The screenshot shows a web application interface with a login form. The navigation bar includes links: Project TA, About Us, Login, and Register. The login form is titled 'Log In' and contains fields for 'Username*' and 'Password*'. Below the password field is a 'Login' button and a link for 'Forgot Password?'. At the bottom of the form, there is a link for 'Need An Account? Sign Up'.

Gambar B.30 Cuplikan layar laman ‘*user-reports*’ pada pengujian pada peramban web ketiga

Pada peramban ketiga, laman *'user-reports'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'user-reports'* dari pengguna terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.31.

Pada laman *'user-collab'* akan diuji aksesibilitas dan interaksi elemennya. Tidak dimungkinkan aktivitas pengunggahan data, makada dari itu tidak diuji. Sedangkan elemen yang diuji adalah tombol yang mengarahkan ke laman lain terkait pengguna ini, serupa dengan pada pengujian pada laman *'user-reports'*.

Pada kasus pengujian ini akan digunakan laman *'user-collab'* untuk akun bernama pengguna admin. Selain itu, otoritas akun uji2 atas fitur 4 akan dimatikan karena laman ini berhubungan dengan fitur 9. Hal tersebut berarti akun uji2 masih memiliki otoritas atas fitur 1, fitur 3, fitur 4, fitur 6, fitur 7, dan fitur 8.

Pada peramban pertama, laman *'user-collab'* dapat diakses. Pada pengujian interaksi elemen terlihat terdapat 4 tombol bertuliskan "Report Reported", "Report Taken", "Career History" dan "Point History" karena akun adminTA memiliki otoritas atas semua fitur, terlihat keempat tombol tersebut. Dan keempat tombol bekerja dengan baik untuk mengalihkan ke laman lain mengenai pengguna terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.32.

The screenshot shows a web application interface for a user named 'admin'. The user's role is 'Head of Unit Pengecoran Rangka' and their email is 'giangervase@gmail.com'. A table displays statistics for the user's reports. The sidebar contains buttons for 'Report Reported', 'Report Taken', 'Collaboration', 'Career History', and 'Point History'. A townhall meeting announcement for August 8, 2020, is displayed on the right side of the page.

Point	-8	Reported Report	3
Report Taken	1	Report Taken On Progress	1
Report Taken Not Approved	0	Report Taken Finished	0

Navigation buttons: Report Reported, Report Taken, Collaboration, Career History, Point History.

Townhall Meeting Announcement (Aug 8 2020):

TOPIC OF DISCUSSION

AT COMMUNITY CENTRE
123 MAIN STREET, SUMMERSVILLE, NY 27634

Gambar B.31 Cuplikan layar laman *'user-collab'* pada pengujian pada peramban web pertama

Pada peramban kedua, laman *'user-collab'* tidak dapat diakses dan dialihkan ke laman *'user-career'* dari pengguna terkait. Pengalihan berhenti di laman *'user-career'* karena akun

uji2 memiliki otoritas atas fitur 6. Bila akun uji2 tidak memiliki otoritas atas fitur 6, akun uji2 akan dialihkan lagi ke laman lain. Pada laman ‘*user-career*’ dapat terlihat terdapat 2 tombol bertuliskan “Report Taken” dan “Point History”, namun tidak ada tombol bertuliskan “Report Reported” dan “Collaboration”. Ketiadaan kedua tombol tersebut dikarenakan pengguna tidak dapat mengakses laman ‘*user-reports*’ dan ‘*user-collab*’, jadi tidak perlu tampilkan tombol yang dapat mengalihkan pengguna ke kedua laman tersebut. Cuplikan pengujian ini dapat terlihat pada gambar B.33.

Gambar B.32 Cuplikan layar laman ‘*user-collab*’ pada pengujian pada peramban web kedua

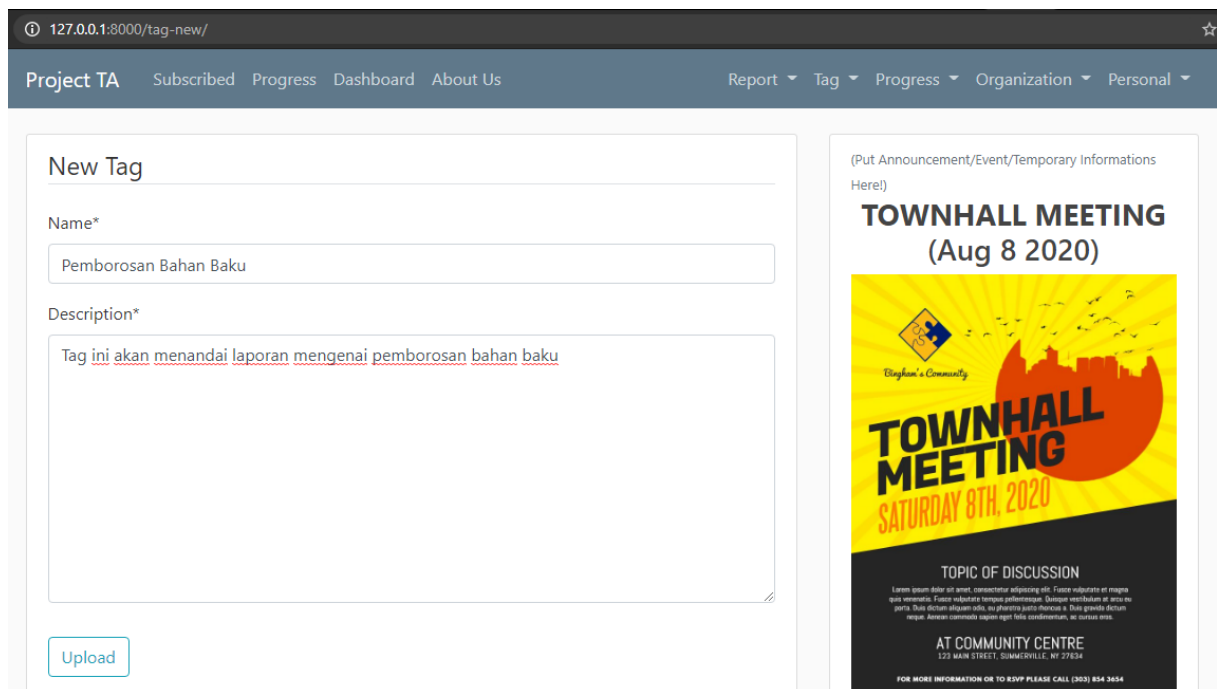
Gambar B.33 Cuplikan layar laman ‘*user-collab*’ pada pengujian pada peramban web ketiga

Pada peramban ketiga, laman ‘*user-collab*’ tidak dapat diakses dan dialihkan ke laman ‘*login*’. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman ‘*user-collab*’ dari pengguna terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.34.

B.3. Pengujian Fitur 3 “Membuat Tag” (*Create a Tag*)

Pengujian pada fitur 3 “Membuat Laporan” akan berupa pengujian laman ‘*tag-create*’. Sedangkan laman ‘*tag-reports*’ dan ‘*tag-list*’ tidak akan diuji lagi di sub sub bab ini karena pengujiannya telah dibahas di sub sub bab B.1 dan B.2. Seperti pada sub bab sebelumnya, pengujian pada fitur 3 akan terdiri dari tiga jenis pengujian (bila memungkinkan) pada laman ‘*tag-create*’ melalui tiga peramban web. Sebelum dilakukan pengujian, otoritas akun uji2 atas fitur 3 akan dimatikan, sedangkan otoritas akun uji2 atas fitur lain tetap diberikan.

Pada laman ‘*tag-create*’ akan diuji aksesibilitasnya, interaksi elemennya, dan pengunggahan datanya. Pada pengujian interaksi elemen, akan diuji kolom isian dan tombol untuk membuat *tag* baru. Sedangkan pada pengujian pengunggahan data, akan diperiksa apakah objek ‘*Tag*’ yang baru dibuat telah tercatat pada basis data.

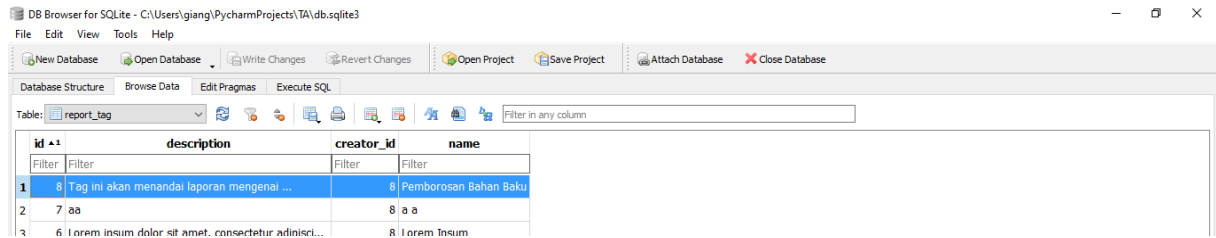


The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/tag-new/'. The browser's navigation bar includes links for 'Project TA', 'Subscribed', 'Progress', 'Dashboard', and 'About Us'. On the right side of the navigation bar, there are dropdown menus for 'Report', 'Tag', 'Progress', 'Organization', and 'Personal'. The main content area is divided into two panels. The left panel, titled 'New Tag', contains a form with two input fields: 'Name*' and 'Description*'. The 'Name*' field contains the text 'Pemborosan Bahan Baku'. The 'Description*' field contains the text 'Tag ini akan menandai laporan mengenai pemborosan bahan baku'. Below the description field is an 'Upload' button. The right panel displays a poster for a 'TOWNHALL MEETING (Aug 8 2020)'. The poster features a yellow background with a red sun and a blue puzzle piece logo. The text on the poster includes 'Put Announcement/Event/Temporary Informations Here!', 'TOWNHALL MEETING (Aug 8 2020)', 'Elyahon's Community', 'TOWNHALL MEETING SATURDAY 8TH, 2020', 'TOPIC OF DISCUSSION', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce eugetur et magna quis interdum. Fusce volutpat tempus pellentesque. Doneque vestibulum et ante eu porta. Duis dictum aliquam odio, in pharetra justo metus a. Duis gravida dictum magna. Aenean consectetur ipsum eget felis condimentum, ac tuncus tunc.', 'AT COMMUNITY CENTRE 123 MAIN STREET, SUMMERSVILLE, NY 27634', and 'FOR MORE INFORMATION OR TO RSVP PLEASE CALL (202) 834 3434'.

Gambar B.34 Cuplikan layar laman ‘*tag-create*’ pada pengujian pada peramban web pertama

Pada peramban pertama, laman ‘*tag-create*’ dapat diakses. Pada pengujian interaksi elemen didapat bahwa setiap kolom isian dapat diisi. Cuplikan pengujian interaksi dapat terlihat

pada gambar B.35. Setelah yakin dengan informasi yang dimasukan, dipencet tombol bertuliskan “Upload ” untuk membuat objek ‘Tag’ baru. Tombol bekerja dengan baik dan objek ‘Tag’ baru telah tercatat di basis data seperti pada gambar B.36.



DB Browser for SQLite - C:\Users\giang\PycharmProjects\TA\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

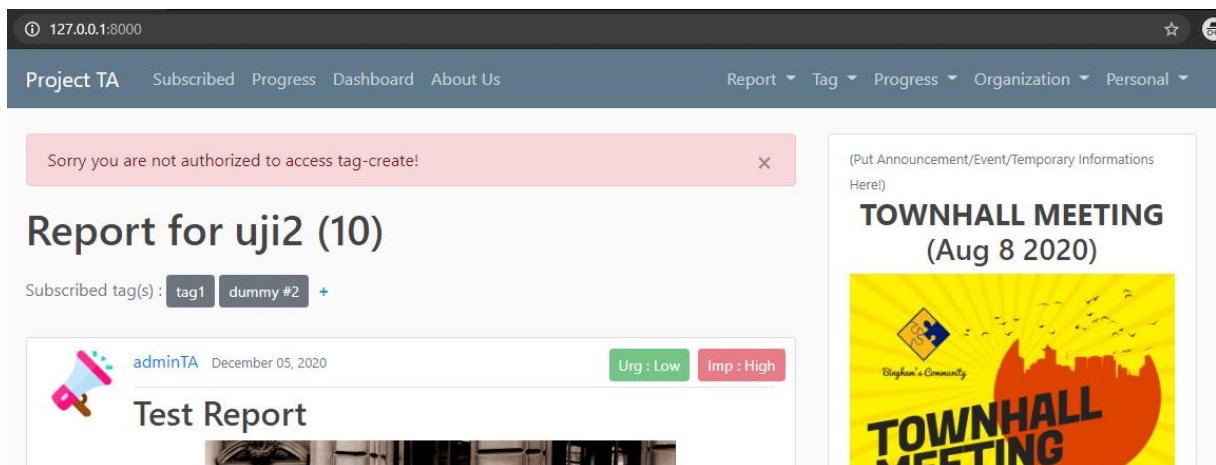
Database Structure Browse Data Edit Pragma Execute SQL

Table: report_tag Filter in any column

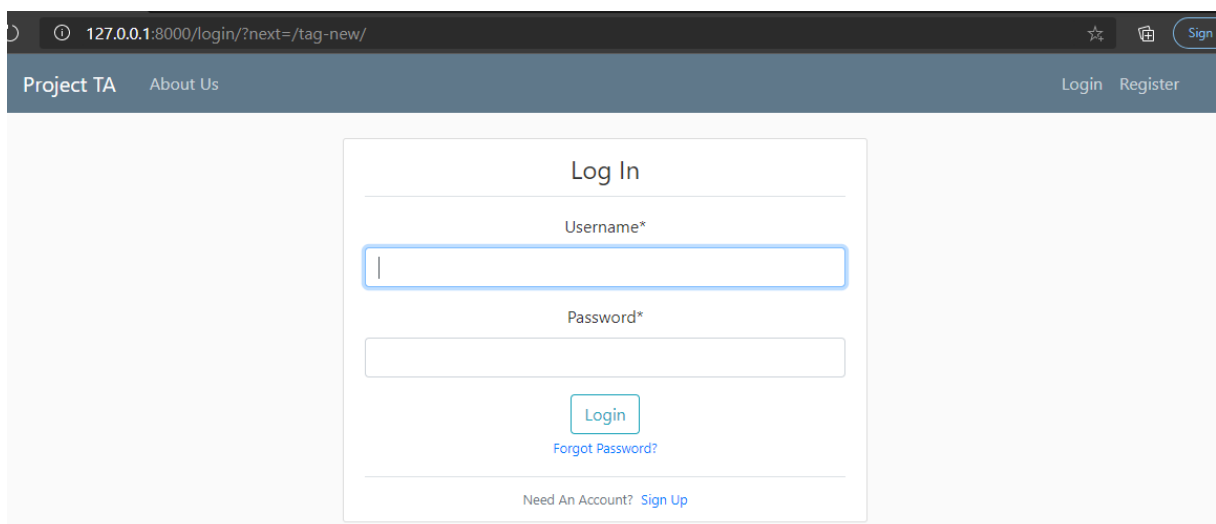
id	description	creator_id	name
1	8 Tag ini akan menandai laporan mengenai ...	8	Pemborosan Bahan Baku
2	7 aa	8	aa
3	6 Lorem Ipsum dolor sit amet, consectetur adipisicing...	8	Lorem Ipsum

Gambar B.35 Cuplikan layar basis data untuk model ‘Tag’ setelah pengujian laman ‘tag-create’

Pada peramban kedua, laman ‘tag-create’ tidak dapat diakses dan dialihkan ke laman ‘home’. Cuplikan pengujian ini dapat terlihat pada gambar B.37



Gambar B.36 Cuplikan layar laman ‘tag-create’ pada pengujian pada peramban web kedua



127.0.0.1:8000/login/?next=/tag-new/ Sign

Project TA About Us Login Register

Log In

Username*

Password*

Login

[Forgot Password?](#)

Need An Account? [Sign Up](#)

Gambar B.37 Cuplikan layar laman ‘tag-create’ pada pengujian pada peramban web ketiga

Pada peramban ketiga, laman '*tag-create*' tidak dapat diakses dan dialihkan ke laman '*login*'. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman '*tag-create*'. Cuplikan pengujian ini dapat terlihat pada gambar B.38.

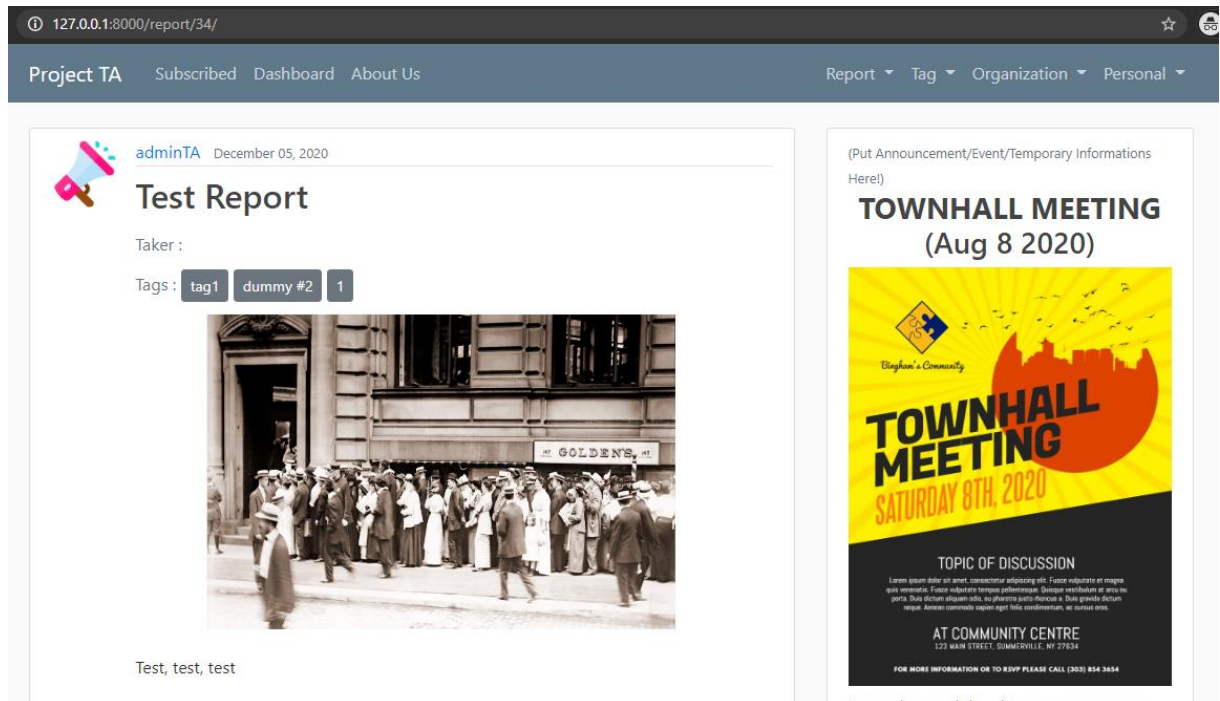
B.4. Pengujian Fitur 4 “Mengambil Laporan” (*Take a Report*)

Pengujian pada fitur 4 “Mengambil Laporan” akan meliputi pengujian laman '*report-detail*', '*progress-taken*', '*progress-subscribed*', dan '*user-taken*'. Walaupun telah diuji di sub sub bab B.1, laman '*report-detail*' akan diuji lagi di sub sub bab ini. Namun elemen interaksi yang diuji pada sub sub bab ini akan berupa elemen yang terkait pengambilan dan penanganan laporan, berbeda dengan elemen yang diuji pada sub sub bab B.1. Seperti pada sub bab sebelumnya, pengujian pada fitur 4 akan terdiri dari tiga jenis pengujian (bila memungkinkan) pada setiap laman seperti yang disebutkan di atas melalui tiga peramban web. Sebelum dilakukan pengujian, otorisasi akun uji2 atas fitur 4 akan dimatikan, sedangkan otoritas akun uji2 atas fitur lain tetap diberikan.

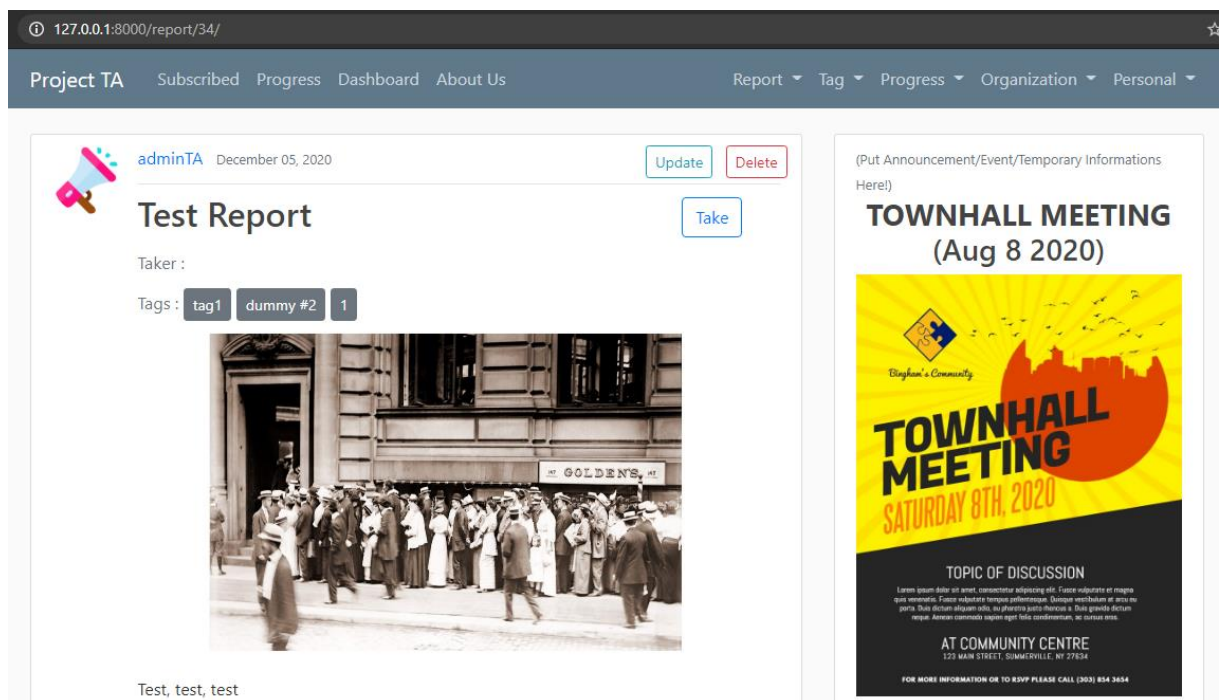
Pada pengujian lanjutan untuk laman '*report-detail*', tidak akan diuji aksesibilitasnya lagi. Namun akun uji2 akan menguji dari perspektif pengguna yang bisa melihat laporan tapi tidak bisa mengambil laporan dan tidak mengambil laporan. Sedangkan akun adminTA akan menjadi pengguna yang bisa mengambil laporan dan nantinya akan mengambil laporan. Dan pada kasus pengguna yang tidak masuk (*login*) dari peramban ketiga tidak akan diuji lagi karena telah dibahas pada sub sub bab B.1. Kedua kasus yang berbeda tersebut akan menghasilkan interaksi elemen yang berbeda. Selain itu juga akan diuji pengunggahan data dari kasus yang dapat mengunggah data.

Pada kasus pengujian ini akan ditinjau kasus pada laporan berjudul “Test Report” dengan kunci primer pk=34. Sebelum pengujian laporan dengan pk=34 ini belum diambil untuk diselesaikan.

Pada peramban pertama, di laman '*report-detail*' milik laporan dengan pk=34, terdapat tombol bertuliskan “Take” yang dapat dipencet untuk mengambil laporan. Hal tersebut dapat dilihat pada gambar B.39. Sedangkan pada peramban kedua, di laman yang sama tidak terdapat tombol tersebut, seperti yang terlihat pada gambar B.40. Hal tersebut sesuai dengan rancangan karena akun uji2 pada peramban kedua tidak memiliki otoritas atas fitur 4.



Gambar B.38 Cuplikan layar laman 'report-detail' pada pengujian kedua pada peramban web pertama



Gambar B.39 Cuplikan layar laman 'report-detail' pada pengujian kedua pada peramban web kedua

Pada peramban pertama kemudian dipencet tombol “Take” untuk mengambil laporan. Laporan berhasil diambil, hal tersebut dapat terlihat bahwa terdapat pengambil laporan berjudul “Test Report” pada basis data untuk objek ‘Report’. Hal tersebut dapat dilihat pada gambar

B.41, berbeda seperti saat laporan berjudul “Test Report” belum diambil seperti pada gambar B.21.

DB Browser for SQLite - C:\Users\giang\PycharmProjects\TA\TA.db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: report_report

id	title	content	date_reported	reporter_id	progress	taker_id	date_last_progress	progress_note	point
1	34 Test Report	Test, test, test	2020-12-04 18:37:29.517443	8	1	8	2020-12-14 06:11:14.675980		NULL
2	33 Test Auth	Test Test	2020-11-24 15:41:57.501451	6	2	8	2020-12-07 03:35:34.634134	Sedang diperiksa	NULL

Gambar B.40 Cuplikan layar basis data untuk model ‘Report’ setelah pengujian ‘report-detail’ kedua

Pada peramban pertama kemudian dilanjutkan pengujian ketiga terhadap laman ‘report-detail’ milik laporan dengan pk=34. Pada kasus ini akun adminTA di peramban pertama menjadi pengambil laporan sehingga dapat mengubah status laporan, menambah catatan pada laporan, serta memberi poin pada pelapor laporan. Terlihat juga pengambil laporan dari laporan terkait, dalam kasus ini adalah akun adminTA. Elemen pada laman ‘report-detail’ pada kasus ini seperti kolom pilihan status laporan dan kolom isian untuk catatan dan poin dapat bekerja dengan baik. Hal tersebut dapat terlihat pada gambar B.42.

