

**SISTEM INFORMASI PENCATATAN BUG
BERBASIS WEB**



Disusun Oleh:

N a m a : Talitha Widya Utami

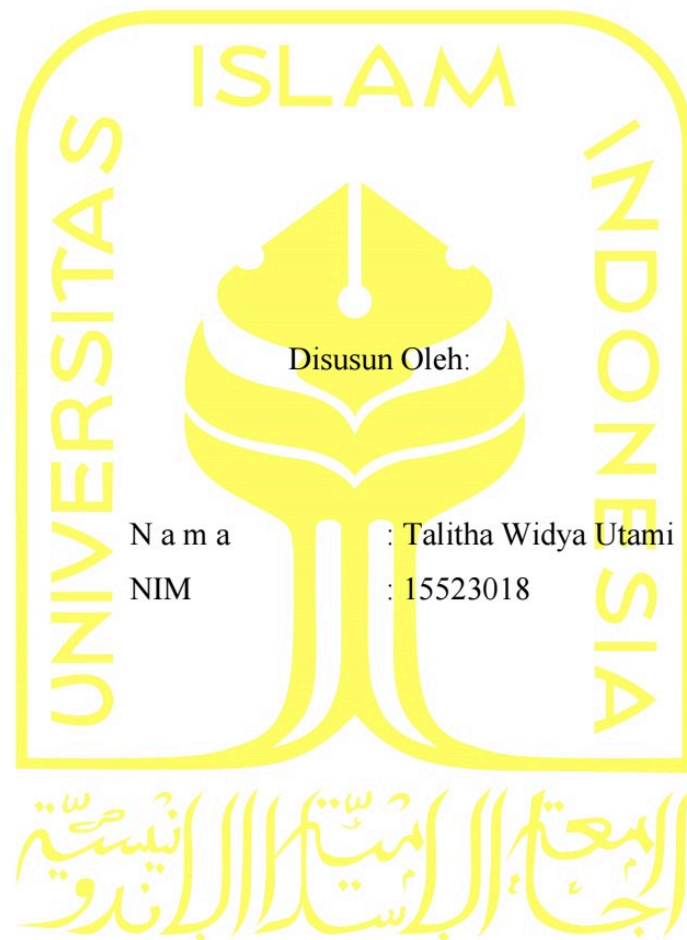
NIM : 15523018

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2021

HALAMAN PENGESAHAN DOSEN PEMBIMBING
SISTEM INFORMASI PENCATATAN BUG
BERBASIS WEB

TUGAS AKHIR



Yogyakarta, 15 Juli 2021

Pembimbing,


(Septia Rani, S.T., M.Cs.)

HALAMAN PENGESAHAN DOSEN PENGUJI

SISTEM INFORMASI PENCATATAN BUG

BERBASIS WEB

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 15 Juli 2021

Tim Penguji

Septia Rani, S.T., M.Cs.

Anggota 1

Aridhanyati Arifin, S.T., M.Cs.

Anggota 2

Hanson Prihantoro Putro, S.T., M.T.



Three handwritten signatures in black and blue ink, each placed above a horizontal line. The signatures are stylized and difficult to read.

Mengetahui,

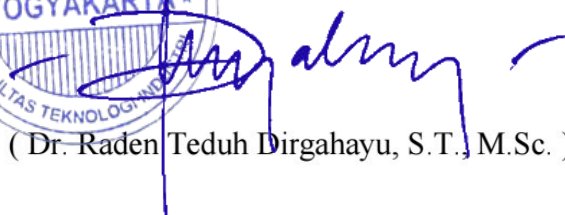
Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)



A handwritten signature in blue ink, placed over the official stamp. The signature is stylized and appears to read 'Raden Teduh Dirgahayu'.