127.0.0.1:8000/report/34/

Project TA Subscribed Progress Dashboard About Us Report Tag Progress Organization Personal

adminTA December 05, 2020 Delete

Test Report

Taker : adminTA


Progress : Not Approved 14Dec20

Laporan ini telah dilaporkan sebelumnya (merupakan duplikat)

Point -2 *once filled, point can't be changed


Save Progress

Tags : tag1 dummy #2 1



(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING (Aug 8 2020)

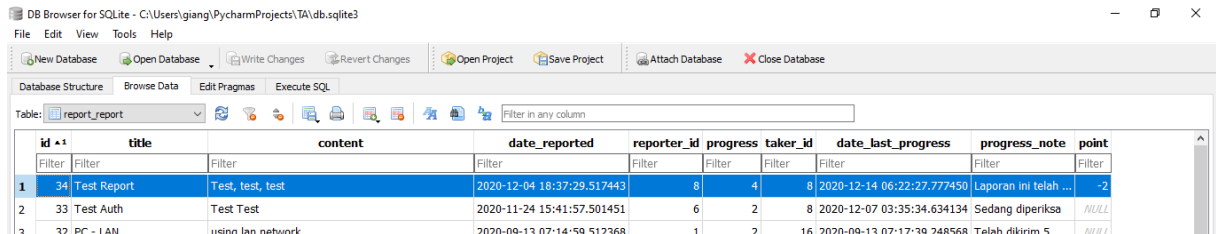


FOR MORE INFORMATION OR TO RSVP PLEASE CALL (302) 854 3454

Lorem ipsum dolor sit amet, consectetur

Gambar B.41 Cuplikan layar laman ‘report-detail’ pada pengujian ketiga pada peramban web pertama

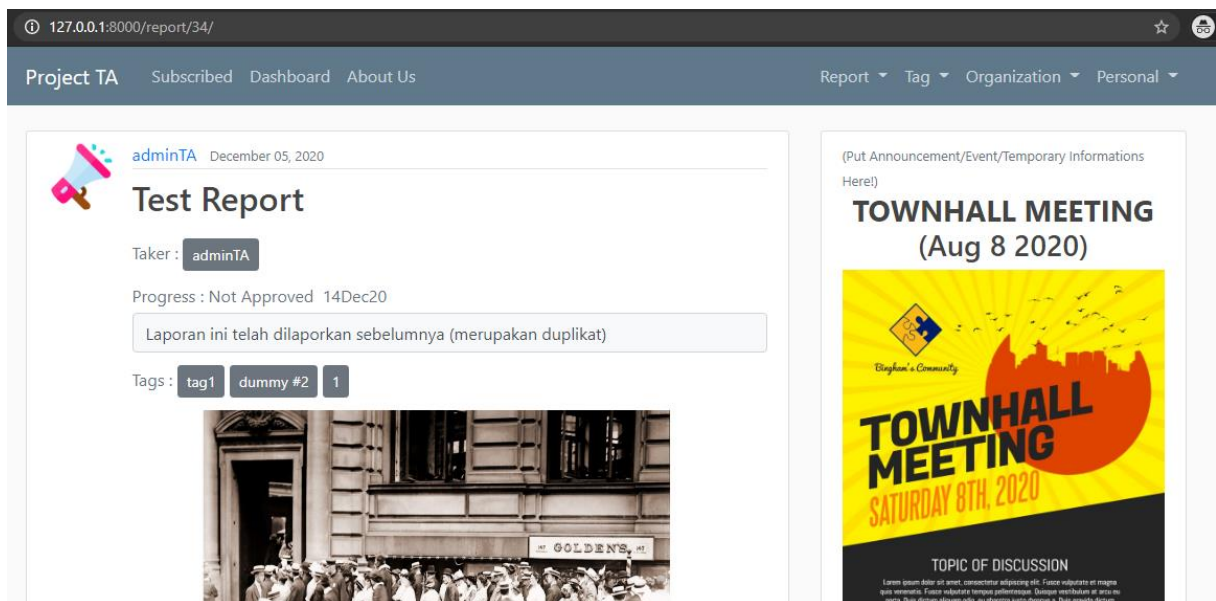
Setelah yakin dengan informasi yang diisi seperti pada gambar B.42. Pengguna memencet tombol bertuliskan “Save Progress” untuk mengunggah data mengenai pengerjaan laporan. Tombol tersebut bekerja dengan baik dan data telah diunggah dan masuk pada basis data untuk objek ‘Report’ seperti yang ditunjukkan pada gambar B.43.



id	title	content	date_reported	reporter_id	progress	taker_id	date_last_progress	progress_note	point
1	34 Test Report	Test, test, test	2020-12-04 18:37:29.517443	8	4	8	2020-12-14 06:22:27.777450	Laporan ini telah ...	-2
2	33 Test Auth	Test Test	2020-11-24 15:41:57.501451	6	2	8	2020-12-07 03:35:34.634134	Sedang diperiksa	NULL
3	32 PC - LAN	usina lan network	2020-09-13 07:14:59.512368	1	2	16	2020-09-13 07:17:39.248568	Telah dikirim 5 ...	NULL

Gambar B.42 Cuplikan layar basis data untuk model ‘Report’ setelah pengujian ‘report-detail’ ketiga

Pada peramban kedua kemudian dilangsungkan pengujian ketiga terhadap laman ‘report-detail’ milik laporan dengan pk=34. Pada kasus ini akun uji2 bukanlah pengambil laporan. Maka dari itu akun uji2 dapat melihat perubahan mengenai pengerjaan laporan, namun tidak dapat merubah. Hal tersebut dapat dilihat pada gambar B.44.



127.0.0.1:8000/report/34/

Project TA Subscribed Dashboard About Us Report Tag Organization Personal

adminTA December 05, 2020

Test Report

Taker : adminTA

Progress : Not Approved 14Dec20

Laporan ini telah dilaporkan sebelumnya (merupakan duplikat)

Tags : tag1 dummy #2 1

(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING (Aug 8 2020)

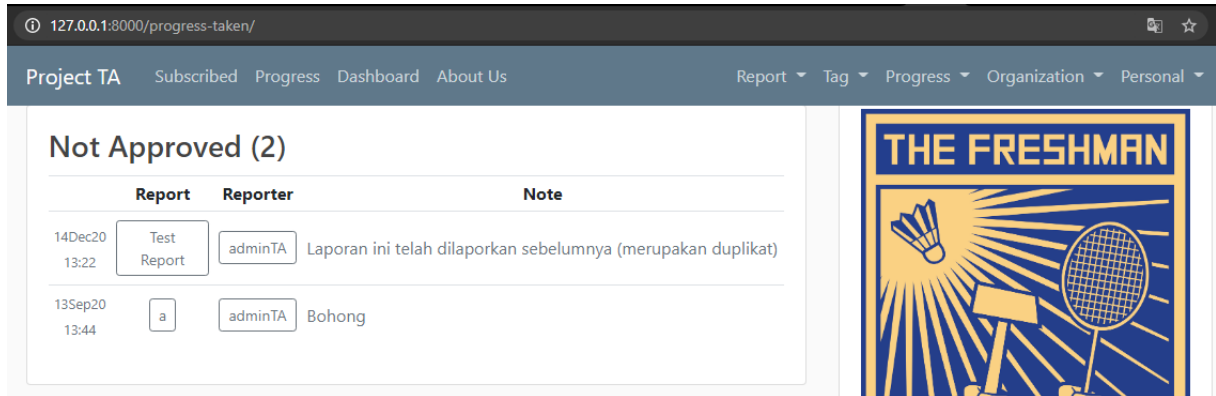
TOWNHALL MEETING SATURDAY 8TH, 2020

TOPIC OF DISCUSSION

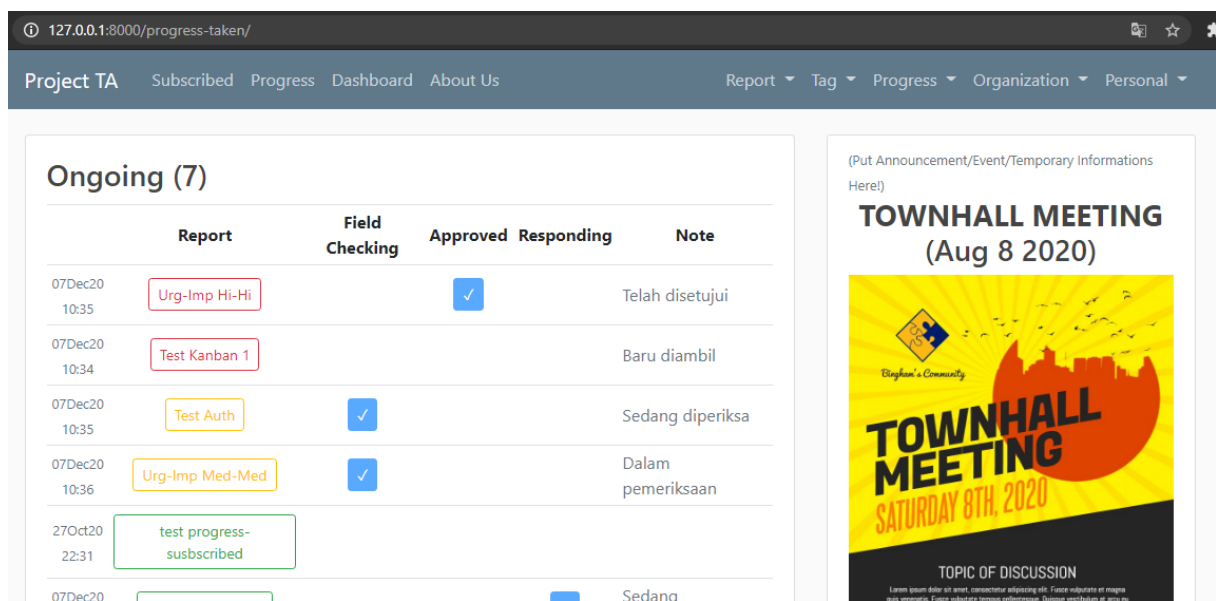
Gambar B.43 Cuplikan layar laman ‘report-detail’ pada pengujian ketiga pada peramban web kedua

Pada laman ‘progress-taken’ akan diuji aksesibilitas dan interaksi elemennya. Interaksi elemen yang diuji adalah munculnya laporan yang diambil beserta statusnya. Sedangkan pengujian pengunggahan data tidak diuji karena tidak ada data yang dapat diunggah.

Pada peramban pertama, laman *'progress-taken'* dapat diakses. Dapat terlihat juga laporan yang telah diambil adminTA seperti pada gambar B.45. Laporan berjudul “Test Report” yang sebelumnya ditolak beserta alasan dan poin yang dikurangi juga dimunculkan pada laman ini bila digulir (*scroll*) kebawah seperti pada gambar B.46.

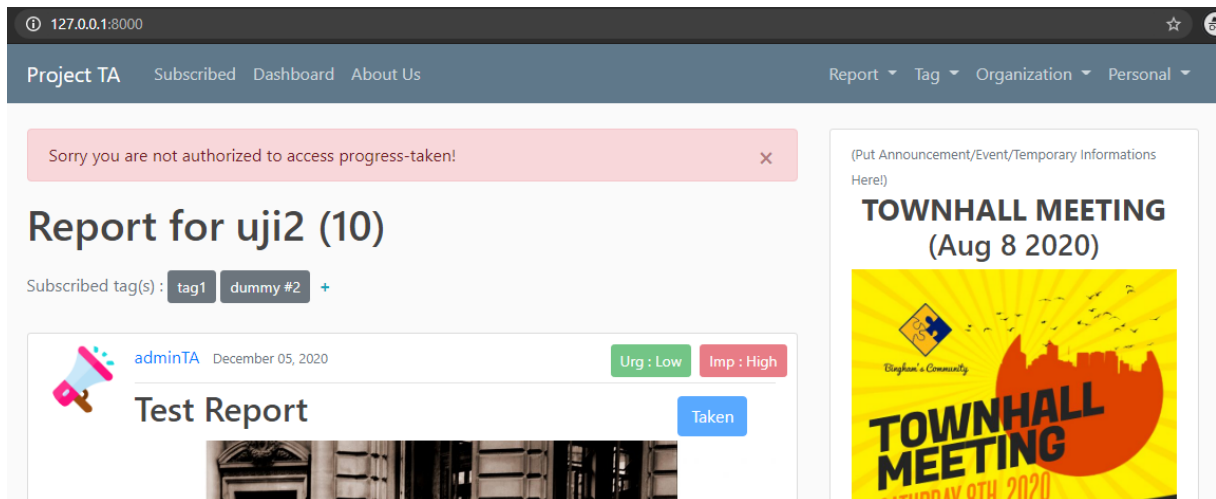


Gambar B.44 Cuplikan layar laman *'progress-taken'* pada pengujian pada peramban web pertama

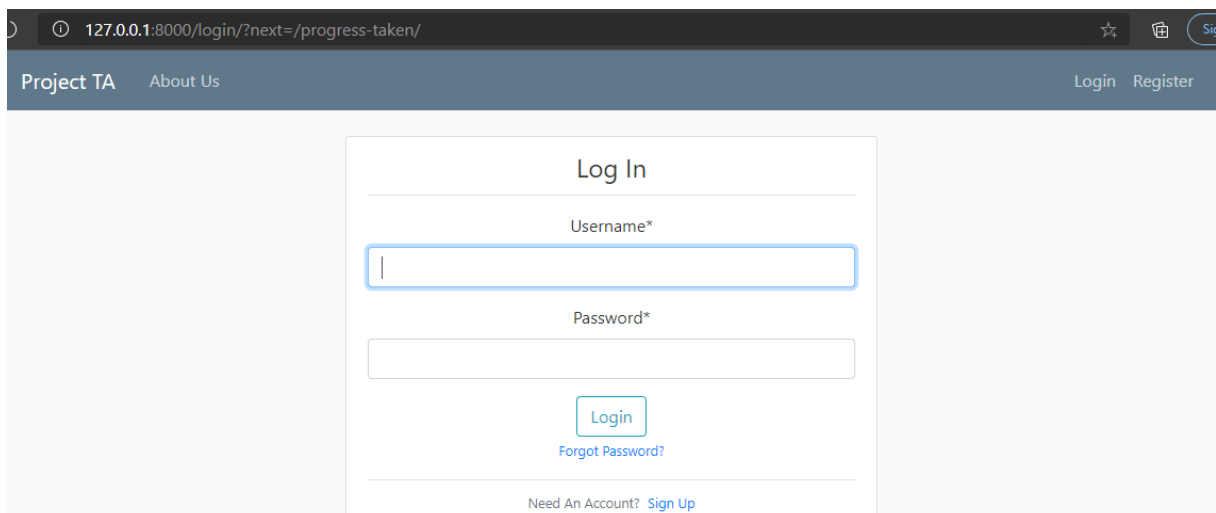


Gambar B.45 Cuplikan layar laman *'progress-taken'* pada pengujian pada peramban web pertama (2)

Pada peramban kedua, laman *'progress-taken'* tidak dapat diakses dan dialihkan ke laman *'home'*. Cuplikan pengujian ini dapat terlihat pada gambar B.47.



Gambar B.46 Cuplikan layar laman *'progress-taken'* pada pengujian pada peramban web kedua



Gambar B.47 Cuplikan layar laman *'progress-taken'* pada pengujian pada peramban web ketiga

Pada peramban ketiga, laman *'progress-taken'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'progress-taken'*. Cuplikan pengujian ini dapat terlihat pada gambar B.48.

Pada laman *'progress-subscribed'* akan diuji aksesibilitas dan interaksi elemennya. Interaksi elemen yang diuji adalah munculnya laporan yang dilanggan beserta statusnya. Sedangkan pengujian pengunggahan data tidak diuji karena tidak ada data yang dapat diunggah.

Pada peramban pertama, laman *'progress-subscribed'* dapat diakses. Dapat terlihat juga laporan yang telah dilanggan adminTA seperti pada gambar B.49.

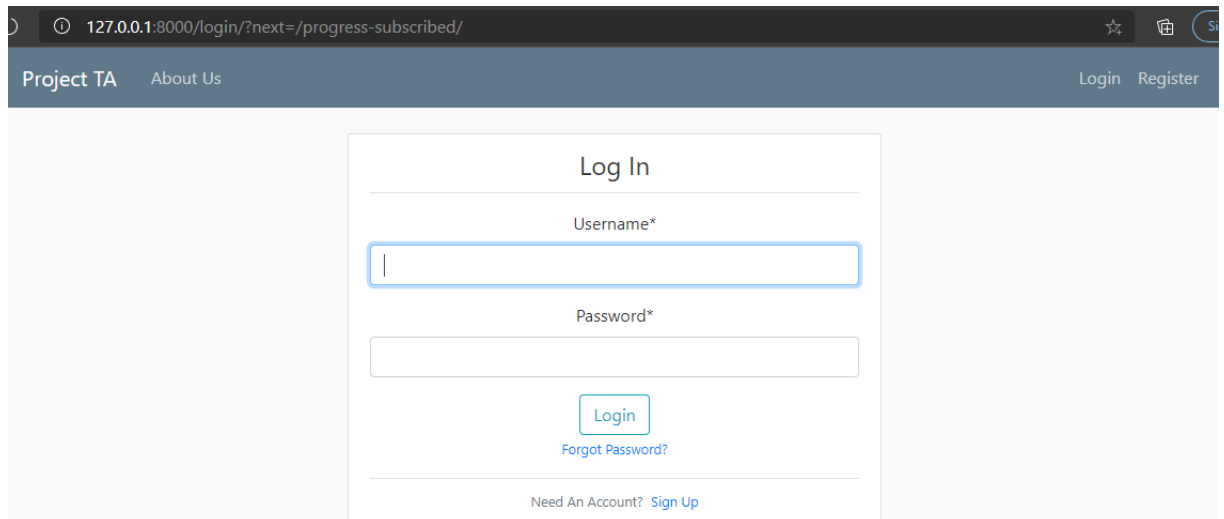
	Taker	Report	Field Checking	Approved	Responding	Note
07Dec20 10:35	adminTA	Urg-Imp Hi-Hi		✓		Telah disetujui
13Sep20	Ujihp2	PC - LAN	✓			Telah dikirim

Gambar B.48 Cuplikan layar laman *'progress-subscribed'* pada pengujian pada peramban web pertama

Pada peramban kedua, laman *'progress-subscribed'* tidak dapat diakses dan dialihkan ke laman *'home'*. Cuplikan pengujian ini dapat terlihat pada gambar B.50.

Gambar B.49 Cuplikan layar laman *'progress-subscribed'* pada pengujian pada peramban web kedua

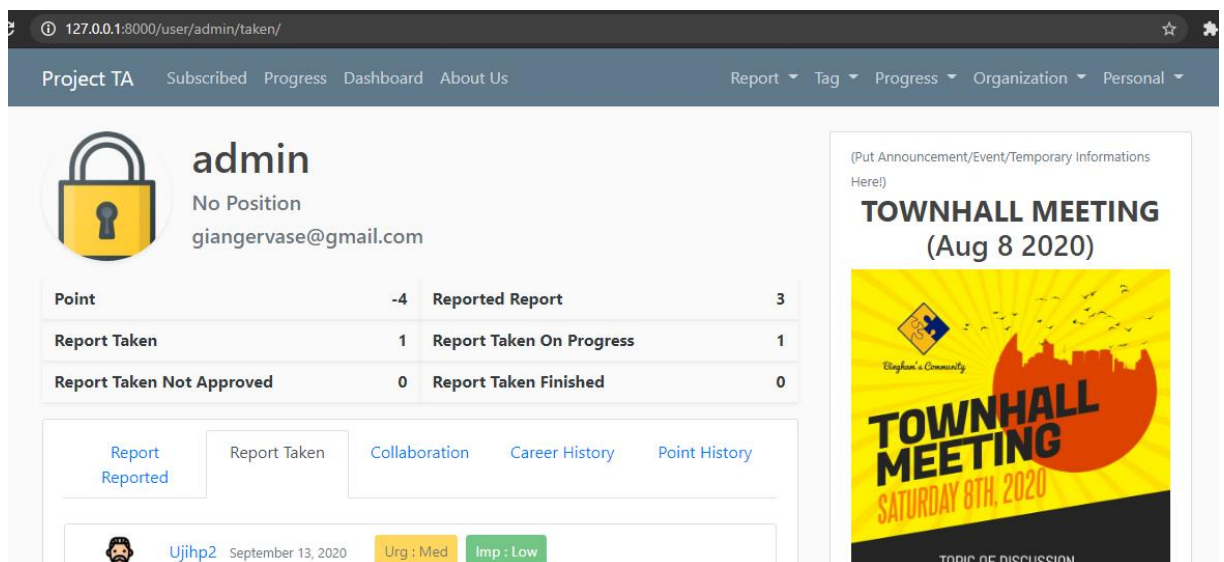
Pada peramban ketiga, laman *'progress-subscribed'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'progress-subscribed'*. Cuplikan pengujian ini dapat terlihat pada gambar B.51.



Gambar B.50 Cuplikan layar laman *'progress-subscribed'* pada pengujian pada peramban web ketiga

Pada laman *'user-taken'* akan diuji aksesibilitas dan interaksi elemennya. Pengunggahan data tidak diuji karena tidak ada data yang dapat diunggah pengguna. Sedangkan elemen yang diuji adalah tombol yang mengarahkan ke laman *'user-reports'*, *'user-collab'*, *'user-career'*, dan *'user-point'*.

Pada kasus pengujian ini akan digunakan laman *'user-taken'* untuk akun bernama pengguna admin. Selain itu, karena laman ini berhubungan dengan fitur 5, maka otoritas akun uji2 atas fitur 5 juga akan dimatikan.



Gambar B.51 Cuplikan layar laman *'user-taken'* pada pengujian pada peramban web pertama

Pada peramban pertama, laman *'user-taken'* dapat diakses. Pada pengujian interaksi elemen terlihat terdapat 4 tombol bertuliskan “Report Reported”, “Collaboration”, “Career

History” dan “Point History” karena akun adminTA memiliki otoritas atas semua fitur, terlihat keempat tombol tersebut. Dan keempat tombol bekerja dengan baik untuk mengalihkan ke laman lain mengenai pengguna terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.52.

Pada peramban kedua, laman ‘user-taken’ tidak dapat diakses dan dialihkan ke laman ‘user-collab’ dari pengguna terkait. Pengalihan berhenti di laman ‘user-collab’ karena akun uji2 memiliki otoritas atas fitur 2. Bila akun uji2 tidak memiliki otoritas atas fitur 2, akun uji2 akan dialihkan lagi ke laman lain. Pada laman ‘user-collab’ dapat terlihat terdapat 3 tombol bertuliskan “Report Reported”, “Career History” dan “Point History”, namun tidak ada tombol bertuliskan “Report Taken”. Ketiadaan tombol “Report Taken” dikarenakan pengguna tidak dapat mengakses laman ‘user-taken’, jadi tidak perlu tampilkan tombol yang dapat mengalihkan pengguna ke laman ‘user-taken’. Cuplikan pengujian ini dapat terlihat pada gambar B.53.

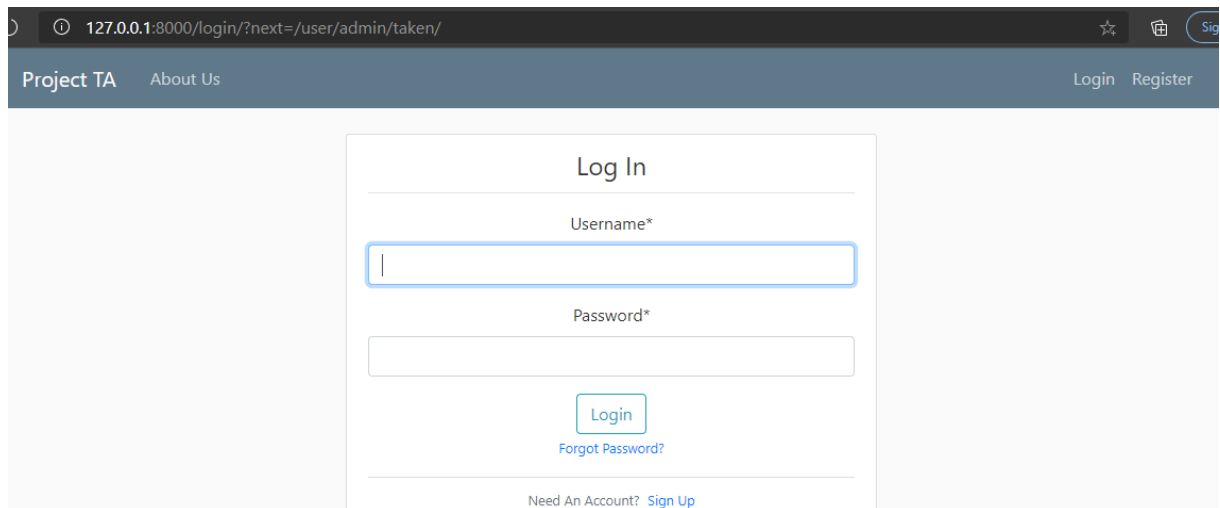
The screenshot shows a web application interface. At the top, there is a navigation bar with links: Project TA, Subscribed, Dashboard, About Us, Report, Tag, Organization, and Personal. Below the navigation bar, the user profile for 'admin' is displayed, showing 'No Position' and the email 'giangervase@gmail.com'. A table of statistics is shown:

Point	-4	Reported Report	3
Report Taken	1	Report Taken On Progress	1
Report Taken Not Approved	0	Report Taken Finished	0

Below the table, there are four tabs: Report Reported, Collaboration, Career History, and Point History. The 'Report Reported' tab is active, showing a post by 'admin' dated December 12, 2020, titled 'Socialism ??' with a subtitle '↳ American dream is better'. The post features an image of the American flag. To the right of the post, there is a 'TOWNHALL MEETING' announcement for August 8, 2020, at the Community Centre. The announcement includes a logo, a sun graphic, and details about the topic of discussion and location.

Gambar B.52 Cuplikan layar laman ‘user-taken’ pada pengujian pada peramban web kedua

Pada peramban ketiga, laman ‘user-taken’ tidak dapat diakses dan dialihkan ke laman ‘login’. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman ‘user-taken’ dari pengguna terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.54.



Gambar B.53 Cuplikan layar laman '*user-taken*' pada pengujian pada peramban web ketiga

B.5. Pengujian Fitur 5 “Melihat Aktivitas Pengguna” (*View User Activities*)

Pengujian pada fitur 5 “Melihat Aktivitas Pengguna” akan meliputi pengujian laman '*user-career*' dan '*user-point*'. Sedangkan laman '*user-reports*', '*user-taken*', dan '*user-collab*' yang merupakan bagian dari fitur 2 dan fitur 4 tidak akan dibahas lagi karena telah diuji pada pengujian fitur 2 dan fitur 4 yang dijabarkan pada sub sub bab B.2 dan B.4. Seperti pada sub bab sebelumnya pengujian pada fitur 5 akan terdiri dari tiga jenis pengujian (bila memungkinkan) pada setiap laman seperti yang disebutkan di atas melalui tiga peramban web. Sebelum dilakukan pengujian, otorisasi akun uji2 atas fitur 5 akan dimatikan, sedangkan otoritas akun uji2 atas fitur lain tetap diberikan.

Pada laman '*user-career*' akan diuji aksesibilitas dan interaksi elemennya. Pengunggahan data tidak diuji karena tidak ada data yang dapat diunggah pengguna. Sedangkan elemen yang diuji adalah tombol yang mengarahkan ke laman '*user-reports*', '*user-taken*', '*user-collab*', dan '*user-point*'.

Pada kasus pengujian ini akan digunakan laman '*user-career*' untuk akun bernama pengguna admin. Selain itu, karena laman ini berhubungan dengan fitur 6, maka otoritas akun uji2 atas fitur 6 juga akan dimatikan.

Pada peramban pertama, laman '*user-career*' dapat diakses. Pada pengujian interaksi elemen terlihat terdapat 4 tombol bertuliskan “Report Reported”, “Report Taken”, “Collaboration” dan “Point History” karena akun adminTA memiliki otoritas atas semua fitur, terlihat keempat tombol tersebut. Dan keempat tombol bekerja dengan baik untuk mengalihkan

ke laman lain mengenai pengguna terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.55.

127.0.0.1:8000/user/admin/career/

Project TA Subscribed Progress Dashboard About Us Report Tag Progress Organization Personal

admin
No Position
giangervase@gmail.com

Point	-4	Reported Report	3
Report Taken	1	Report Taken On Progress	1
Report Taken Not Approved	0	Report Taken Finished	0

Report Reported Report Taken Collaboration Career History Point History

Reg-ID	Date Started	Date Ended	Unit	Job
33	Sep 18, 2020	Sep 18, 2020	Division A1	Head

(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING
(Aug 8 2020)

TOWNHALL MEETING
SATURDAY 8TH, 2020

TOPIC OF DISCUSSION

Gambar B.54 Cuplikan layar laman ‘user-career’ pada pengujian pada peramban web pertama

127.0.0.1:8000/user/admin/point/

Project TA Subscribed Progress Dashboard About Us Report Tag Progress Organization Personal

admin
No Position
giangervase@gmail.com

Point	-4	Reported Report	3
Report Taken	1	Report Taken On Progress	1
Report Taken Not Approved	0	Report Taken Finished	0

Report Reported Report Taken Collaboration Point History

Reg-ID	Date	Point	Writer	Note
2	Oct 17, 2020	-4	adminTA	Tukar dengan gaji

(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING
(Aug 8 2020)

TOWNHALL MEETING
SATURDAY 8TH, 2020

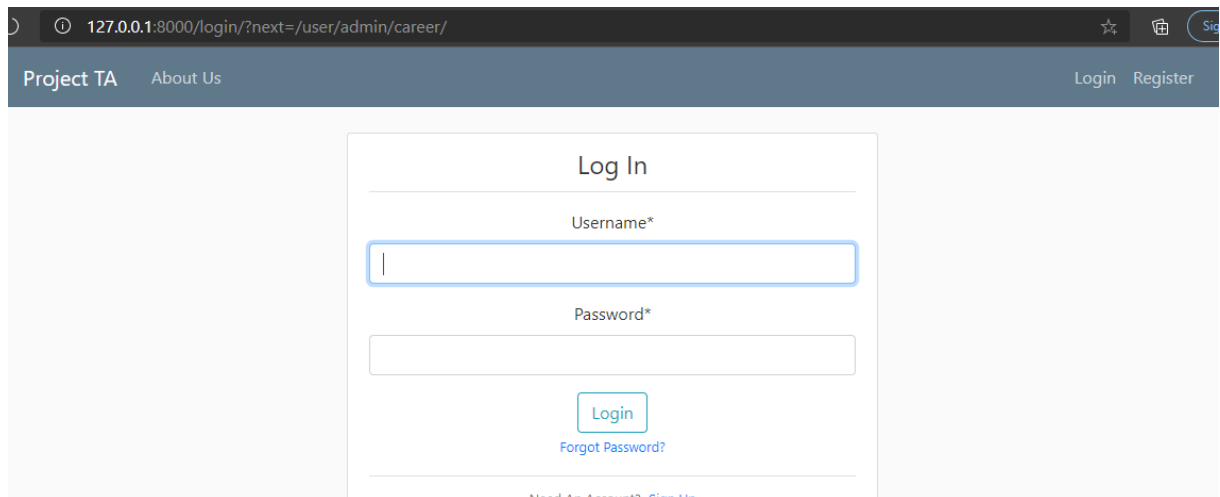
TOPIC OF DISCUSSION

Gambar B.55 Cuplikan layar laman ‘user-career’ pada pengujian pada peramban web kedua

Pada peramban kedua, laman ‘user-career’ tidak dapat diakses dan dialihkan ke laman ‘user-point’ dari pengguna terkait. Pengalihan berhenti di laman ‘user-point’ karena akun uji2 memiliki otoritas atas fitur 8. Bila akun uji2 tidak memiliki otoritas atas fitur 8, akun uji2 akan dialihkan lagi ke laman ‘home’. Pada laman ‘user-point’ dapat terlihat terdapat 3 tombol bertuliskan “Report Reported”, “Report Taken” dan “Collaboration”, namun tidak ada tombol

bertuliskan “Report Career”. Ketiadaan tombol “Report Career” dikarenakan pengguna tidak dapat mengakses laman ‘*user-career*’, jadi tidak perlu tampilkan tombol yang dapat mengalihkan pengguna ke laman ‘*user-career*’. Cuplikan pengujian ini dapat terlihat pada gambar B.56.

Pada peramban ketiga, laman ‘*user-career*’ tidak dapat diakses dan dialihkan ke laman ‘*login*’. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman ‘*user-career*’ dari pengguna terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.57.



Gambar B.56 Cuplikan layar laman ‘*user-career*’ pada pengujian pada peramban web ketiga

Pada laman ‘*user-point*’ akan diuji aksesibilitas dan interaksi elemennya. Pengunggahan data tidak diuji karena tidak ada data yang dapat diunggah pengguna. Sedangkan elemen yang diuji adalah tombol yang mengarahkan ke laman ‘*user-reports*’, ‘*user-taken*’, ‘*user-collab*’, dan ‘*user-career*’.

Pada kasus pengujian ini akan digunakan laman ‘*user-point*’ untuk akun bernama pengguna admin. Selain itu, karena laman ini berhubungan dengan fitur 8, maka otoritas akun uji2 atas fitur 8 juga akan dimatikan. Dan juga otoritas uji2 atas fitur 6 dikembalikan lagi karena tidak berhubungan dengan laman ini.

Pada peramban pertama, laman ‘*user-point*’ dapat diakses. Pada pengujian interaksi elemen terlihat terdapat 4 tombol bertuliskan “Report Reported”, “Report Taken”, “Collaboration” dan “Career History” karena akun adminTA memiliki otoritas atas semua fitur, terlihat keempat tombol tersebut. Dan keempat tombol bekerja dengan baik untuk mengalihkan

ke laman lain mengenai pengguna terkait. Cuplikan pengujian ini dapat terlihat pada gambar B.58.

The screenshot shows a web application interface. At the top, there's a navigation bar with 'Project TA' and several dropdown menus. Below this, the user profile for 'admin' is displayed, including a lock icon, the name 'admin', 'No Position', and the email 'giangervase@gmail.com'. A table shows various report statistics:

Point	-4	Reported Report	3
Report Taken	1	Report Taken On Progress	1
Report Taken Not Approved	0	Report Taken Finished	0

Below the table, there are tabs for 'Report Reported', 'Report Taken', 'Collaboration', 'Career History', and 'Point History'. The 'Report Taken' tab is active, showing a table with columns: Reg-ID, Date, Point, Writer, and Note. The data row shows: 2, Oct 17, 2020, -4, adminTA, and Tukar dengan gaji. On the right side, there's a sidebar with a 'TOWNHALL MEETING (Aug 8 2020)' announcement.

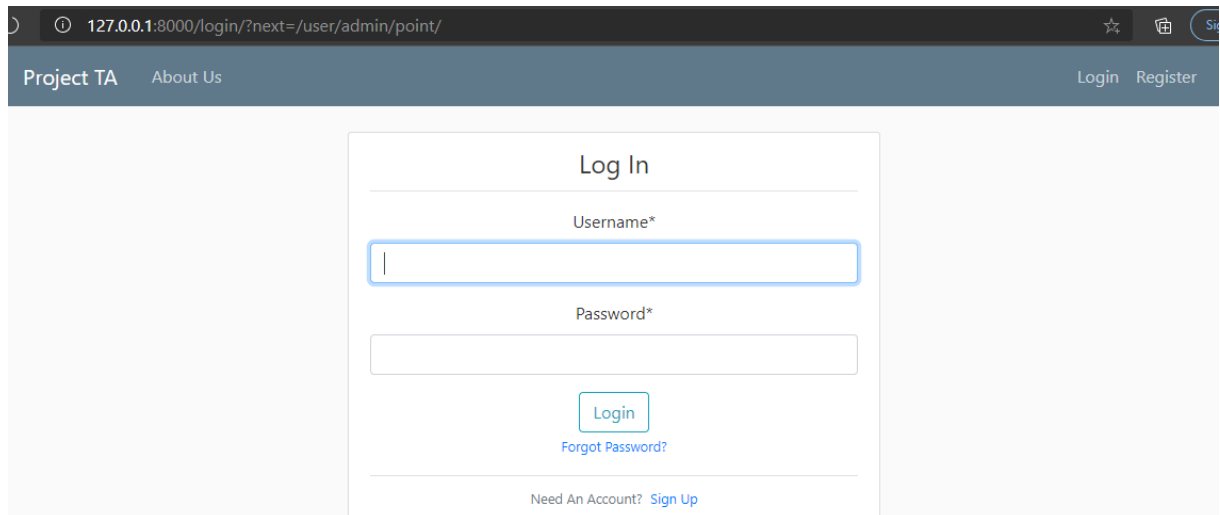
Gambar B.57 Cuplikan layar laman 'user-point' pada pengujian pada peramban web pertama

Pada peramban kedua, laman 'user-points' tidak dapat diakses dan dialihkan ke laman 'home'. Cuplikan pengujian ini dapat terlihat pada gambar B.59.

The screenshot shows a web application interface. At the top, there's a navigation bar with 'Project TA' and several dropdown menus. Below this, a red error message is displayed: 'Sorry you are not authorized to access user-profile!'. The main content area shows a 'Report for uji2 (10)' with subscribed tags 'tag1' and 'dummy #2'. Below this, there's a 'Test Report' section with a megaphone icon, the name 'adminTA', the date 'December 05, 2020', and buttons for 'Urg : Low', 'Imp : High', and 'Taken'. A small image of a building is shown below the 'Test Report' section. On the right side, there's a sidebar with a 'TOWNHALL MEETING (Aug 8 2020)' announcement.

Gambar B.58 Cuplikan layar laman 'user-point' pada pengujian pada peramban web kedua

Pada peramban ketiga, laman 'user-point' tidak dapat diakses dan dialihkan ke laman 'login'. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman 'user-point'. Cuplikan pengujian ini dapat terlihat pada gambar B.60.



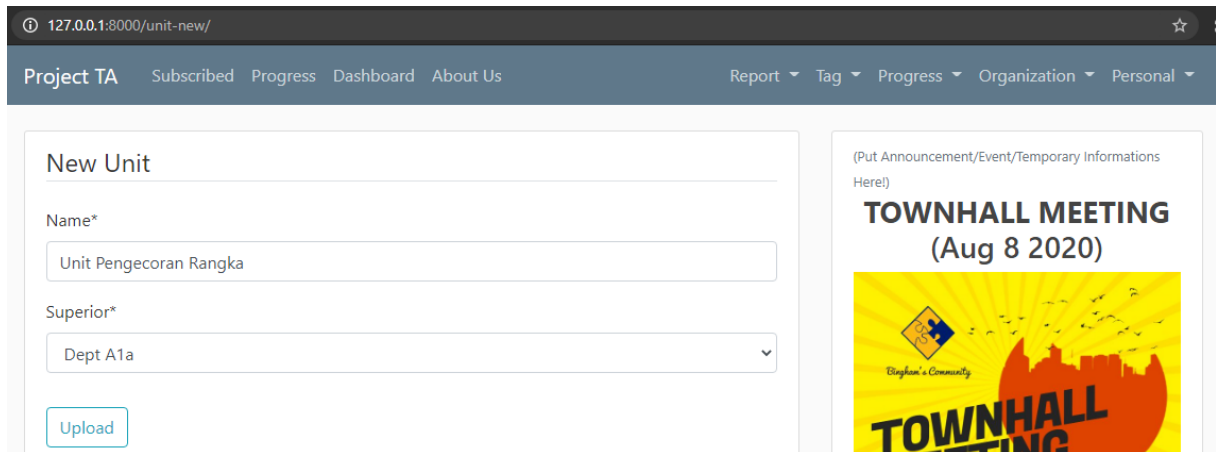
Gambar B.59 Cuplikan layar laman ‘user-point’ pada pengujian pada peramban web ketiga

B.6. Pengujian Fitur 6 “Mengelola Organisasi” (*Manage Organization*)

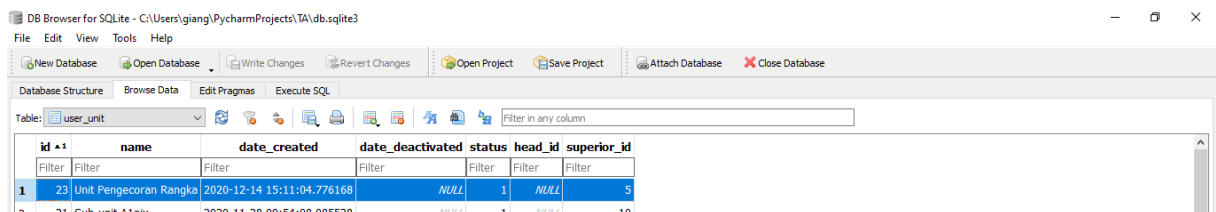
Pengujian pada fitur 6 “Mengelola Organisasi” akan meliputi pengujian laman ‘unit-create’, ‘unit-detail’, ‘unit-list’, dan ‘user-list’. Sedangkan laman ‘user-career’ yang merupakan bagian dari fitur 5 tidak akan dibahas lagi karena telah diuji pada pengujian fitur 5 yang dijabarkan pada sub sub bab B.5. Seperti pada sub bab sebelumnya pengujian pada fitur 6 akan terdiri dari tiga jenis pengujian (bila memungkinkan) pada setiap laman seperti yang disebutkan di atas melalui tiga peramban web. Sebelum dilakukan pengujian, otorisasi akun uji2 atas fitur 6 akan dimatikan, sedangkan otorisasi akun uji2 atas fitur lain tetap diberikan.

Pada laman ‘unit-create’ akan diuji aksesibilitasnya, interaksi elemennya, dan pengunggahan datanya. Interaksi yang diuji dengan mengisi kolom isian dan kolom pilihan untuk membuat unit baru. Sedangkan data yang diunggah akan berupa data yang diisi yang akan membuat objek ‘Unit’ yang baru.

Pada peramban pertama, laman ‘unit-create’ dapat diakses. Pada pengujian ini dicoba untuk berinteraksi dengan elemen dengan mengisi kolom isian dan kolom pilihan. Kedua kolom pilihan dan kolom isian dapat bekerja dengan baik. Cuplikan pengujian interaksi dapat terlihat pada gambar B.61. Setelah mengisi informasi yang akan diunggah, pengguna memencet tombol bertuliskan “Upload” untuk membuat unit baru. Tombol bekerja dengan baik dan objek ‘Unit’ baru telah tercatat di basis data. Cuplikan basis data pada gambar B.62 menyoroti data yang berhasil diunggah.

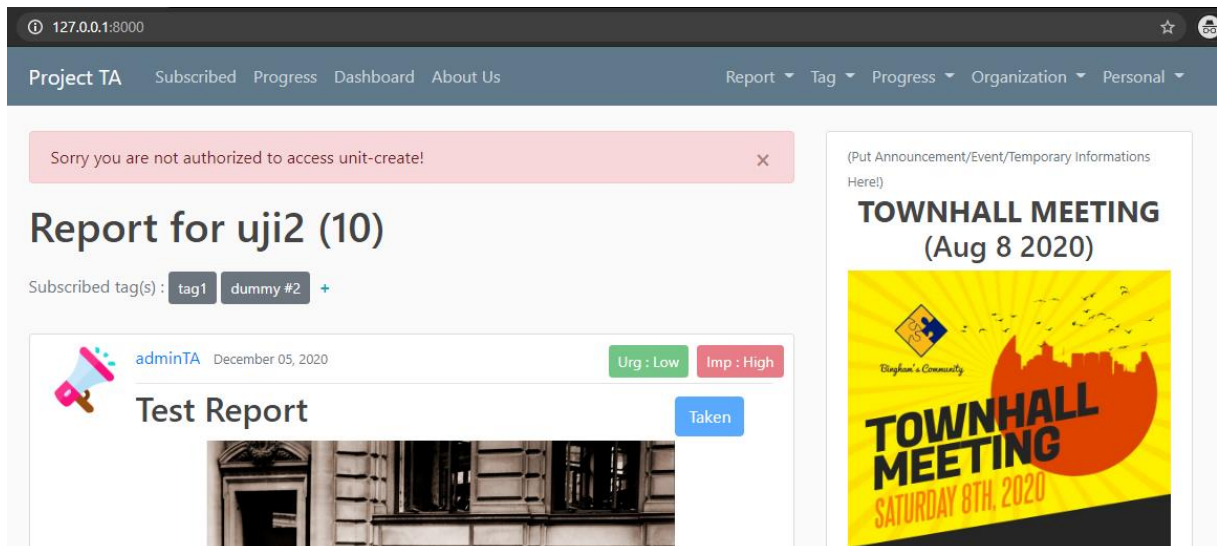


Gambar B.60 Cuplikan layar laman 'unit-create' pada pengujian pada peramban web pertama



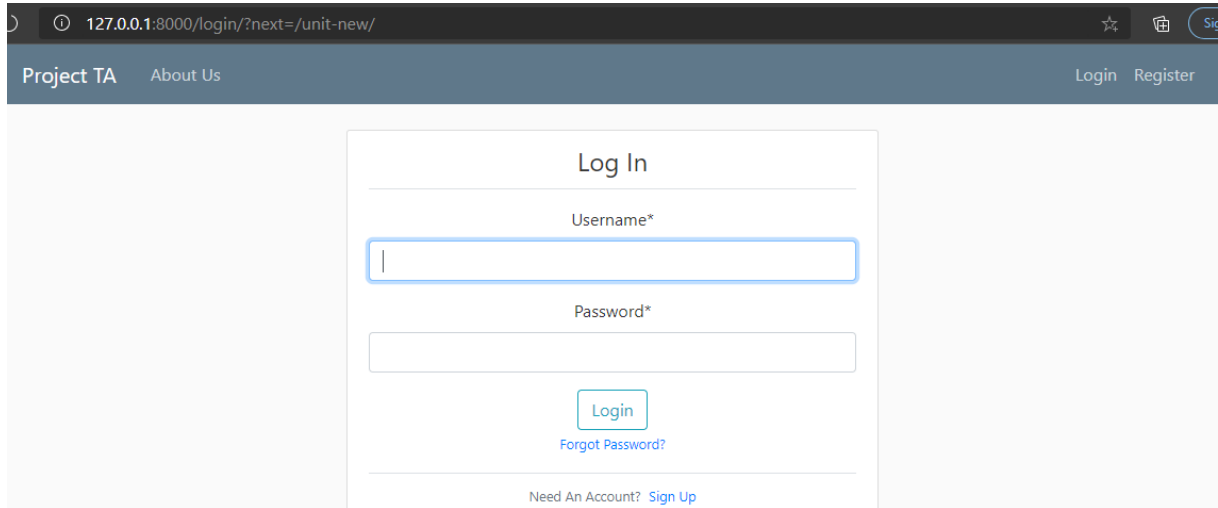
Gambar B.61 Cuplikan layar basis data untuk model 'Unit' setelah pengujian laman 'unit-create'

Pada peramban kedua, laman 'unit-create' tidak dapat diakses dan dialihkan ke laman 'home'. Cuplikan pengujian ini dapat terlihat pada gambar B.63.



Gambar B.62 Cuplikan layar laman 'unit-create' pada pengujian pada peramban web kedua

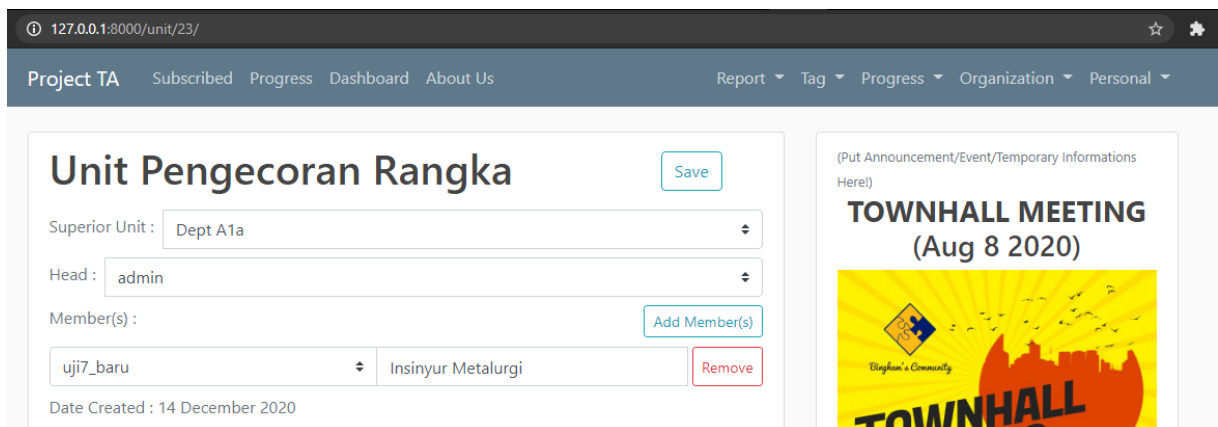
Pada peramban ketiga, laman *'unit-create'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'unit-create'*. Cuplikan pengujian ini dapat terlihat pada gambar B.64



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/login/?next=/unit-new/'. The page has a dark blue header with 'Project TA' and 'About Us' on the left, and 'Login' and 'Register' on the right. The main content area is white and contains a 'Log In' form. The form has two input fields: 'Username*' and 'Password*'. Below the password field is a 'Login' button. There is a link 'Forgot Password?' below the login button. At the bottom of the form, there is a link 'Need An Account? Sign Up'.

Gambar B.63 Cuplikan layar laman *'unit-create'* pada pengujian pada peramban web ketiga

Pada laman *'unit-detail'* akan diuji aksesibilitas, pengujian interaksinya, dan pengunggahan datanya. Pada pengujian interaksi elemennya, akan diuji tombol bertuliskan “Add Member(s)” dan kolom isian. Sedangkan data yang diunggah adalah data yang diisi pada kolom isian.