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Talitha Widya Utami

NIM : 15523018

Tugas akhir dengan judul:

**SISTEM INFORMASI PENCATATAN BUG
BERBASIS WEB**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 15 Juli 2021



(Talitha Widya Utami)

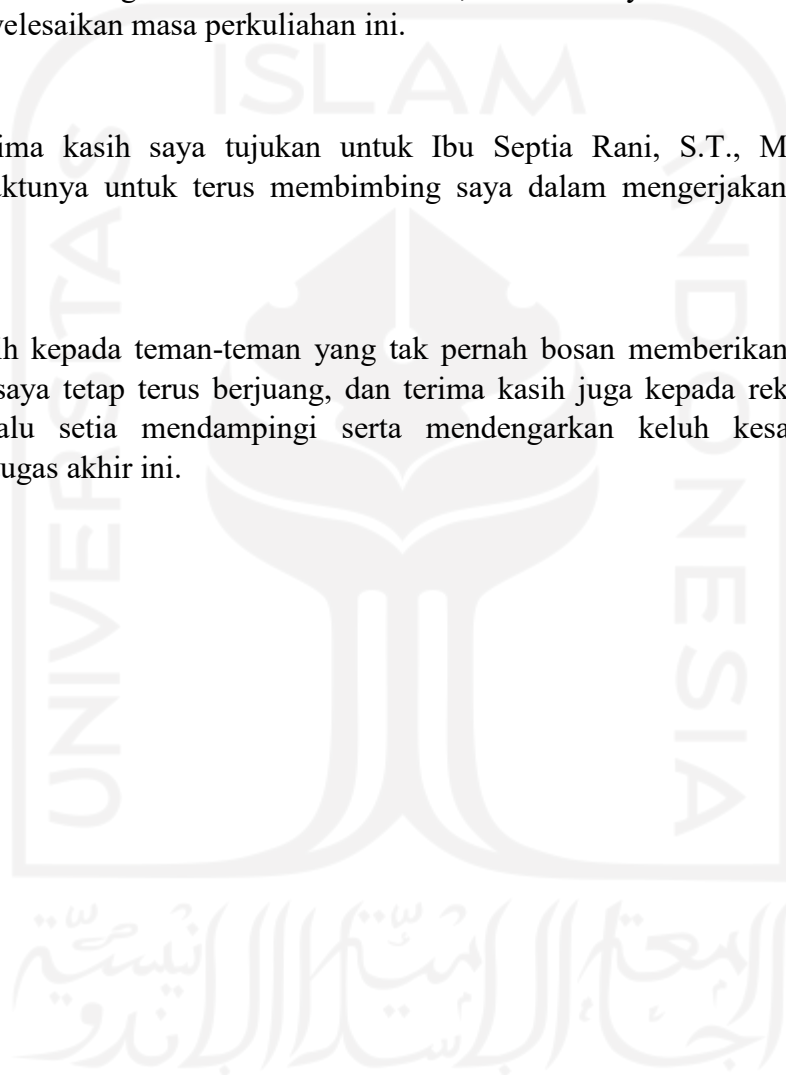
HALAMAN PERSEMBAHAN

Puji dan syukur Alhamdulillah saya haturkan kepada Allah SWT yang telah memberikan nikmat kesehatan dan kesempatan sehingga saya dapat menyelesaikan tugas akhir ini.

Tugas akhir ini saya persembahkan khusus untuk kedua orang tua saya yang senantiasa mendampingi, mendukung secara moril dan materil, serta tentunya memberi doa kepada saya agar dapat menyelesaikan masa perkuliahan ini.

Ucapan terima kasih saya tujukan untuk Ibu Septia Rani, S.T., M.Cs. yang telah meluangkan waktunya untuk terus membimbing saya dalam mengerjakan tugas akhir ini hingga selesai.

Terima kasih kepada teman-teman yang tak pernah bosan memberikan dukungan serta semangat agar saya tetap terus berjuang, dan terima kasih juga kepada rekan seperjuangan saya yang selalu setia mendampingi serta mendengarkan keluh kesah saya selama menyelesaikan tugas akhir ini.



HALAMAN MOTO

“May you find the light of God, whenever you are in darkness.”

“Barangsiapa diuji lalu bersabar, dikaruniai lalu bersyukur, didzalimi lalu memaafkan, dan mendzalimi lalu istighfar, maka bagi mereka keselamatan dan hidayah.”

(H.R. Al Baihaqi)

“Perhaps you hate a thing that is good for you, and perhaps you love a thing that is bad for you. Allah knows, while you don't.”