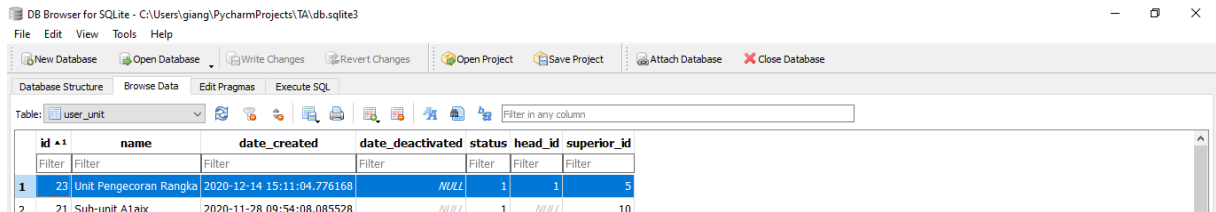


The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/unit/23/'. The page has a dark blue header with 'Project TA' and 'Subscribed Progress Dashboard About Us' on the left, and 'Report Tag Progress Organization Personal' on the right. The main content area is white and contains a form titled 'Unit Pengecoran Rangka'. The form has a 'Save' button. It includes a dropdown for 'Superior Unit' (Dept A1a), a dropdown for 'Head' (admin), and a section for 'Member(s)'. The 'Member(s)' section has a table with one row containing 'uji7_baru' and 'Insinyur Metalurgi', with 'Add Member(s)' and 'Remove' buttons. Below the table, it says 'Date Created : 14 December 2020'. On the right side of the page, there is a sidebar with a yellow background and a red sun-like graphic, containing the text 'TOWNHALL MEETING (Aug 8 2020)' and 'Diagon's Community'.

Gambar B.64 Cuplikan layar laman *'unit-detail'* pada pengujian pada peramban web pertama

Pada peramban pertama, laman *'unit-detail'* dapat diakses. Pada pengujian ini dicoba untuk berinteraksi dengan tombol bertuliskan “Add Member(s)”. Tombol bekerja dengan baik, dan muncul satu baris kolom isian untuk satu kali memencet tombol tersebut. Pada baris yang muncul tersebut terdapat tombol bertuliskan “Remove”. Tombol Remove tersebut juga bekerja dengan baik untuk menghilangkan baris terkait. Setiap kolom isian dan kolom pilihan baik yang

telah ada ataupun yang muncul dari pencetan tombol “Add Member(s)” dapat bekerja dengan baik. Cuplikan pengujian interaksi dapat terlihat pada gambar B.65. Setelah dapat mengisi informasi yang akan diunggah, pengguna memencet tombol bertuliskan “Save” untuk mengubah data pada suatu unit. Tombol bekerja dengan baik dan data terkait unit tersebut telah tercatat di basis data. Cuplikan basis data pada gambar B.66 dan B.67 menyoroti data yang berhasil diunggah.



DB Browser for SQLite - C:\Users\giang\PycharmProjects\TA\db.sqlite3

File Edit View Tools Help

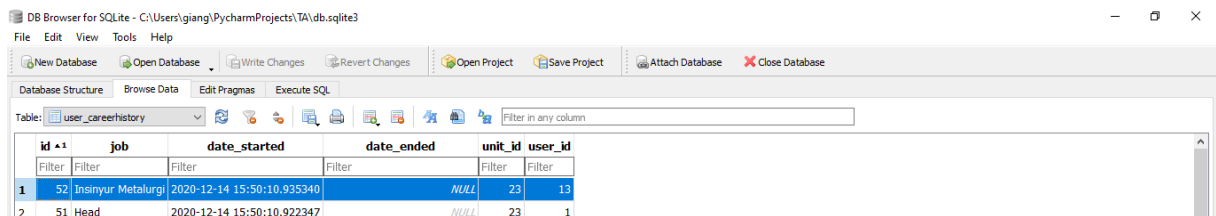
New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: user_unit

	id	name	date_created	date_deactivated	status	head_id	superior_id
1	23	Unit Pengecoran Rangka	2020-12-14 15:11:04.776168	NULL	1	1	5
2	21	Sub-unit A1aix	2020-11-28 09:54:08.085528	NULL	1	NULL	10

Gambar B.65 Cuplikan layar basis data untuk model ‘Unit’ setelah pengujian laman ‘unit-detail’



DB Browser for SQLite - C:\Users\giang\PycharmProjects\TA\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

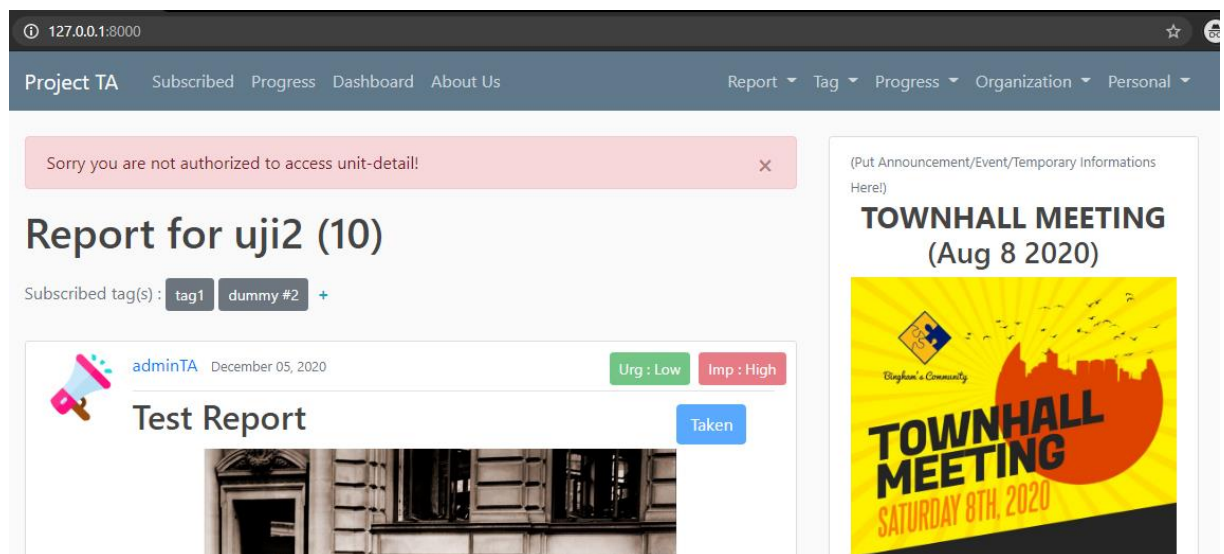
Database Structure Browse Data Edit Pragma Execute SQL

Table: user_careerhistory

	id	job	date_started	date_ended	unit_id	user_id
1	52	Insinyur Metalurgi	2020-12-14 15:50:10.935340	NULL	23	13
2	51	Head	2020-12-14 15:50:10.922347	NULL	23	1

Gambar B.66 Cuplikan layar basis data untuk model ‘CareerHistory’ setelah pengujian laman ‘unit-detail’

Pada peramban kedua, laman ‘unit-detail’ tidak dapat diakses dan dialihkan ke laman ‘home’. Cuplikan pengujian ini dapat terlihat pada gambar B.68.



127.0.0.1:8000

Project TA Subscribed Progress Dashboard About Us Report Tag Progress Organization Personal

Sorry you are not authorized to access unit-detail!

Report for uji2 (10)

Subscribed tag(s): tag1 dummy #2 +

adminTA December 05, 2020 Urg : Low Imp : High Taken

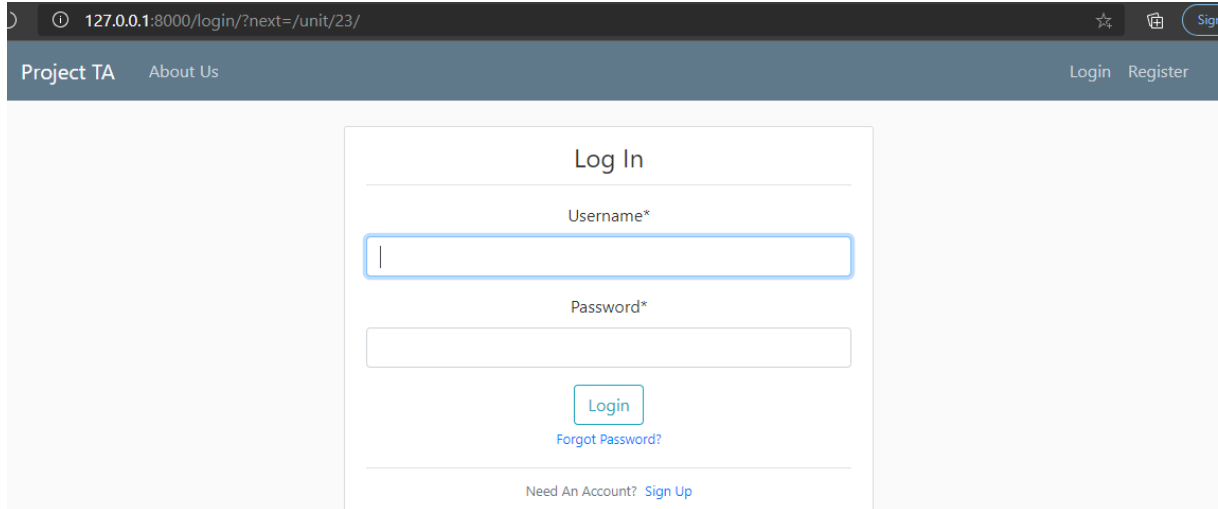
Test Report

TOWNHALL MEETING (Aug 8 2020)

TOWNHALL MEETING SATURDAY 8TH, 2020

Gambar B.67 Cuplikan layar laman ‘unit-detail’ pada pengujian pada peramban web kedua

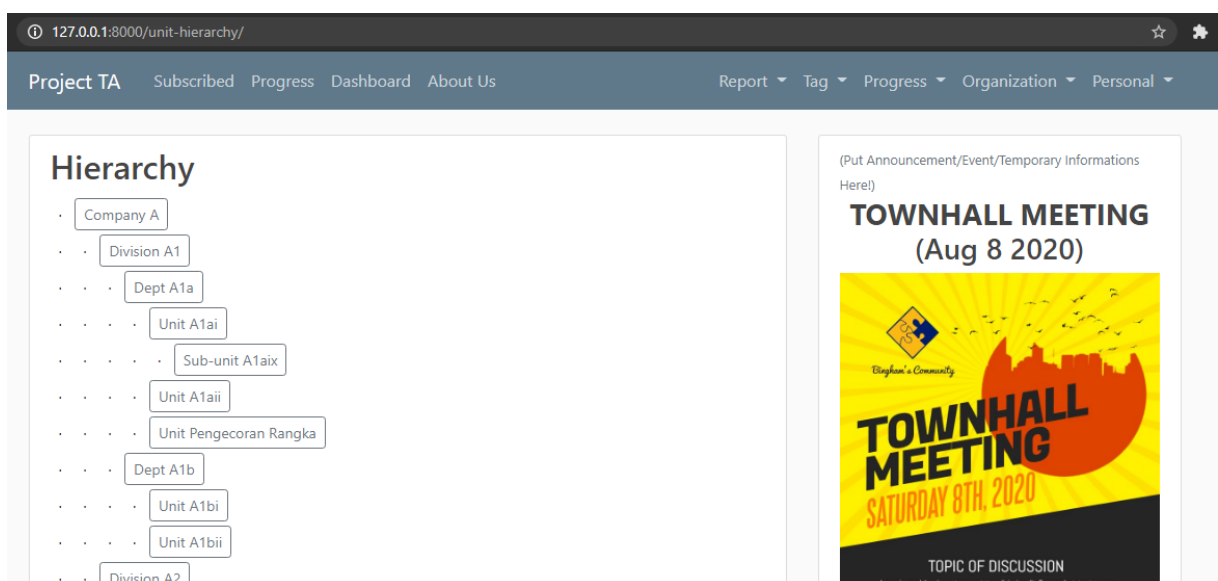
Pada peramban ketiga, laman *'unit-detail'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'unit-detail'*. Cuplikan pengujian ini dapat terlihat pada gambar B.69.



Gambar B.68 Cuplikan layar laman *'unit-detail'* pada pengujian pada peramban web ketiga

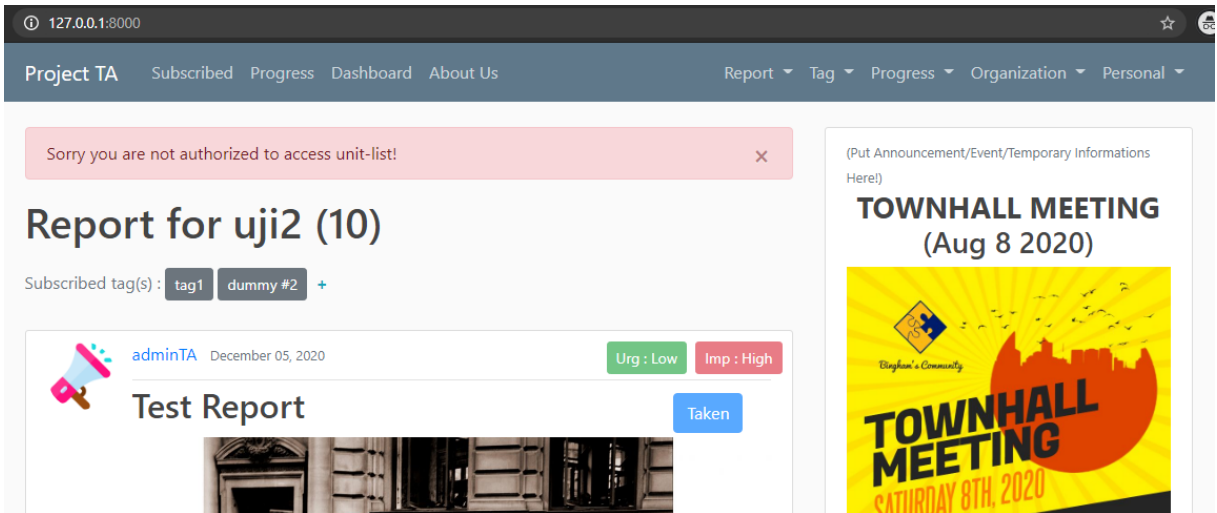
Pada laman *'unit-list'* akan diuji aksesibilitas dan interaksi elemennya. Pengunggahan data tidak diuji karena tidak ada data yang dapat diunggah. Untuk elemen yang diuji adalah tombol yang bertuliskan nama unit. Semua unit harus dimunculkan dalam bentuk tombol dan semua tombol harus dapat mengalihkan pengguna ke laman *'unit-detail'* dari unit tersebut.

Pada peramban pertama, laman *'unit-list'* dapat diakses. Pada pengujian ini dicoba untuk memencet tombol bertuliskan nama unit. Dan semua tombol tersebut dapat mengalihkan pengguna ke laman yang tepat. Cuplikan pengujian interaksi dapat terlihat pada gambar B.70.



Gambar B.69 Cuplikan layar laman *'unit-list'* pada pengujian pada peramban web pertama

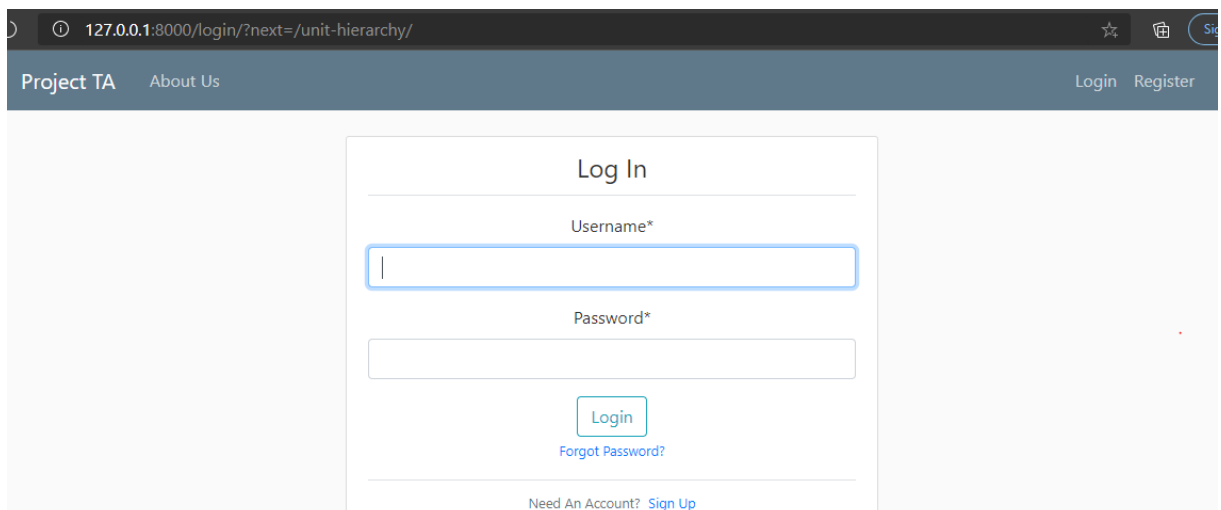
Pada peramban kedua, laman *'unit-list'* tidak dapat diakses dan dialihkan ke laman *'home'*. Cuplikan pengujian ini dapat terlihat pada gambar B.71.



Gambar B.70 Cuplikan layar laman *'unit-list'* pada pengujian pada peramban web kedua

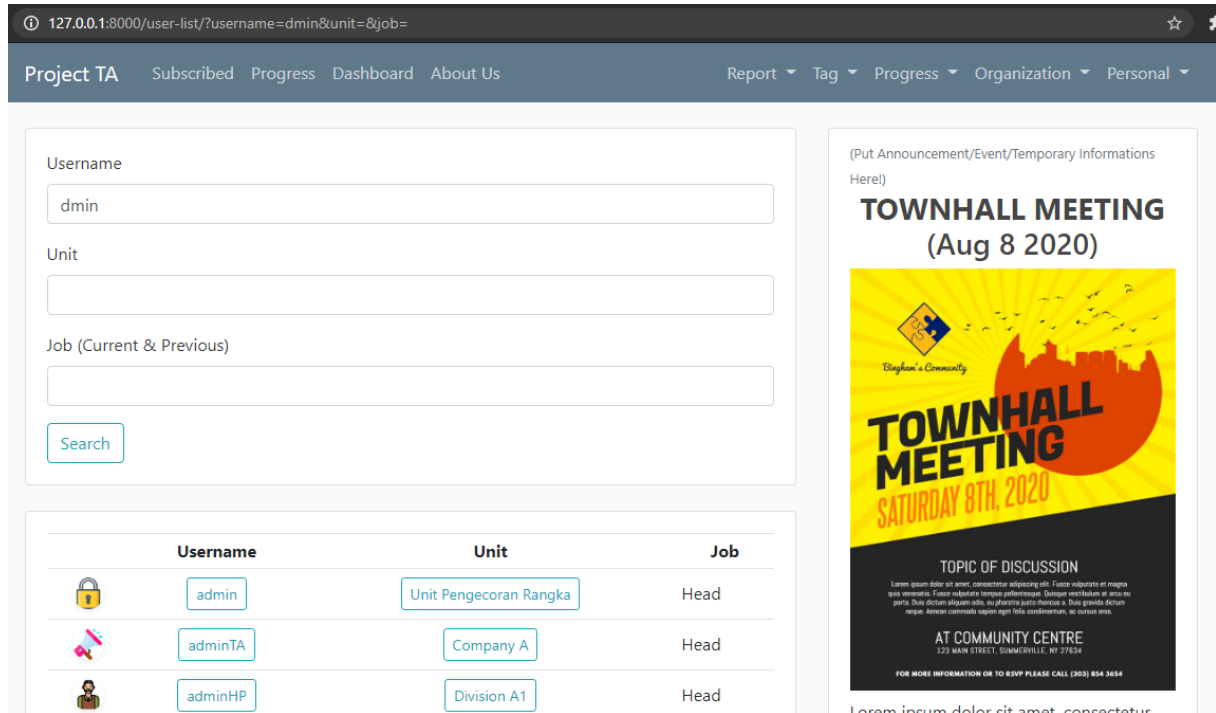
Pada peramban ketiga, laman *'unit-list'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'unit-list'*. Cuplikan pengujian ini dapat terlihat pada gambar B.72.

Pada laman *'user-list'* akan diuji aksesibilitas dan interaksi elemennya. Pengunggahan data tidak diuji karena tidak ada data yang dapat diunggah ke basis data. Untuk elemen yang diuji adalah filter untuk user beserta kolom isian dan tombol terkait.



Gambar B.71 Cuplikan layar laman *'unit-list'* pada pengujian pada peramban web ketiga

Pada peramban pertama, laman ‘*user-list*’ dapat diakses. Semua elemen yang terkait dengan filter seperti kolom isian dan tombol bertuliskan “Search” dapat bekerja dengan baik. Filter juga bekerja dengan baik. Pada kasus ini pengguna mencari pengguna dengan nama pengguna yang mengandung karakter “dmin”, dan hanya muncul pengguna dengan nama pengguna yang mengandung karakter “dmin” seperti pada gambar B.73.



127.0.0.1:8000/user-list/?username=dmin&unit=&job=

Project TA Subscribed Progress Dashboard About Us Report Tag Progress Organization Personal

Username
dmin

Unit

Job (Current & Previous)

Search

	Username	Unit	Job
🔒	admin	Unit Pengecoran Rangka	Head
🔑	adminTA	Company A	Head
👤	adminHP	Division A1	Head

(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING
(Aug 8 2020)

TOWNHALL MEETING
SATURDAY 8TH, 2020

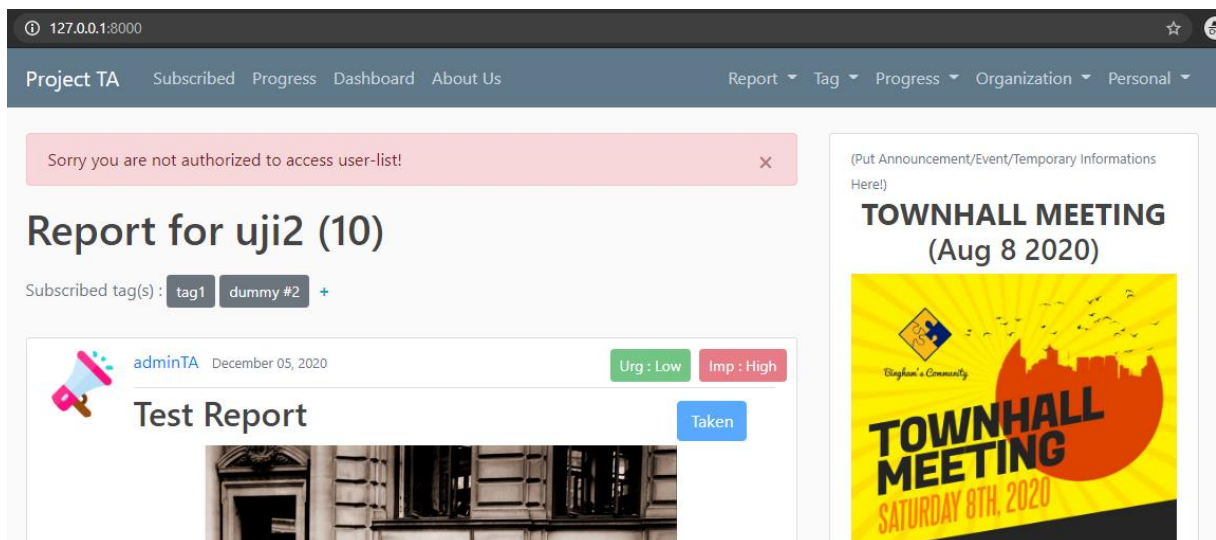
TOPIC OF DISCUSSION

AT COMMUNITY CENTRE
123 MAIN STREET, SUMMERSVILLE, NY 27834

FOR MORE INFORMATION OR TO RSVP PLEASE CALL (303) 834 3434

Gambar B.72 Cuplikan layar laman ‘*user-list*’ pada pengujian pada peramban web pertama

Pada peramban kedua, laman ‘*user-list*’ tidak dapat diakses dan dialihkan ke laman ‘*home*’. Cuplikan pengujian ini dapat terlihat pada gambar B.74.



127.0.0.1:8000

Project TA Subscribed Progress Dashboard About Us Report Tag Progress Organization Personal

Sorry you are not authorized to access user-list!

Report for uji2 (10)

Subscribed tag(s): tag1 dummy #2 +

adminTA December 05, 2020 Urg : Low Imp : High

Test Report Taken

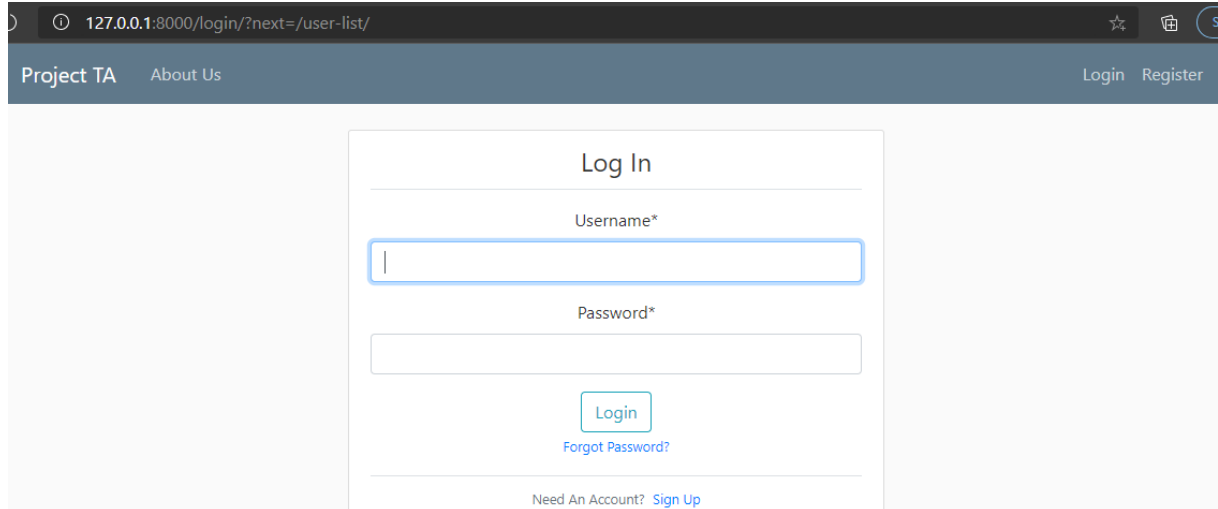
(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING
(Aug 8 2020)

TOWNHALL MEETING
SATURDAY 8TH, 2020

Gambar B.73 Cuplikan layar laman ‘*user-list*’ pada pengujian pada peramban web kedua

Pada peramban ketiga, laman *'user-list'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'user-list'*. Cuplikan pengujian ini dapat terlihat pada gambar B.75.



Gambar B.74 Cuplikan layar laman *'user-list'* pada pengujian pada peramban web ketiga

B.7. Pengujian Fitur 7 “Mengelola Otorisasi” (*Manage Authorization*)

Pengujian pada fitur 7 “Mengelola Otorisasi” akan berupa pengujian laman *'auth-detail'*. Seperti sub bab sebelumnya, pengujian pada fitur 7 akan terdiri dari tiga jenis pengujian (bila memungkinkan) pada laman *'auth-detail'* melalui tiga peramban web. Sebelum dilakukan pengujian, otoritas akun uji2 atas fitur 7 akan dimatikan, sedangkan otoritas akun uji2 atas fitur lainnya akan tetap diberikan.

Pada laman *'auth-detail'* akan diuji aksesibilitasnya, interaksi elemennya, dan pengunggahan datanya. Pada pengujian interaksi elemen, akan diuji tombol bertuliskan “Add User(s)”, tombol bertuliskan “Add Unit(s)”, dan tombol bertuliskan “Add Level(s)” beserta kolom pilihan yang muncul dari pemencetan tombol-tombol tersebut. Sedangkan data yang diunggah adalah terkait pengguna (*user*), pengguna (*unit*), tingkatan (*level*), serta fitur terkait yang telah disetel pada laman tersebut.

Pada peramban pertama, laman *'auth-detail'* dapat diakses. Pada pengujian ini dicoba untuk berinteraksi dengan tombol bertuliskan “Add User(s)”, tombol bertuliskan “Add Unit(s)”, dan tombol bertuliskan “Add Level(s)” dengan cara memencet tombol-tombol tersebut. Tombol-tombol tersebut bekerja dengan baik dengan memunculkan satu baris isian yang berisi kolom pilihan dan tombol bertuliskan “Remove” untuk setiap satu kali pencet dari tombol-tombol tersebut. Tombol bertuliskan “Remove” juga dapat menghapus baris isian

dengan tepat. Cuplikan pengujian interaksi dapat terlihat pada gambar B.76. Setelah dapat menyetel otorisasi yang akan diunggah, pengguna memencet tombol bertuliskan “Save” untuk mengubah otorisasi yang tersimpan pada basis data. Tombol bekerja dengan baik dan data terkait otorisasi tersebut telah tercatat di basis data. Cuplikan basis data pada gambar B.77 menyoroti data yang berhasil diunggah.

The screenshot displays a web application interface for managing authorization. The top navigation bar includes links for Project TA, Subscribed, Progress, Dashboard, and About Us, along with dropdown menus for Report, Tag, Progress, Organization, and Personal. The main content area is titled 'Manage Authorization' and features a 'Save' button. Below this, there are three sections: 'Create A Report', 'View Reports', and 'Create A Tag'. Each section allows users to select users and units for authorization. The 'Create A Tag' section also includes a 'Level' dropdown menu. On the right side of the interface, there are two event announcements: 'TOWNHALL MEETING (Aug 8 2020)' and 'INTERNAL BADMINTON SPARRING (Aug 17 2020)'. The first announcement includes a poster for the town hall meeting and a paragraph of placeholder text. The second announcement includes a poster for the badminton sparring event.

Manage Authorization [Save]

Create A Report

User [Add User(s)]

adminTA	[Remove]
uji2	[Remove]
admin	[Remove]

Unit [Add Unit(s)]

Level [Add Level(s)]

View Reports

User [Add User(s)]

adminTA	[Remove]
uji2	[Remove]

Unit [Add Unit(s)]

Division A1	[Remove]
-------------	----------

Level [Add Level(s)]

Create A Tag

User [Add User(s)]

adminTA	[Remove]
uji2	[Remove]

Unit [Add Unit(s)]

Level [Add Level(s)]

3 [Remove]

Take A Report

(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING (Aug 8 2020)

TOPIC OF DISCUSSION

AT COMMUNITY CENTRE

FOR MORE INFORMATION OR TO RSVP PLEASE CALL (303) 854 3434

INTERNAL BADMINTON SPARRING (Aug 17 2020)

THE FRESHMAN

Gambar B.75 Cuplikan layar laman 'auth-detail' pada pengujian pada peramban web pertama

DB Browser for SQLite - C:\Users\giang\PYcharmProjects\TA\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

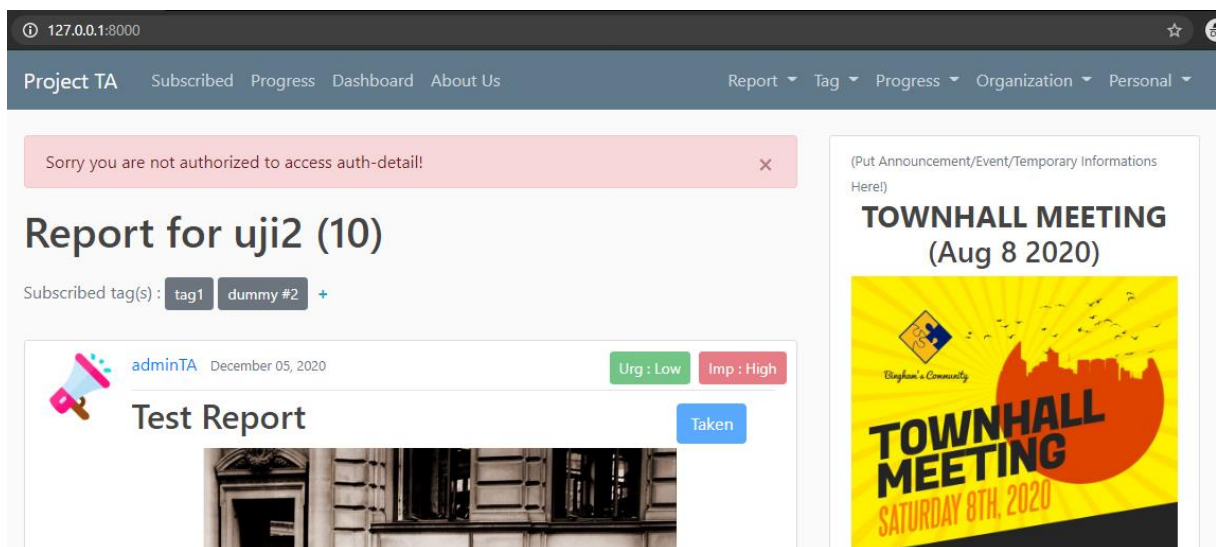
Table: user_auth

	id	auth_level	auth_unit_id	auth_user_id	feature
	Filter	Filter	Filter	Filter	Filter
1	112	3	NULL	NULL	3
2	111	NULL	3	NULL	2
3	110	NULL	NULL	1	1
4	109	NULL	NULL	6	9

Gambar B.76 Cuplikan layar basis data untuk model 'Auth' setelah pengujian laman 'auth-detail'

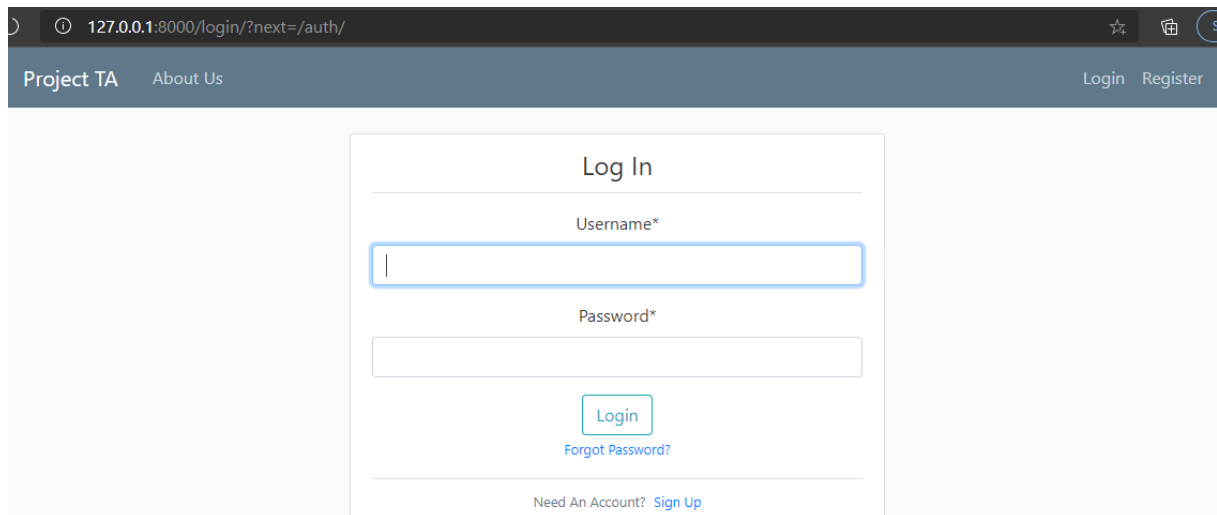
Mengenai kinerja dari fitur 7 perihal membatasi aksesibilitas dan membatasi interaksi tertentu juga telah teruji di sub bab sebelumnya dan sub sub bab setelah ini. Dimana bila dalam kasus otorisasi diberikan, pengguna terkait dapat mengakses laman atau berinteraksi dengan elemen pada laman. Sedangkan bila dalam kasus otorisasi dimatikan, pengguna terkait tidak dapat mengakses laman atau berinteraksi dengan elemen pada laman.

Pada peramban kedua, laman 'auth-detail' tidak dapat diakses dan dialihkan ke laman 'home'. Cuplikan pengujian ini dapat terlihat pada gambar B.78.



Gambar B.77 Cuplikan layar laman 'auth-detail' pada pengujian pada peramban web kedua

Pada peramban ketiga, laman 'auth-detail' tidak dapat diakses dan dialihkan ke laman 'login'. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman 'auth-detail'. Cuplikan pengujian ini dapat terlihat pada gambar B.79.



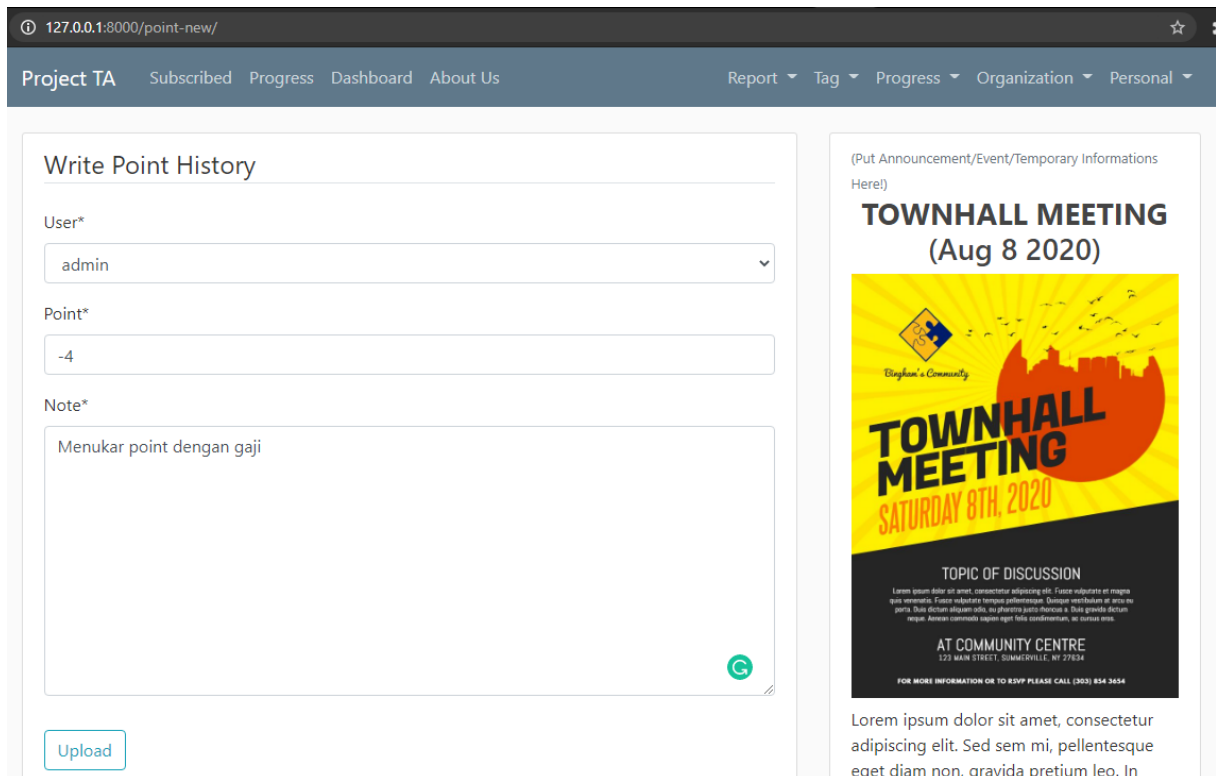
Gambar B.78 Cuplikan layar laman '*auth-detail*' pada pengujian pada peramban web ketiga

B.8. Pengujian Fitur 8 “Mengelola Point” (*Manage Point*)

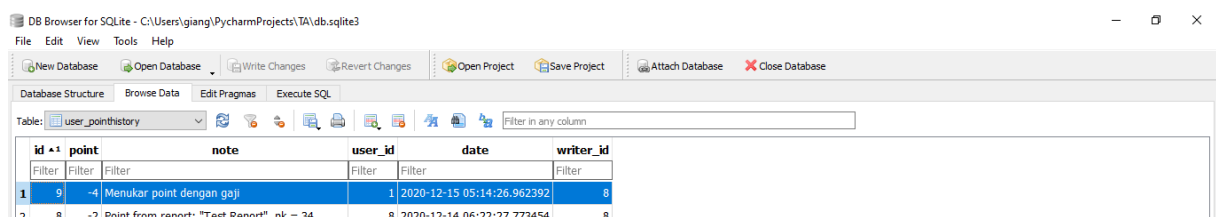
Pengujian pada fitur 8 “Mengelola Poin” akan meliputi pengujian laman '*point-create*' dan '*point-list*'. Sedangkan laman '*user-point*' yang merupakan bagian dari fitur 8 tidak akan dibahas lagi karena telah diuji pada pengujian fitur 5 yang dijabarkan pada sub sub bab B.5. Seperti pada sub bab sebelumnya, pengujian pada fitur 8 akan terdiri dari tiga jenis pengujian (bila memungkinkan) pada setiap laman seperti yang disebutkan di atas melalui tiga peramban web. Sebelum dilakukan pengujian, otorisasi akun uji2 atas fitur 8 akan dimatikan, sedangkan otoritas akun uji2 atas fitur lain tetap diberikan.

Pada laman '*point-create*' akan diuji aksesibilitasnya, interaksi elemennya, dan pengunggahan datanya. Interaksi yang diuji dengan mengisi kolom isian dan kolom pilihan untuk membuat log poin baru. Sedangkan data yang diunggah akan berupa data yang diisi yang akan membuat objek '*PointHistory*' yang baru.

Pada peramban pertama, laman '*point-create*' dapat diakses. Pada pengujian ini dicoba untuk berinteraksi dengan elemen dengan mengisi kolom isian dan kolom pilihan. Kedua kolom pilihan dan kolom isian dapat bekerja dengan baik. Cuplikan pengujian interaksi dapat terlihat pada gambar B.80. Setelah mengisi kolom isian dan pilihan tersebut, pengguna memencet tombol bertuliskan “Upload” untuk membuat log poin baru. Tombol bekerja dengan baik dan objek '*PointHistory*' baru telah tercatat di basis data. Cuplikan basis data pada gambar B.81 menyoroti data yang berhasil diunggah.

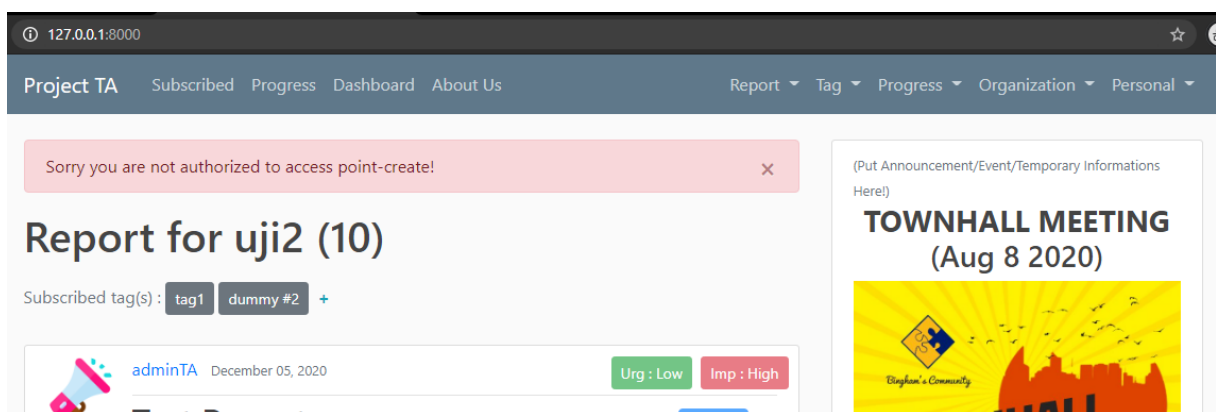


Gambar B.79 Cuplikan layar laman 'point-create' pada pengujian pada peramban web pertama



Gambar B.80 Cuplikan layar basis data untuk model 'PointHistory' setelah pengujian laman 'point-create'

Pada peramban kedua, laman 'point-create' tidak dapat diakses dan dialihkan ke laman 'home'. Cuplikan pengujian ini dapat terlihat pada gambar B.82.



Gambar B.81 Cuplikan layar laman 'point-create' pada pengujian pada peramban web kedua

Pada peramban ketiga, laman *'point-create'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'point-create'*. Cuplikan pengujian ini dapat terlihat pada gambar B.83.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/login/?next=/point-new/`. The page has a dark blue header with 'Project TA' and 'About Us' on the left, and 'Login' and 'Register' on the right. The main content area is white and contains a 'Log In' form. The form has two input fields: 'Username*' and 'Password*'. Below the password field is a 'Login' button and a 'Forgot Password?' link.

Gambar B.82 Cuplikan layar laman *'point-create'* pada pengujian pada peramban web ketiga

Pada laman *'point-list'* akan diuji aksesibilitas dan interaksi elemennya. Pengunggahan data tidak diuji karena tidak ada data yang dapat diunggah ke basis data. Untuk elemen yang diuji adalah filter untuk user beserta kolom isian dan tombol terkait.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/point-list/?date_after=2020-11-01&date_before=&point_gte=-3&point_lte=-2&receiver=&writer=¬e=`. The page has a dark blue header with 'Project TA' and navigation links: 'Subscribed', 'Progress', 'Dashboard', 'About Us', 'Report', 'Tag', 'Progress', 'Organization', and 'Personal'. The main content area is white and contains a search filter section, a table of logs, and a sidebar with a town hall meeting announcement.

Search Filter:

- Log(s) After: 01/11/2020
- Log(s) Before: dd/mm/yyyy
- Point ≥: -3
- Point ≤: -2
- Receiver's Username:
- Writer's Username:
- Note:
- Search button

Summary:

- Number of Log(s): 3
- Sum of Point(s): -8

Table of Logs:

Date	Point	Receiver	Writer	Note
14Dec20 13:22	-2	adminTA	adminTA	Point from report: "Test Report", pk = 34
07Dec20 22:12	-3	adminTA	adminTA	Ditukar dengan bonus bulan Desember
12Nov20 09:35	-3	uji3	adminTA	Ditukar dengan gaji

Sidebar:

(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING (Aug 8 2020)

Topic of Discussion: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce volutpat et magna...

AT COMMUNITY CENTRE
123 MAIN STREET, SUMMITVILLE, NY 27834

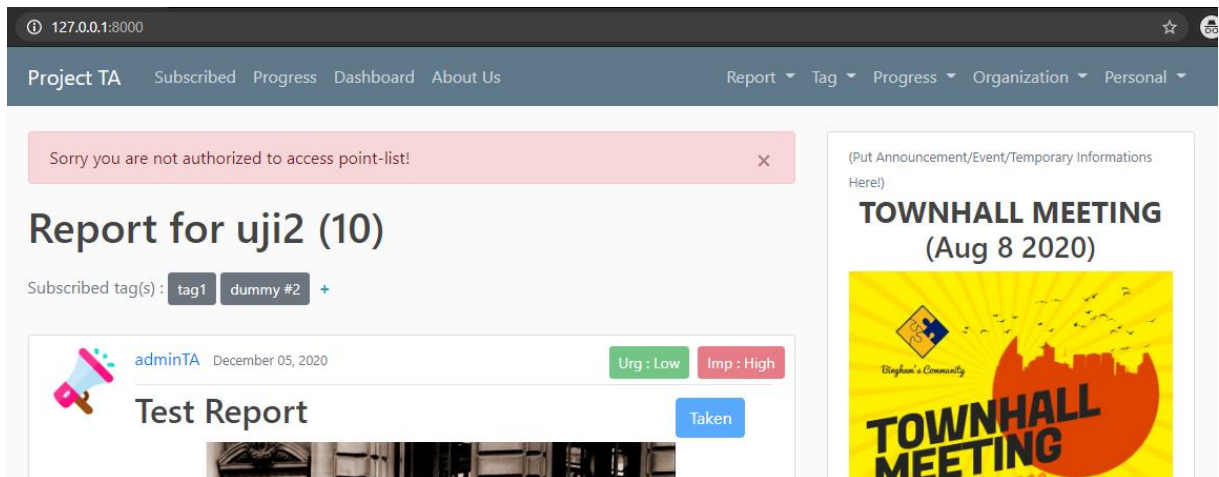
FOR MORE INFORMATION OR TO RSVP PLEASE CALL (303) 854 3454

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sem mi, pellentesque eget diam non, gravida pretium leo. In pretium suscipit leo, eu hendrerit augue ullamcorper ac. Praesent ex nisi, viverra a est et, varius ultricies turpis. Ut vel nunc est.

Gambar B.83 Cuplikan layar laman *'point-list'* pada pengujian pada peramban web pertama

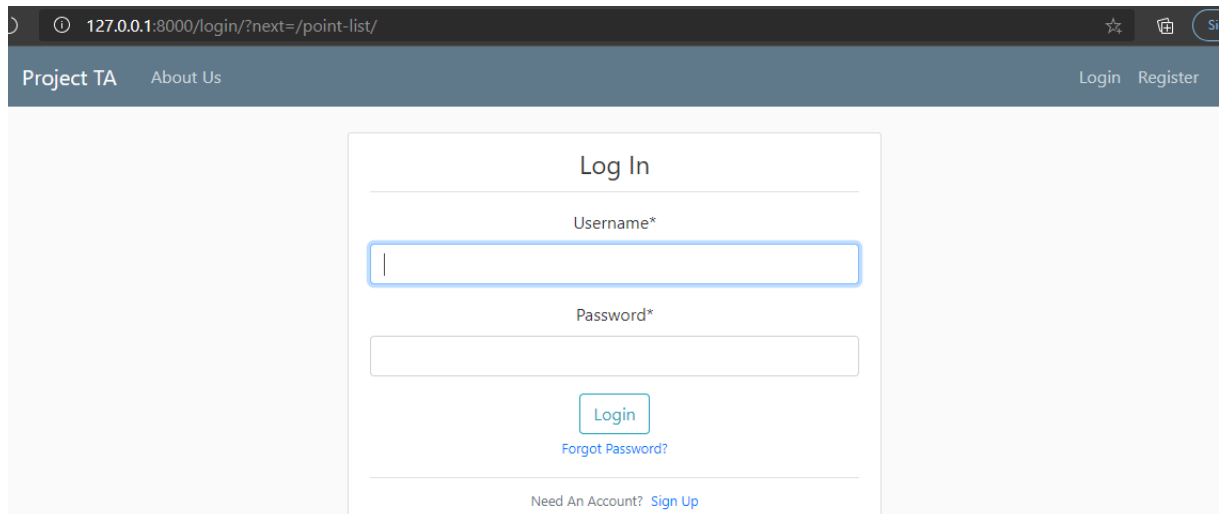
Pada peramban pertama, laman *'point-list'* dapat diakses. Semua elemen terkait dengan filter seperti kolom isian dan tombol bertuliskan "Search" dapat bekerja dengan baik. Filter juga bekerja dengan baik. Pada kasus ini pengguna mencari log poin yang tercatat setelah 1 November 2020 dan log poin yang dicari bernilai diantara -2 dan -3. Hasil yang muncul adalah semua log poin yang sesuai dengan ketiga kriteria di atas seperti pada gambar B.84.

Pada peramban kedua, laman *'point-list'* tidak dapat diakses dan dialihkan ke laman *'home'*. Cuplikan pengujian ini dapat terlihat pada gambar B.85.



Gambar B.84 Cuplikan layar laman *'point-list'* pada pengujian pada peramban web kedua

Pada peramban ketiga, laman *'point-list'* tidak dapat diakses dan dialihkan ke laman *'login'*. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman *'point-list'*. Cuplikan pengujian ini dapat terlihat pada gambar B.86.



Gambar B.85 Cuplikan layar laman *'point-list'* pada pengujian pada peramban web ketiga

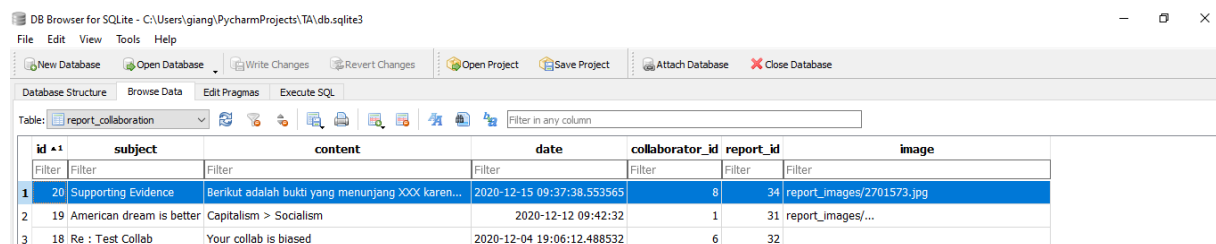
B.9. Pengujian Fitur 9 “Berkolaborasi pada Laporan” (*Collaborate on a Report*)

Pengujian pada fitur 9 “Berkolaborasi pada Laporan” akan meliputi pengujian laman ‘*report-detail*’ lagi, walaupun telah diuji di sub sub bab B.1. Namun elemen interaksi yang diuji pada sub sub bab ini akan berupa elemen yang terkait kolaborasi pada suatu laporan. Sedangkan laman ‘*user-collab*’ yang terkait dengan fitur 9 tidak akan dibahas lagi karena telah dibahas di sub sub bab B.2. Sebelum dilakukan pengujian, otorisasi akun uji 2 atas fitur 9 akan dimatikan, sedangkan otoritas akun uji2 atas fitur lain tetap diberikan.

Pada pengujian lanjutan untuk laman ‘*report-detail*’, tidak akan diuji aksesibilitasnya lagi. Seperti halnya pada sub sub bab B.4 akan diuji perspektif dimana akun adminTA akan merepresentasikan pengguna dengan otoritas atas fitur 9 sehingga dapat berkolaborasi pada suatu laporan. Sedangkan di lain sisi digunakan akun uji 2 untuk merepresentasikan pengguna tanpa otoritas atas fitur 9 sehingga tidak dapat berkolaborasi. Dan pada kasus pengguna yang tidak masuk (*login*) dari peramban ketiga tidak akan diuji lagi karena telah dibahas pada sub sub bab B.1. Kasus menggunakan akun adminTA dan akun uji2 akan menghasilkan interaksi elemen yang berbeda. Selain itu juga akan diuji pengunggahan data kolaborasi dan penghapusan kolaborasi dari kasus yang dapat berkolaborasi.

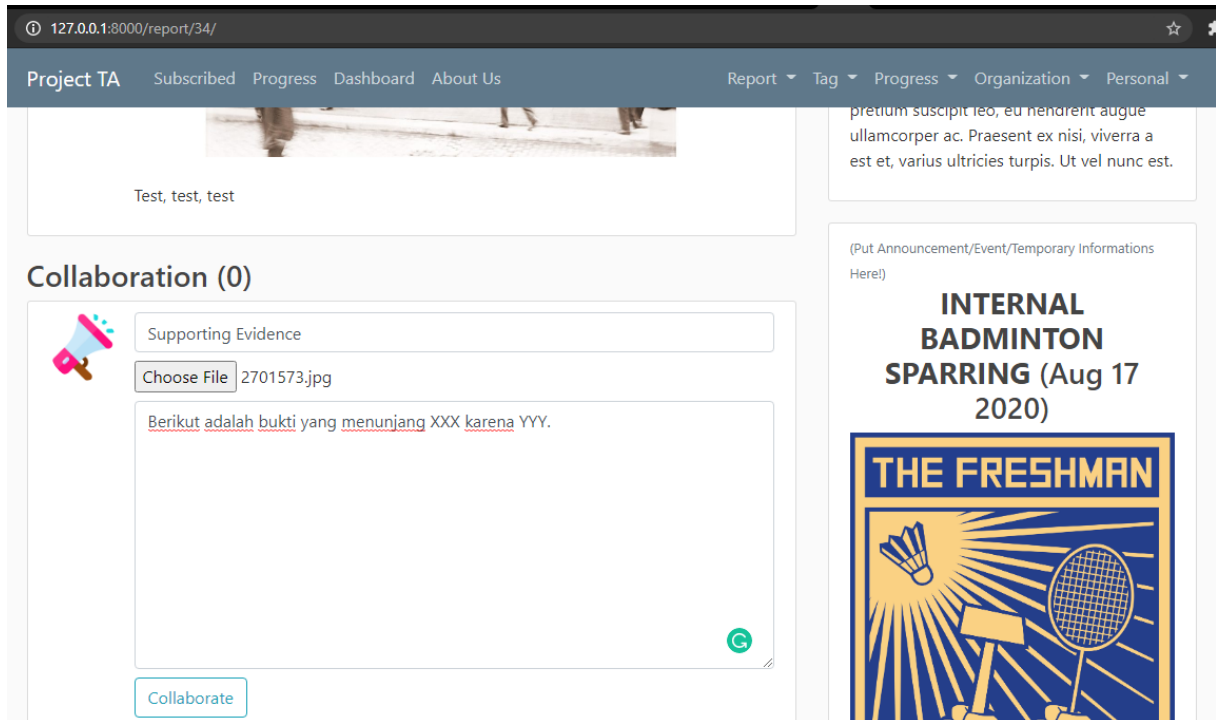
Pada kasus pengujian ini akan ditinjau kasus untuk berkolaborasi pada laporan berjudul “Test Report” dengan kunci primer pk=34. Sebelum pengujian laporan dengan pk=34 ini tidak memiliki kolaborasi.

Pada peramban pertama, di laman ‘*report-detail*’ milik laporan dengan pk=34, terdapat seksi konten (*content section*) yang terletak di paling bawah laman yang terdiri dari kolom isian untuk membuat kolaborasi baru. Dicoba kolom isian itu dan elemen untuk mengunggah gambar. Elemen dan kolom isian tersebut bekerja dengan baik. Hal tersebut dapat dilihat pada gambar B.88. Kemudian dipencet tombol bertuliskan “Collaborate” untuk membuat objek ‘*Collaboration*’ baru. Cuplikan basis data pada gambar B.87 menyoroti data yang berhasil diunggah.



id	subject	content	date	collaborator_id	report_id	image
1	20 Supporting Evidence	Berikut adalah bukti yang menunjang XXX karen...	2020-12-15 09:37:38.553565	8	34	report_images/2701573.jpg
2	19 American dream is better	Capitalism > Socialism	2020-12-12 09:42:32	1	31	report_images/...
3	18 Re : Test Collab	Your collab is biased	2020-12-04 19:06:12.488532	6	32	

Gambar B.86 Cuplikan layar basis data untuk model ‘*Collaboration*’ setelah pengujian laman ‘*report-detail*’ keempat




Gambar B.87 Cuplikan layar laman ‘report-detail’ pada pengujian keempat pada peramban web pertama

Sedangkan pada peramban kedua, di laman yang sama tidak terdapat seksi konten (*content section*) untuk membuat kolaborasi baru, seperti yang terlihat pada gambar B.89. Hal tersebut sesuai dengan rancangan karena akun uji2 pada peramban kedua tidak memiliki otoritas atas fitur 9. Kolaborasi yang telah dibuat oleh akun adminTA sebelumnya juga dapat terlihat pada pengujian laman ini.

Kemudian pada peramban pertama akan dilaksanakan pengujian kelima terhadap laman ‘report-detail’ milik laporan dengan pk=34. Pada kasus ini, terdapat kolaborasi yang dibuat oleh adminTA, yaitu kolaborasi yang baru dibuat sebelumnya. Karena merupakan kolaborasi yang dibuat oleh pengguna yang membuat laman tersebut, maka muncul tombol bertuliskan “Delete”. Tombol bertuliskan “Delete” bekerja dengan baik dan muncul elemen *modal* di atas laman yang mengonfirmasi keinginan pengguna untuk menghapus kolaborasi terkait seperti pada gambar B.90. Pengguna yakin untuk menghapus kolaborasi tersebut dan dipencet tombol bertuliskan “Delete” pada elemen *modal*. Proses penghapusan berjalan dengan baik dan kolaborasi yang telah dihapus telah menghilang dari basis data seperti pada gambar B.91.

127.0.0.1:8000/report/34/
Project TA
Subscribed
Progress
Dashboard
About Us
Report
Tag
Progress
Organization
Personal


adminTA
December 05, 2020


Test Report

Taker : adminTA

Progress : Not Approved 14Dec20

Laporan ini telah dilaporkan sebelumnya (merupakan duplikat)


Tags : tag1 dummy #2 1



Test, test, test

(Put Announcement/Event/Temporary Informations Here!)

TOWNHALL MEETING (Aug 8 2020)




Topic of Discussion

AT COMMUNITY CENTRE
123 MAIN STREET, SUMMERSVILLE, NY 27634


FOR MORE INFORMATION OR TO RSVP PLEASE CALL (303) 854 3854

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sem mi, pellentesque eget diam non, gravida pretium leo. In pretium suscipit leo, eu hendrerit augue ullamcorper ac. Praesent ex nisi, viverra a est et, varius ultricies turpis. Ut vel nunc est.

Collaboration (1)


adminTA
December 15, 2020


Supporting Evidence



Berikut adalah bukti yang menunjang XXX karena YYY.

(Put Announcement/Event/Temporary Informations Here!)

INTERNAL BADMINTON SPARRING (Aug 17 2020)

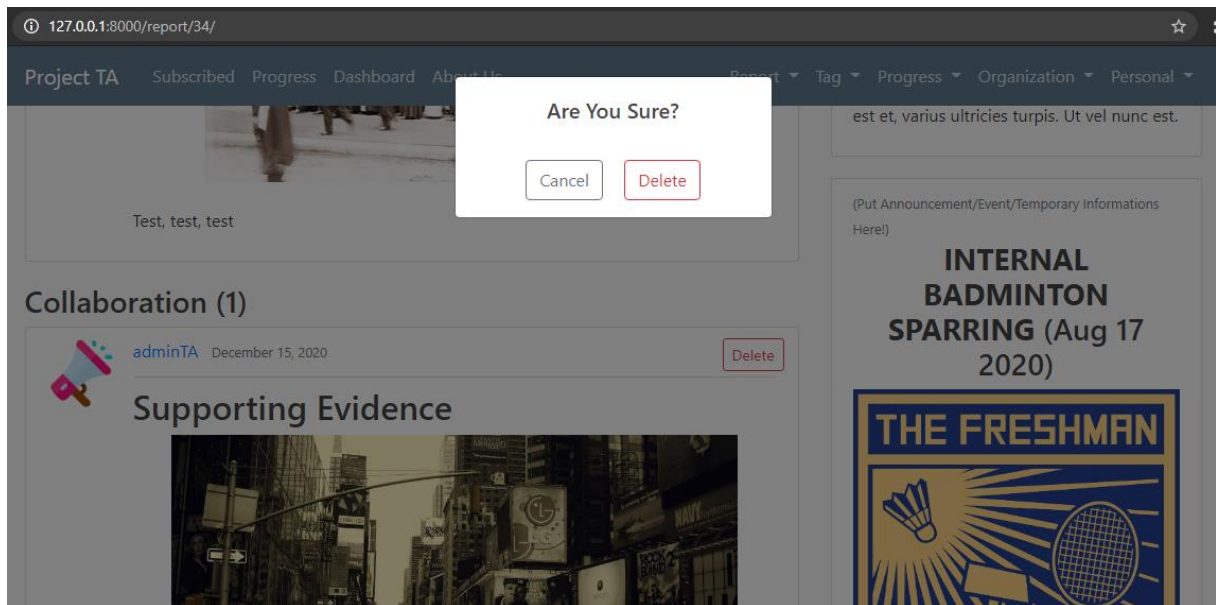


THE FRESHMAN

SINCE 2012

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sem mi, pellentesque eget diam non, gravida pretium leo. In pretium suscipit leo, eu hendrerit augue

Gambar B.88 Cuplikan layar laman ‘report-detail’ pada pengujian keempat pada peramban web kedua



Gambar B.89 Cuplikan layar laman *'report-detail'* pada pengujian kelima pada peramban web pertama

id	subject	content	date	collaborator_id	report_id	image
19	American dream is better	Capitalism > Socialism	2020-12-12 09:42:32	1	31	report_images/...
18	Re : Test Collab	Your collab is biased	2020-12-04 19:06:12.488532	6	32	

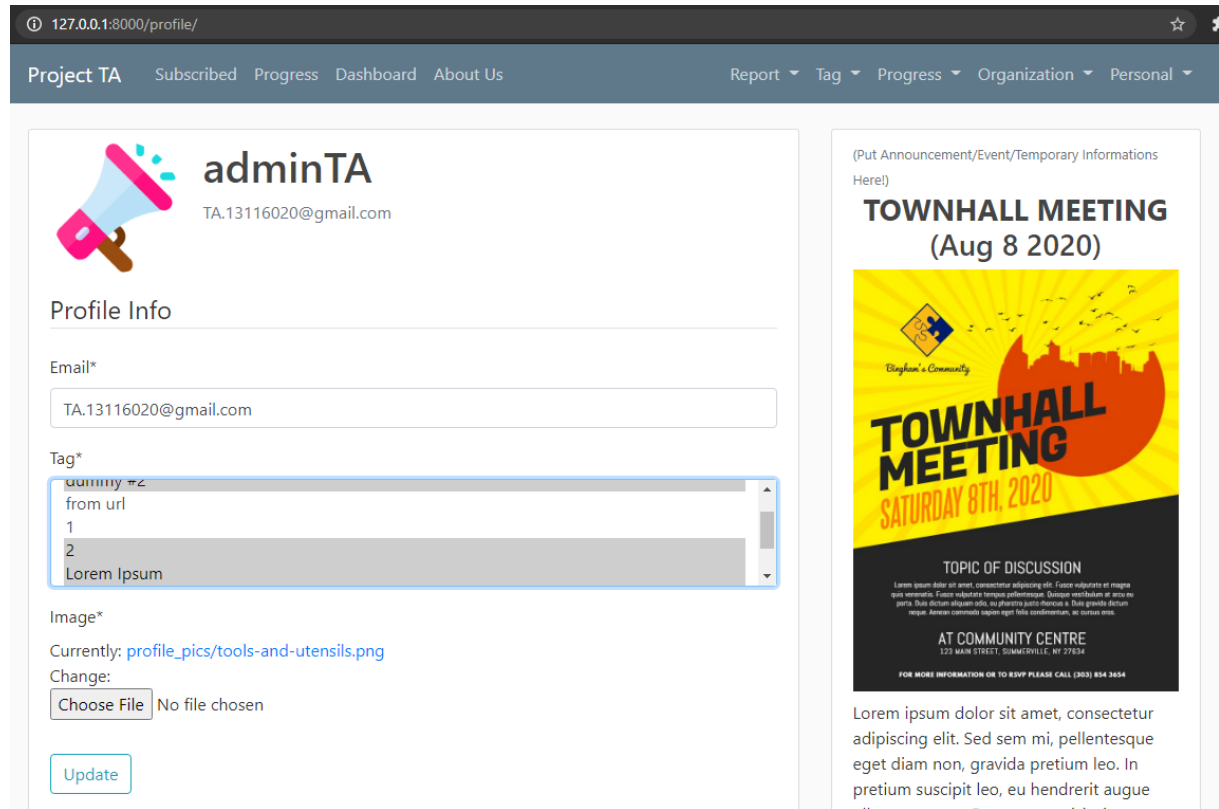
Gambar B.90 Cuplikan layar basis data untuk model *'Collaboration'* setelah pengujian laman *'report-detail'* kelima

B.10. Pengujian Fitur “Akun Pengguna”

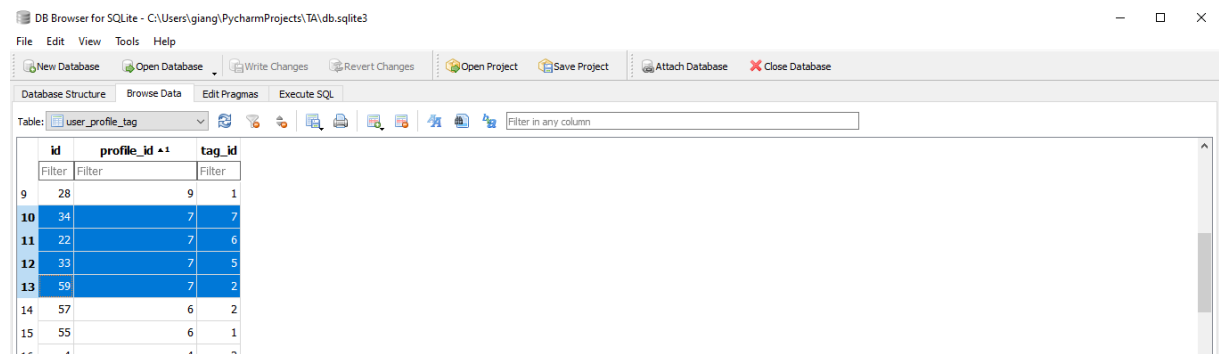
Pengujian pada fitur “Akun Pengguna” akan meliputi pengujian laman *'login'*, *'logout'*, *'password_reset'*, *'password_reset_done'*, *'password_reset_confirm'*, *'password_reset_complete'*, *'register'*, dan *'profile'*. Berbeda pada sub bab sebelumnya, karena mayoritas dari laman-laman ini tidak dilengkapi dengan perintah keamanan, maka laman - laman tersebut hanya akan diuji pada satu peramban. Kecuali untuk laman *'profile'*, karena dilengkapi dengan perintah keamanan, walaupun hanya untuk membatasi pengguna yang belum masuk (*login*). Maka dari itu pengujian laman *'profile'* hanya akan dilakukan pada dua peramban, peramban yang sudah masuk dan pengguna yang belum masuk.

Pada laman *'profile'* akan diuji aksesibilitasnya, interaksi elemennya, dan pengunggahan datanya. Interaksi elemen diuji dengan mengubah data pribadi terkait akun yang diuji, pada kasus ini adalah akun bernama pengguna adminTA. Pada kasus ini akan diubah *tag* yang dilanggan oleh akun adminTA. Sebelumnya, terdapat enam *tag* yang dilanggan oleh akun

adminTA seperti pada gambar B.26. Setelah itu pada laman ini, dihentikan langganan atas dua tag seperti pada gambar B.92. Setelah pengguna yakin, dipencet tombol bertuliskan “Upload”. Tombol berfungsi dengan baik, data terkirim, dan basis data telah berubah seperti pada gambar B.93.

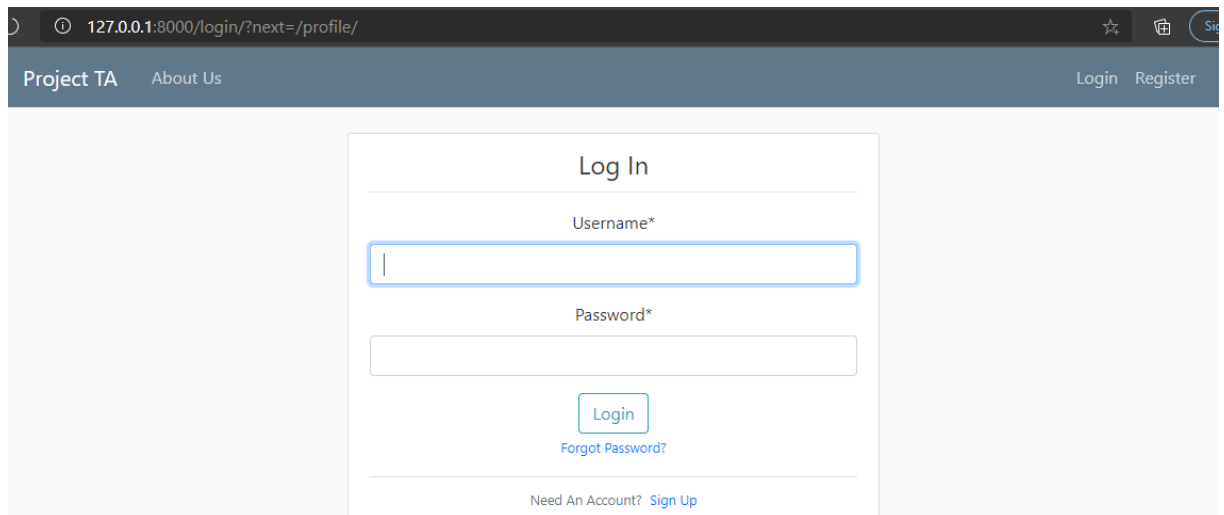


Gambar B.91 Cuplikan layar laman ‘profile’ pada pengujian pada pengguna yang telah masuk



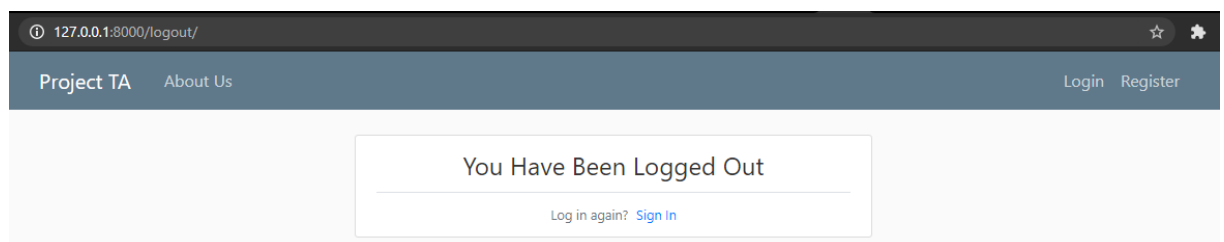
Gambar B.92 Cuplikan layar basis data untuk hubungan model ‘Profile’ dengan model ‘Tag’ setelah pengujian laman ‘profile’

Sedangkan pada pengguna yang belum masuk (*login*), laman ‘profile’ tidak dapat diakses dan dialihkan ke laman ‘login’. Bila pengguna masuk melalui laman ini, pengguna akan langsung dialihkan ke laman ‘profile’. Cuplikan pengujian ini dapat terlihat pada gambar B.94.



Gambar B.93 Cuplikan layar laman 'profile' pada pengujian pada pengguna yang belum masuk

Pada aktivitas keluar (*logout*) dapat bekerja dengan baik bila pengguna memencet tombol bertuliskan "Logout" pada menu *dropdown* yang muncul dari batang navigasi. Setelah dikeluarkan, pengguna akan dialihkan ke laman 'logout' seperti pada gambar B.95.



Gambar B.94 Cuplikan layar laman 'logout' pada pengujian fitur "Akun Pengguna"

Kemudian akan dicoba untuk membuat akun baru pada laman 'register'. Interaksi elemen yang diuji adalah kolom isian data mengenai akun baru. Kolom isian bekerja dengan baik dan dapat menampung data. Hal tersebut dapat dilihat pada gambar B.98. Kemudian dipencet tombol bertuliskan "Sign Up" untuk membuat akun baru. Tombol bekerja dengan baik dan terbentuk objek 'User' baru dan objek 'Profile' baru terkait akun yang baru dibuat. Hal tersebut dapat dilihat pada cuplikan basis data pada gambar B.96 dan gambar B.97.

DB Browser for SQLite - C:\Users\giang\P\PycharmProjects\TA\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: auth_user

id	password	last_login	is_superuser	username	email	is_staff	is_active	date_joined	last_name
1	pbkdf2_sha256\$180000\$KINIMR9Xc6sq\$WxS+C...	NULL	0	staffTekProd	messatekprodians@gmail.com	0	1	2020-12-15 15:24:05.138260	
2	16: nhlvdF7 eha75661800004F3rVuuMhTYn1kvdM/	2020-08-25 08:43:34 056R7R 0	0	liluh2	utilhm@urli.com	0	1	2020-08-25 08:43:08 638512	

Gambar B.95 Cuplikan layar basis data untuk model 'User' setelah pengujian fitur "Akun Pengguna"

DB Browser for SQLite - C:\Users\giang\PycharmProjects\TA\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: user_profile

id	image	user_id	unit_id
1	14 default.jpg	17	NULL

Gambar B.96 Cuplikan layar basis data untuk model *'Profile'* setelah pengujian fitur “Akun Pengguna”

127.0.0.1:8000/register/

Project TA About Us Login Register

Register

Username*

staffTekProd

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

massatekprodians@gmail.com

Password*

.....

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

.....

Enter the same password as before, for verification.

Sign Up

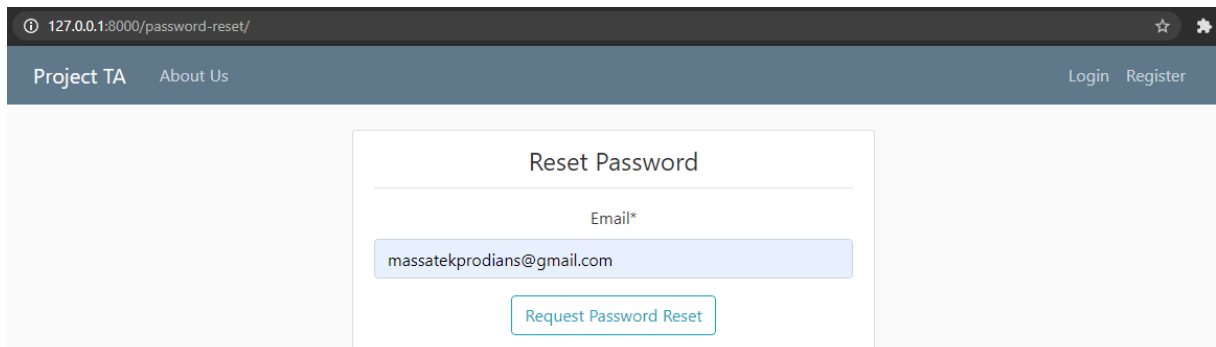
Already Have An Account? [Sign In](#)

Gambar B.97 Cuplikan layar laman *'register'* pada pengujian fitur “Akun Pengguna”

Setelah itu diuji aktivitas terkait dengan penyetelan ulang kata sandi. Aktivitas yang berkaitan dengan penyetelan ulang kata sandi akan melibatkan laman *'password_reset'*, *'password_reset_done'*, *'password_reset_confirm'*, dan *'password_reset_complete'*. Pada pengujian ini akan dicoba menyetel ulang kata sandi untuk akun staffTekProd yang baru saja dibuat pada pengujian laman *'register'*.

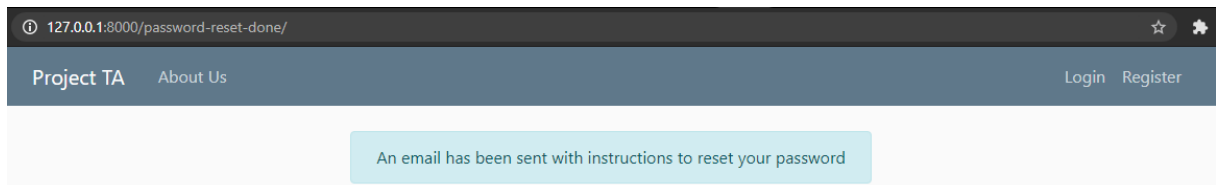
Pada awal penyetelan kata sandi, pengguna akan masuk ke laman *'password_reset'* dan memasukkan alamat surel terkait dengan akun yang akan disetel ulang kata sandinya. Pada laman ini kolom isian bekerja dengan baik. Hal tersebut dapat dilihat pada gambar B.99. Kemudian dipencet tombol bertuliskan “Request Password Reset” untuk mengirimkan surel

(*email*) konfirmasi penyetelan ulang kata sandi ke alamat surel terkait. Tombol bekerja dengan baik dan pengguna dialihkan ke laman '*password_reset_done*'.

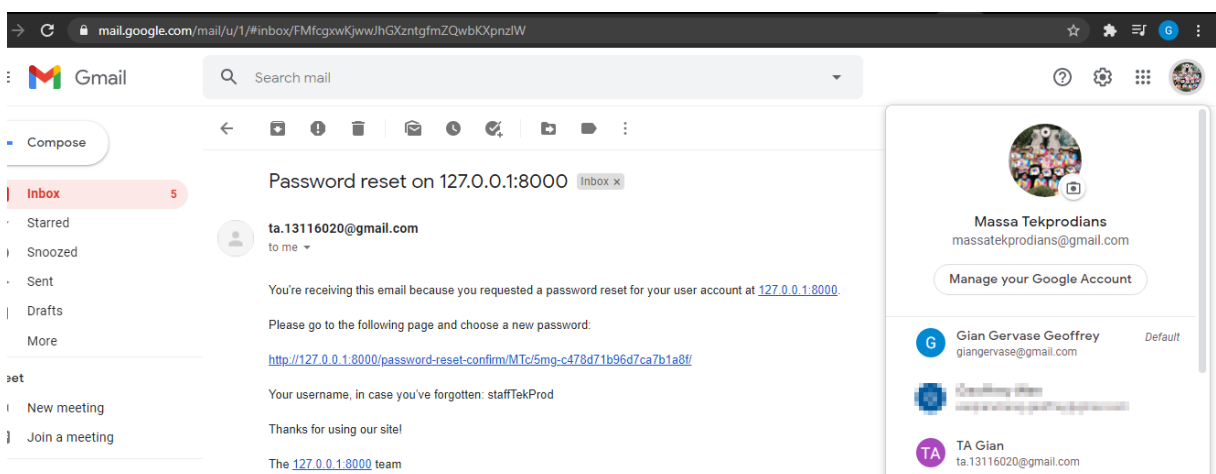


Gambar B.98 Cuplikan layar laman '*password_reset*' pada pengujian fitur “Akun Pengguna”

Laman '*password_reset_done*' akan memberi umpan balik (*feedback*) kepada pengguna bahwa surel konfirmasi penyetelan ulang kata sandi telah dikirim. Laman tersebut dapat terlihat pada gambar B.100. Surel konfirmasi tersebut terkirim dengan baik, pada kasus ini surel konfirmasi dikirim dari alamat surel ta.13116020@gmail.com sebagai alamat surel peladen ke alamat surel massatekprodians@gmail.com sebagai pemohon penyetelan ulang kata sandi. Surel konfirmasi penyetelan kata sandi tersebut berisi tautan (*link*) untuk menyetel ulang kata sandi untuk akun terkait. Surel tersebut dapat dilihat pada gambar B.101.



Gambar B.99 Cuplikan layar laman '*password_reset_done*' pada pengujian fitur “Akun Pengguna”



Gambar B.100 Cuplikan layar laman surel konfirmasi yang dikirimkan pada pengujian fitur “Akun Pengguna”

Bila pengguna masuk pada tautan pada surel konfirmasi tersebut, pengguna akan dialihkan ke laman *'password_reset_confirm'*. Pada laman tersebut pengguna diminta memasukkan kata sandi baru dan mengulangi lagi untuk konfirmasi pada kolom isian yang tersedia. Kolom isian bekerja dengan baik. Hal tersebut dapat dilihat pada gambar B.102. Setelah pengguna yakin dengan kata sandi baru, pengguna memencet tombol bertuliskan “Reset Password”. Kata sandi akan terganti dan pengguna dialihkan ke laman *'password_reset_complete'*.

Gambar B.101 Cuplikan layar laman *'password_reset_confirm'* pada pengujian fitur “Akun Pengguna”

Laman *'password_reset_complete'* akan memberi umpan balik (*feedback*) kepada pengguna bahwa kata sandi telah berhasil disetel ulang. Laman tersebut dapat terlihat pada gambar B.103. Adapun kata sandi pengguna pada basis data juga berubah seperti pada gambar B.104. Untuk menjamin keamanan kata sandi, kata sandi akan dienkrpsi menggunakan fungsi *hash* SHA-256 sebelum disimpan pada basis data. Pembaca basis data tidak dapat mengetahui pasti kata sandi pengguna, namun pembaca basis data dapat mengetahui bila ada perubahan basis data seperti pada kasus ini, dari gambar B.96 menjadi seperti pada gambar B.104.

Gambar B.102 Cuplikan layar laman *'password_reset_complete'* pada pengujian fitur “Akun Pengguna”

D8 Browser for SQLite - C:\Users\giang\P\PycharmProjects\TA\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: auth_user

id	password	last_login	is_superuser	username	email	is_staff	is_active	date_joined	last_name
1	pbkdf2_sha256\$18000\$Vbhw09QZqbr7\$ZsM5m...	NULL	0	staffTekProd	massatekprodians@gmail.com	0	1	2020-12-15 15:24:05.138260	
2	pbkdf2_sha256\$18000\$F3cVwVhIJYaJsx9M/...	2020-08-25 08:43:34.056878	0	uiiho2	uiiho@uii.com	0	1	2020-08-25 08:43:08.638512	

Gambar B.103 Cuplikan layar basis data untuk model ‘User’ setelah pengujian fitur “Akun Pengguna”

Kemudian diuji masuk menggunakan akun yang baru diganti kata sandinya pada laman ‘login’. Kolom isian pada laman tersebut bekerja dengan baik dan dapat menampung nama pengguna dan kata sandi dari akun tersebut. Hal tersebut dapat terlihat pada gambar B.105. Tombol bertuliskan “Login” juga bekerja dengan baik. Pengguna dapat masuk ke dalam aplikasi web setelah mengisi data dengan benar dan memencet tombol tersebut.

127.0.0.1:8000/login/

Project TA About Us Login Register

Log In

Username*

staffTekProd

Password*

.....

Login

[Forgot Password?](#)

Need An Account? [Sign Up](#)

Gambar B.104 Cuplikan layar laman ‘login’ pada pengujian fitur “Akun Pengguna”

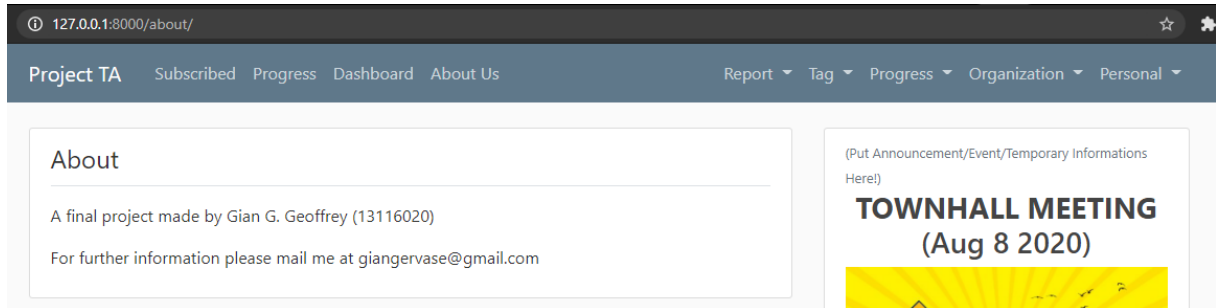
B.11. Pengujian Fitur “Tambahan”

Pengujian pada fitur “Tambahan” akan meliputi pengujian laman ‘about’, laman ‘dashboard’, dan seksi batang samping (*sidebar section*). Walaupun bukan merupakan bagian dari suatu fitur, seksi batang navigasi (*navbar section*) juga akan dibahas di sub sub bab ini.

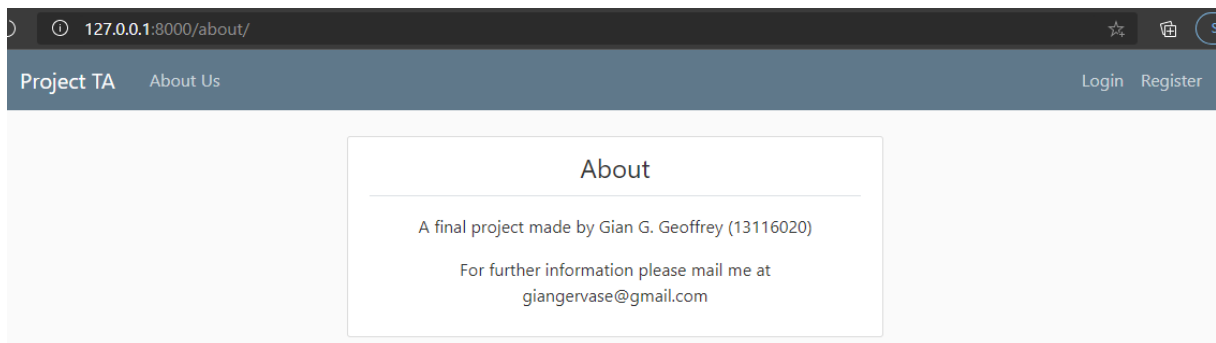
Untuk seksi batang samping akan muncul bila pengguna telah masuk (*login*) dan tidak muncul bila pengguna belum masuk. Hal tersebut dapat dilihat dari pengujian dan gambar cuplikan layar pada sub bab sebelumnya.

Begitu pula dengan seksi batang navigasi, tombol yang muncul disesuaikan dengan masuk atau tidaknya pengguna. Bila pengguna telah masuk, tombol yang muncul pun berbeda bergantung pada otoritas yang dimiliki pengguna tersebut.

Untuk laman *'about'* memang dapat diakses oleh pengguna baik yang telah masuk maupun yang belum masuk. Hal tersebut dapat terlihat pada gambar B.106 untuk kasus pengguna yang telah masuk dan gambar B.107 untuk kasus pengguna yang belum masuk.

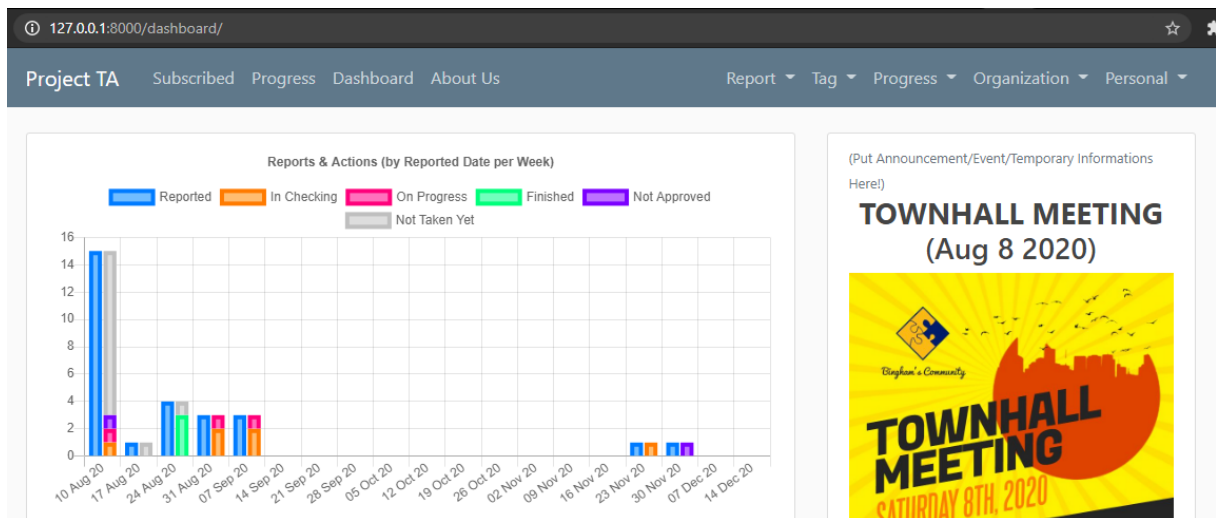


Gambar B.105 Cuplikan layar laman *'about'* pada pengujian pada pengguna yang telah masuk

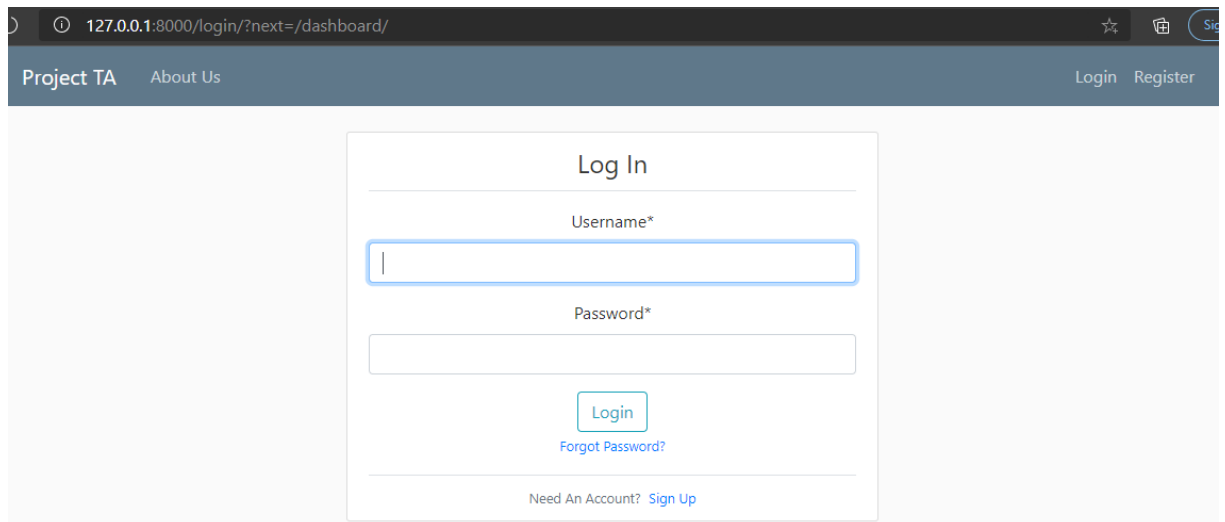


Gambar B.106 Cuplikan layar laman *'about'* pada pengujian pada pengguna yang belum masuk

Sedangkan untuk laman *'dashboard'* yang hanya dapat diakses oleh pengguna telah masuk. Sedangkan pengguna yang belum masuk akan dialihkan ke laman *'login'*. Hal tersebut dapat terlihat pada gambar B.108 untuk kasus pengguna yang telah masuk dan gambar B.109 untuk kasus pengguna yang belum masuk.



Gambar B.107 Cuplikan layar laman *'dashboard'* pada pengujian pada pengguna yang telah masuk



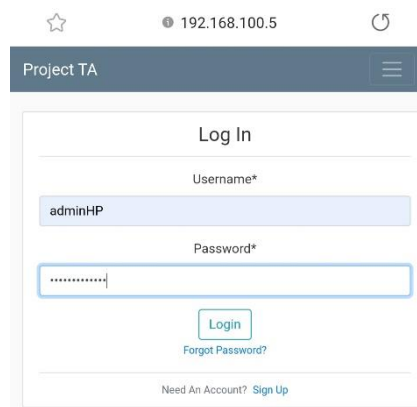
Gambar B.108 Cuplikan layar laman '*dashboard*' pada pengujian pada pengguna yang belum masuk

Lampiran C

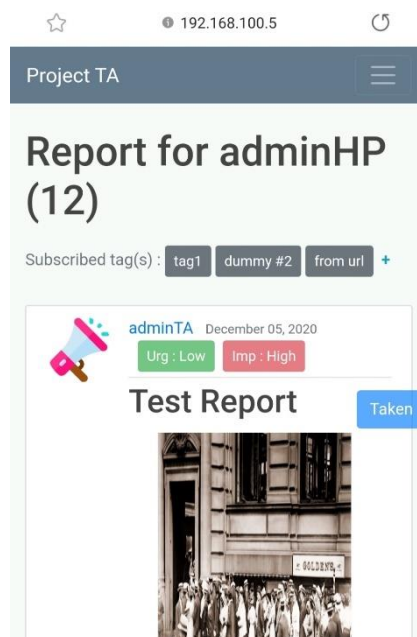
Pengujian Kinerja Aplikasi Web pada Jaringan WLAN

C.1. Pengujian Kinerja Aplikasi Web pada Ponsel Pintar

Dengan menggunakan ponsel pintar berbasis Android, pengguna akan menguji kinerja aplikasi web melalui jaringan WLAN. Pertama-tama dihubungkan jaringan Wi-Fi terkait dengan ponsel terkait. Selanjutnya pengguna akan masuk dengan menggunakan akun bernama pengguna adminHP seperti pada gambar C.1. Laman *'login'* beserta elemen yang ada disitu bekerja dengan baik dan pengguna dialihkan ke laman *'home'* seperti pada gambar C.2.



Gambar C.1 Cuplikan layar laman *'login'* pada pengujian masuk menggunakan ponsel



Gambar C.2 Cuplikan layar laman *'home'* setelah pengujian masuk menggunakan ponsel

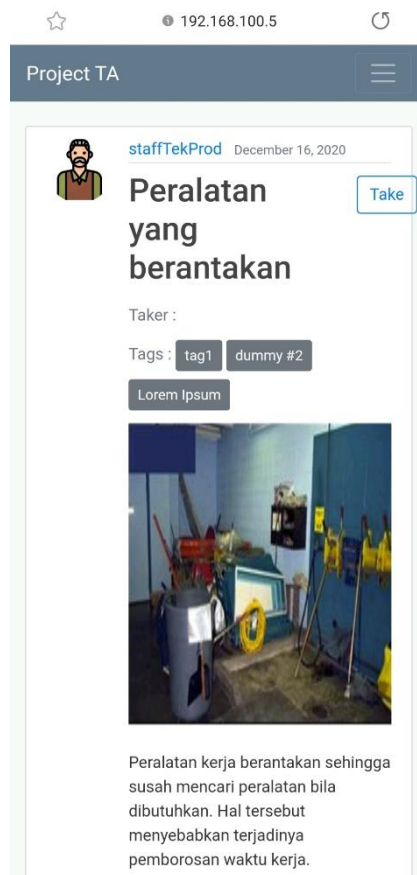
Kemudian diuji pembuatan laporan melalui laman '*report-create*' seperti pada gambar C.3. Seluruh elemen dapat berinteraksi dengan baik pada ponsel. Dan setelah dipencet tombol bertuliskan "Upload", laporan baru terbentuk. Laporan tersebut akan ditangani pada pengujian kinerja aplikasi web pada komputer (selain hos) yang dijabarkan pada sub bab C.2.

The screenshot shows a mobile application interface for creating a new report. At the top, there is a status bar with a star icon, the IP address 192.168.100.5, and a refresh icon. Below this is a dark blue header bar with the text 'Project TA' and a hamburger menu icon. The main content area is titled 'New Report' and contains several form fields: 'Title*' with the value 'Kebocoran keran', 'Content*' with a paragraph of text about a water leak, 'Image*' with a 'Choose file' button and a file name, 'Tag*' with a dropdown menu showing '0 selected', 'Urgency*' with a dropdown menu showing 'Low', and 'Importance*' with a dropdown menu showing 'Medium'. At the bottom of the form is an 'Upload' button.

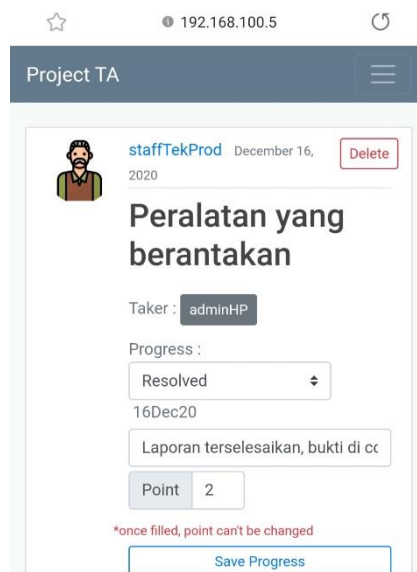
Gambar C.3 Cuplikan layar laman '*report-create*' pada pengujian pembuatan laporan menggunakan ponsel

Setelah membuat laporan, akan diuji mengambil laporan dan mengubah status laporan. Laporan yang akan diambil adalah laporan yang baru dibuat oleh akun staffTekProd seperti pada penjabaran pada sub bab C.2. Laporan tersebut dapat diambil karena pada laman '*report-detail*' dari laporan tersebut terdapat tombol bertuliskan "Take" seperti pada gambar C.4. Tombol tersebut bekerja dengan baik dan laporan berhasil diambil. Diuji juga pengubahan status dari laporan beserta catatan dan pemberian poin seperti gambar C.5. Kolom isian dan

kolom pilihan bekerja dengan baik dan data berhasil diunggah setelah memencet tombol “Save Progress”.

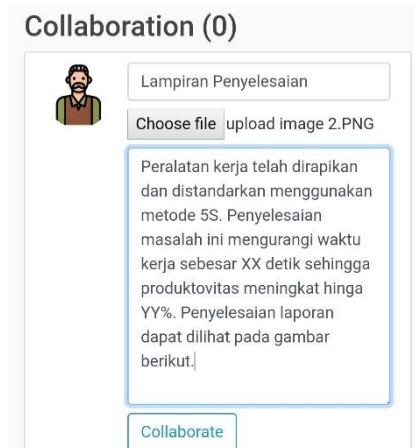


Gambar C.4 Cuplikan layar laman ‘report-detail’ pada pengujian pengambilan laporan menggunakan ponsel

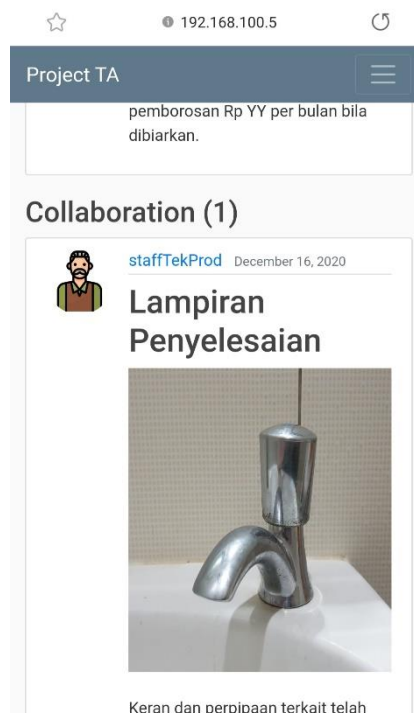


Gambar C.5 Cuplikan layar laman ‘report-detail’ pada pengujian perubahan status laporan menggunakan ponsel

Kemudian diuji untuk berkolaborasi pada laporan tersebut. Pada kasus ini kolaborasi yang disampaikan adalah mengenai penyelesaian laporan ini. Seluruh elemen pada seksi konten (*content section*) yang ditujukan untuk membuat kolaborasi baru dapat bekerja dengan baik. Data dapat diisi pada kolom isian dan gambar dapat diunggah pada elemen yang dapat menerima masukan gambar seperti pada gambar C.6. Setelah dipencet tombol bertuliskan “Collaborate”, terbentuk kolaborasi baru untuk laporan tersebut.



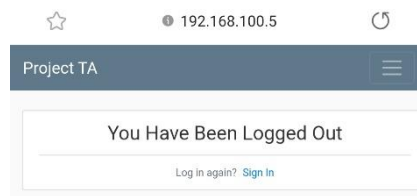
Gambar C.6 Cuplikan layar laman *'report-detail'* pada pengujian pembuatan kolaborasi menggunakan ponsel



Gambar C.7 Cuplikan layar laman *'report-detail'* setelah pengujian pembuatan kolaborasi menggunakan komputer lain

Bila pengguna dengan akun adminHP kembali pada laman ‘*report-detail*’ dari laporan yang ia baru buat (yang berjudul “Kebocoran Keran”), dapat dilihat kolaborasi yang dibuat oleh akun staffTekProd mengenai penyelesaian laporan mengenai pemborosan air tersebut seperti pada gambar C.7. Pembuatan kolaborasi tersebut dijabarkan pada sub bab C.2.

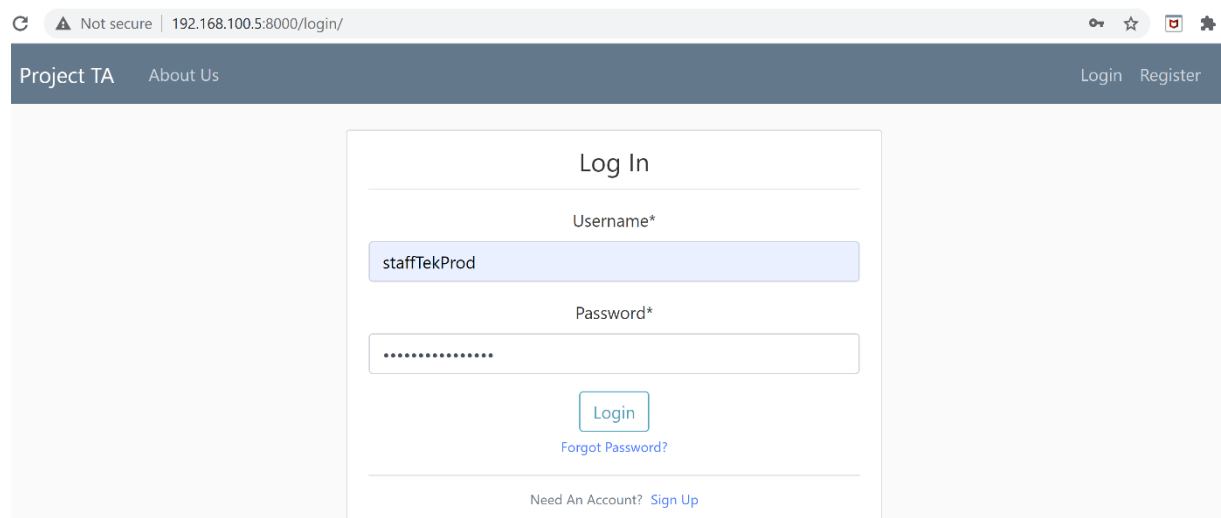
Untuk mengakhiri pengujian kinerja aplikasi web menggunakan ponsel, akan diuji untuk mengeluarkan pengguna dari aplikasi web dengan memencet tombol bertuliskan “Logout” pada batang navigasi. Setelah tombol terkait dipencet, pengguna dikeluarkan dari aplikasi web dan dialihkan ke laman ‘*logout*’ seperti pada gambar C.8.



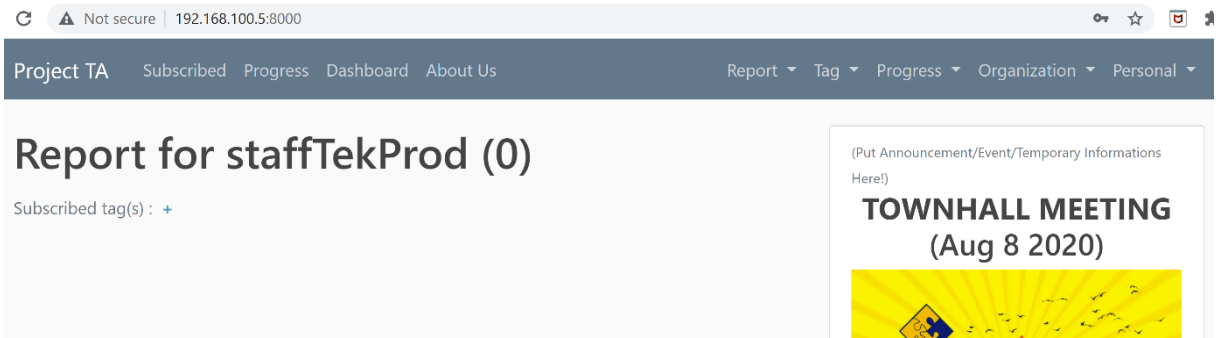
Gambar C.8 Cuplikan layar laman ‘*logout*’ setelah pengujian keluar menggunakan ponsel

C.2. Pengujian Kinerja Aplikasi Web pada Komputer (Selain Hos)

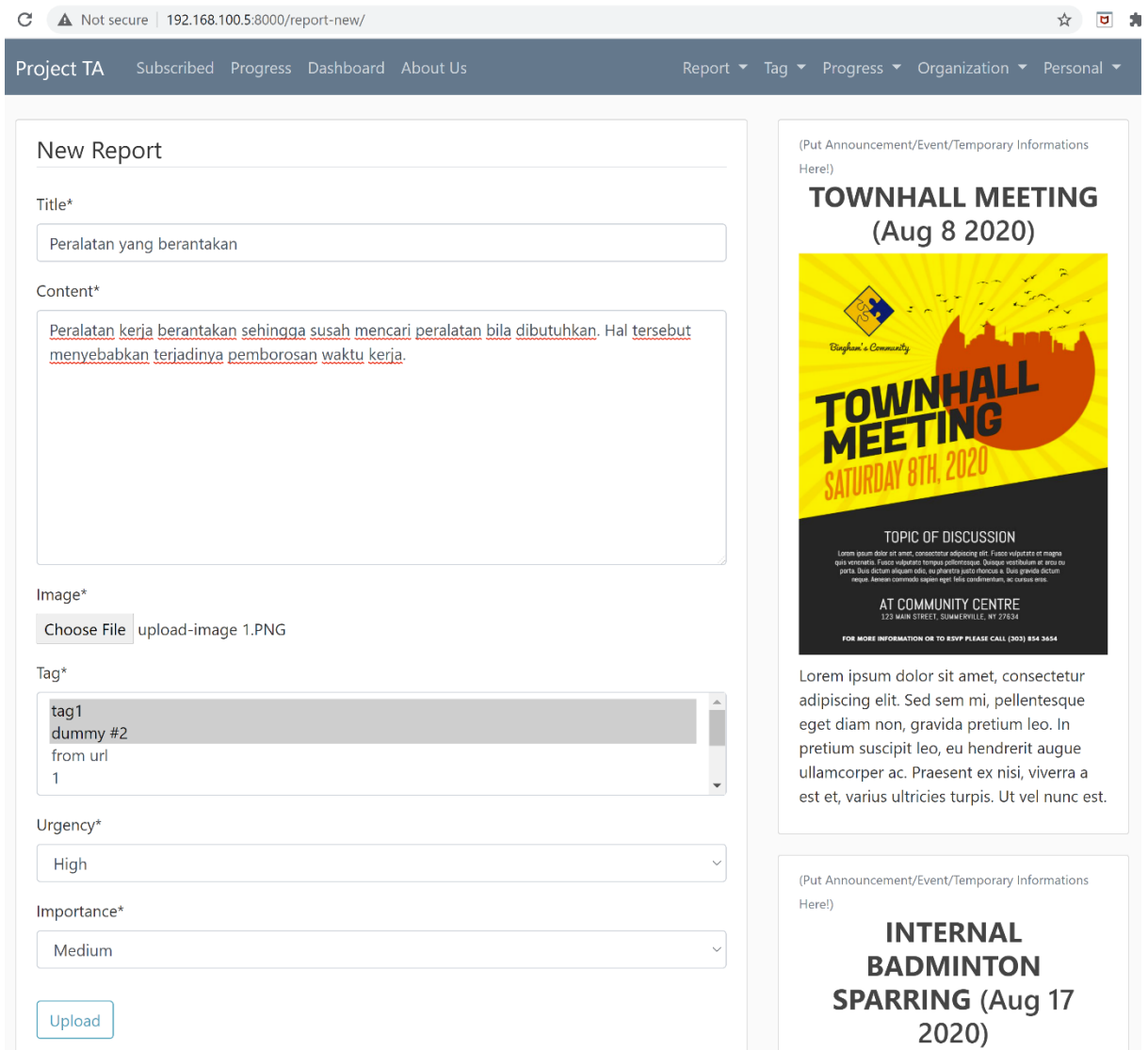
Dengan menggunakan komputer (selain hos) berbasis Window, pengguna akan menguji kinerja aplikasi web melalui jaringan WLAN. Pertama-tama dihubungkan jaringan Wi-Fi terkait dengan komputer terkait. Selanjutnya pengguna akan masuk dengan menggunakan akun bernama staffTekProd seperti pada gambar C.9. Laman ‘*login*’ beserta elemen yang ada disitu bekerja dengan baik dan pengguna dialihkan ke laman ‘*home*’ seperti pada gambar C.10. Pada kasus ini akun staffTekProd belum melanggan satu *tag* puh, sehingga laman ‘*home*’ yang ia akses juga kosong.



Gambar C.9 Cuplikan layar laman ‘*login*’ pada pengujian masuk menggunakan komputer lain

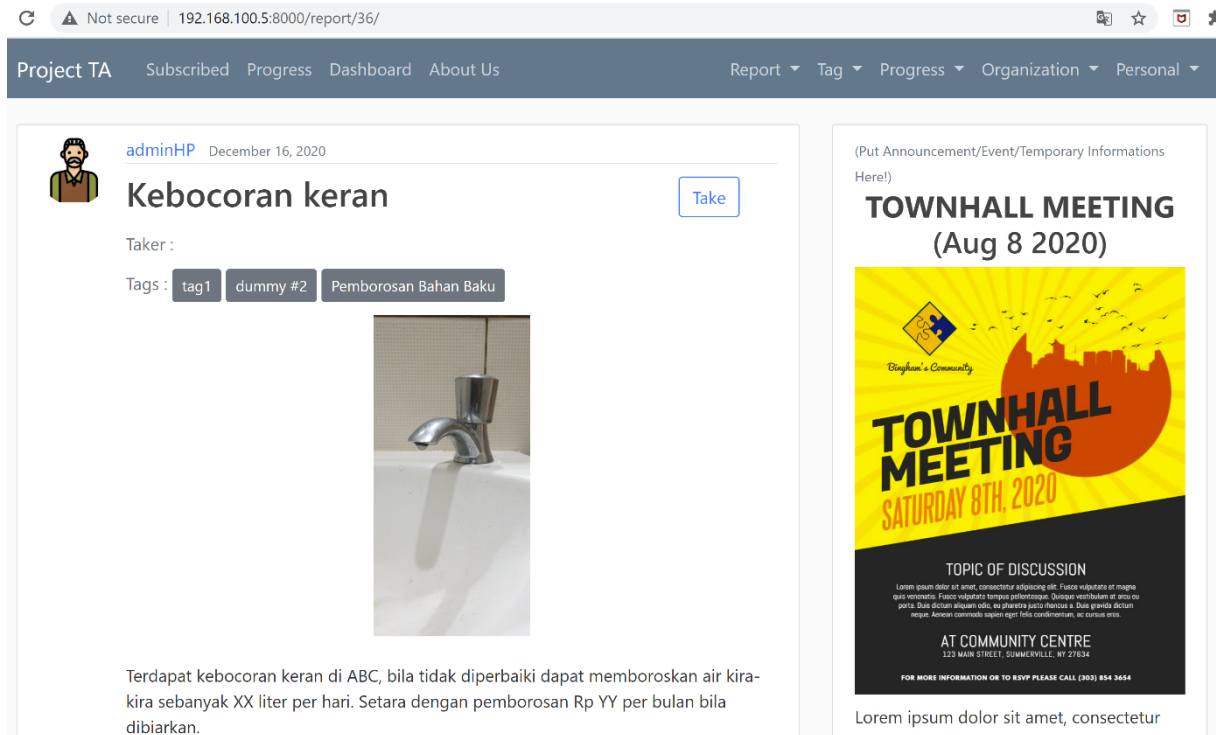


Gambar C.10 Cuplikan layar laman 'home' setelah pengujian masuk menggunakan komputer lain

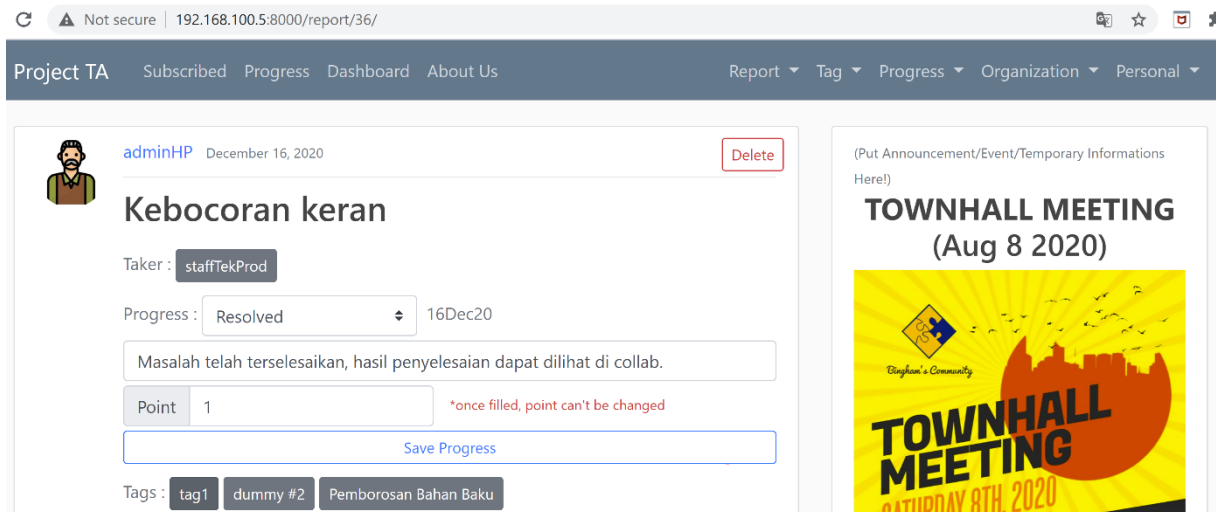


Gambar C.11 Cuplikan layar laman 'report-create' pada pengujian pembuatan laporan menggunakan komputer lain

Kemudian diuji pembuatan laporan melalui laman *'report-create'* seperti pada gambar C.11. Seluruh elemen dapat berinteraksi dengan baik pada ponsel. Dan setelah dipencet tombol bertuliskan “Upload”, laporan baru terbentuk. Laporan tersebut akan ditangani pada pengujian kinerja aplikasi web pada ponsel pintar yang dijabarkan pada sub bab C.1.



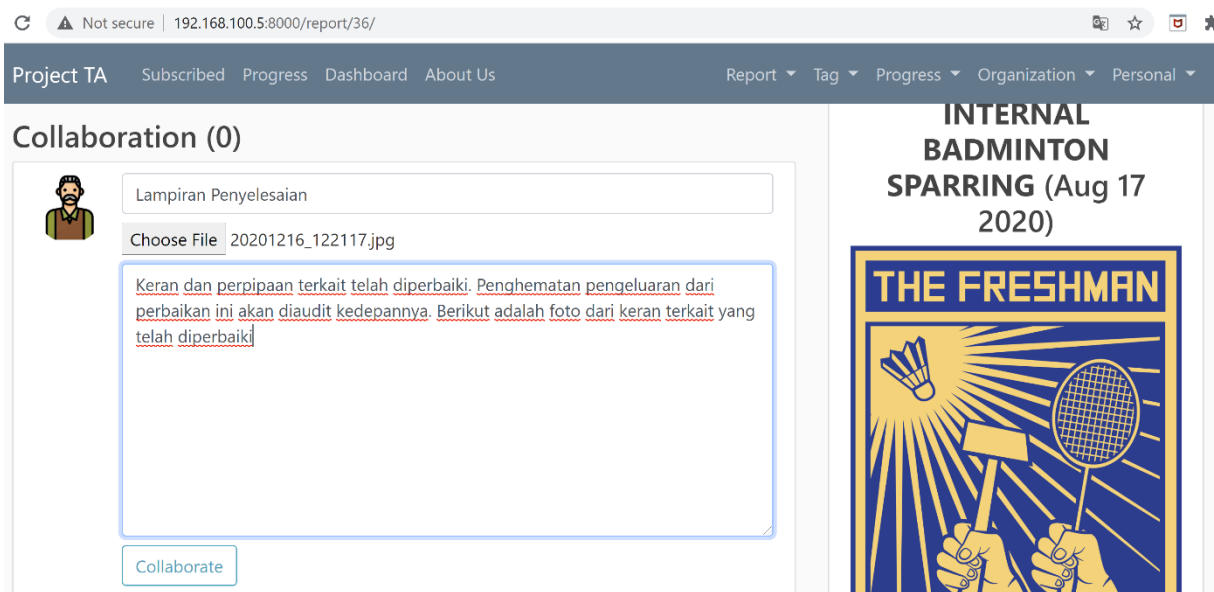
Gambar C.12 Cuplikan layar laman *'report-detail'* pada pengujian pengambilan laporan menggunakan komputer lain



Gambar C.13 Cuplikan layar laman *'report-detail'* pada pengujian perubahan status laporan menggunakan komputer lain

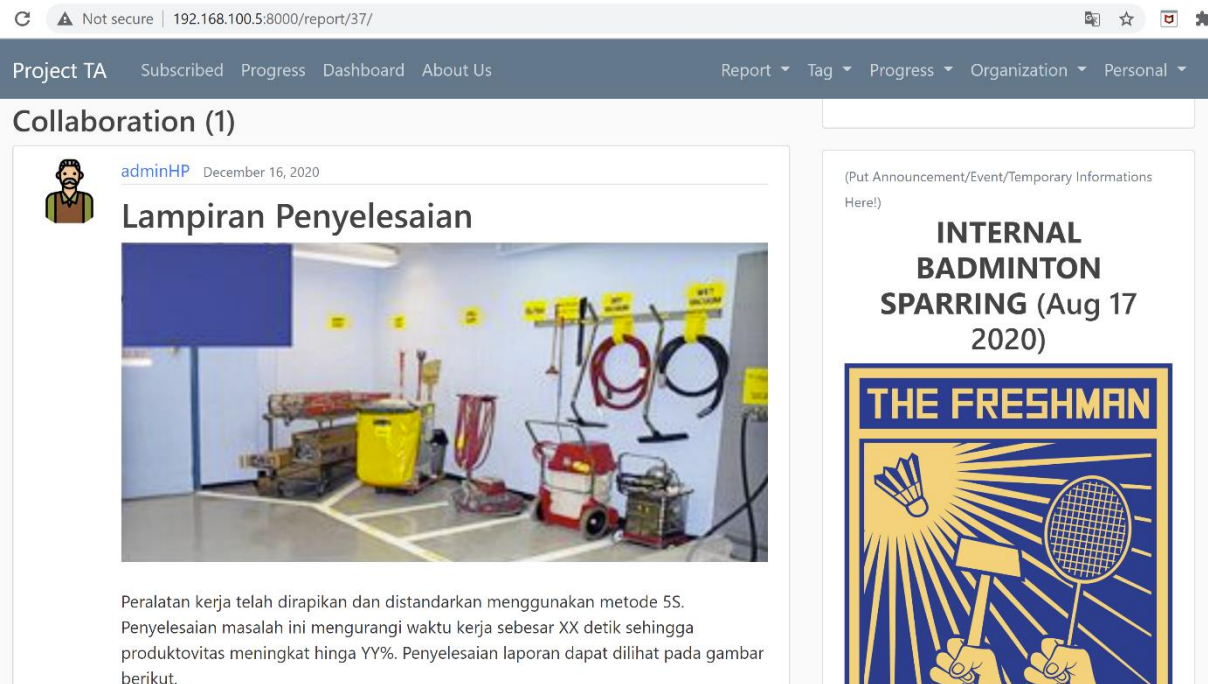
Setelah membuat laporan, akan diuji mengambil laporan dan mengubah status laporan. Laporan yang akan diambil adalah laporan yang baru dibuat oleh akun adminHP seperti pada penjabaran pada sub bab C.1. Laporan tersebut dapat diambil karena pada laman *'report-detail'* dari laporan tersebut terdapat tombol bertuliskan "Take" seperti pada gambar C.12. Tombol tersebut bekerja dengan baik dan laporan berhasil diambil. Diuji juga pengubahan status dari laporan beserta catatan dan pemberian poin seperti gambar C.13. Kolom isian dan kolom pilihan bekerja dengan baik dan data berhasil diunggah setelah memencet tombol "Save Progress".

Kemudian diuji untuk berkolaborasi pada laporan tersebut. Pada kasus ini kolaborasi yang disampaikan adalah mengenai penyelesaian laporan ini. Seluruh elemen pada seksi konten (*content section*) yang ditujukan untuk membuat kolaborasi baru dapat bekerja dengan baik. Data dapat diisi pada kolom isian dan gambar dapat diunggah pada elemen yang dapat menerima masukan gambar seperti pada gambar C.14. Setelah dipencet tombol bertuliskan "Collaborate", terbentuk kolaborasi baru untuk laporan tersebut.



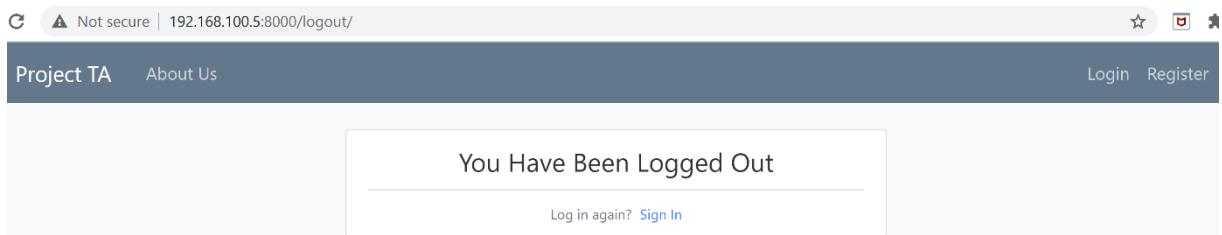
Gambar C.14 Cuplikan layar laman *'report-detail'* pada pengujian pembuatan kolaborasi menggunakan komputer lain

Bila pengguna dengan akun staffTekProd kembali pada laman *'report-detail'* dari laporan yang ia baru buat (yang berjudul "Peralatan yang Berantakan"), dapat dilihat kolaborasi yang dibuat oleh akun adminHP mengenai penyelesaian laporan mengenai peralatan kerja tersebut seperti pada gambar C.15. Pembuatan kolaborasi tersebut dijabarkan pada sub bab C.1.



Gambar C.15 Cuplikan layar laman *'report-detail'* setelah pengujian pembuatan kolaborasi menggunakan ponsel

Untuk mengakhiri pengujian kinerja aplikasi web menggunakan komputer (selain hos), akan diuji untuk mengeluarkan pengguna dari aplikasi web dengan memencet tombol bertuliskan "Logout" pada batang navigasi. Setelah tombol terkait dipencet, pengguna dikeluarkan dari aplikasi web dan dialihkan ke laman *'logout'* seperti pada gambar C.16.



Gambar C.16 Cuplikan layar laman *'logout'* setelah pengujian keluar menggunakan komputer lain