(2:216)

“Teguklah kesabaran, apabila ia membunuhmu, ia membunuhmu dalam keadaan syahid, dan apabila ia menghidupkanmu, ia menghidupkanmu dalam keadaan mulia.”

(Ibnul Qayyim rahimahullaah)

“Pray, because He always listens. Pray, because only He knows, yet He still listens.”

“There is hope, the is Allah.”

المعهد الإسلامي
الاستدلال والتدو

KATA PENGANTAR

Assalaamualaykum warahmatullaahi wabarakatuh.

Puji serta syukur saya haturkan kepada Allah SWT yang telah memberikan kesehatan dan kesempatan bagi saya untuk bisa menyelesaikan tugas akhir yang berjudul “Sistem Informasi Pencatatan Bug Berbasis Web” ini dalam rangka memenuhi salah satu syarat untuk menyelesaikan studi dan memperoleh gelar sarjana Strata-I Program Studi Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

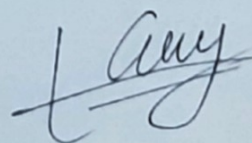
Saya menyadari bahwa dalam penyelesaian tugas akhir ini tak terlepas dari dukungan serta bantuan dari banyak pihak yang ada disekeliling saya. Untuk itu, dengan rasa hormat saya ucapkan terima kasih kepada :

1. Kedua orang tua serta keluarga yang tak henti-hentinya memberikan dukungan dan doa.
2. Bapak Hari Purnomo, Prof., Dr., Ir., M.T. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak R. Teduh Dirgahayu, Dr., S.T., M.Sc. selaku Kepala Program Studi Informatika Universitas Islam Indonesia.
4. Ibu Septia Rani, S.T., M.Cs. selaku dosen pembimbing skripsi dan dosen pembimbing akademik di Universitas Islam Indonesia.
5. Teman-teman Sekar Ayu yang selalu memberi dukungan dan semangat agar saya bisa menyelesaikan tugas akhir ini.
6. Rekan seperjuangan saya yang setia menemani serta mendengarkan keluh kesah saya dalam menyelesaikan tugas akhir ini.

Akhir kata, saya juga menyadari bahwa penulisan tugas akhir ini masih terdapat banyak kekurangan, maka dari itu saya mengharapkan kritik dan saran yang membangun demi kesempurnaannya. Semoga apa yang telah saya usahakan ini dapat bermanfaat bagi banyak orang *Aamiin*

Wassalaamualaykum warahmatullaahi wabarakatuh.

Yogyakarta, 15 Juli 2021



(Talitha Widya Utami)

SARI

Bug perangkat lunak merupakan suatu kesalahan atau kecacatan yang terjadi pada sebuah perangkat lunak. Pendeteksian adanya *bug* sangat penting dilakukan sehingga dapat diperbaiki oleh pengembang agar perangkat lunak berhasil mencapai keinginan atau tujuannya. Pelaporan temuan *bug* ini biasanya dilakukan melalui *message*, *email*, atau catatan biasa. Pelaporan seperti ini biasanya kurang rapi dan kurang efisien karena dapat menyulitkan pengembang dalam proses pencarian data atau riwayat pelaporan *bug*. Dalam kasus lain, pengembang juga kesulitan memahami maksud pelapor karena kurangnya *detail* pada pelaporan. Berdasarkan kondisi yang ditemui seperti ini, untuk membantu para pengguna atau *tester* perangkat lunak dalam hal pelaporan dan membantu pengembang untuk melakukan pemantauan dan mempercepat proses perbaikan *bug*, maka dibuatlah sebuah sistem informasi pencatatan *bug* agar laporan temuan *bug* lebih rapi dan terperinci. Sistem informasi pencatatan *bug* ini dibuat berbasis *website* agar sumber informasi menjadi terpusat serta dapat dengan mudah dan cepat diakses. Sistem ini dapat diakses oleh semua anggota *Center of Data Science* Universitas Islam Indonesia yang telah terdaftar sebagai pengguna sistem dengan mengakses alamat URL *cdsbugbase.site*. Sistem berbasis *web* ini dibuat secara mendetail dan sesuai dengan kebutuhan pengguna sehingga semua data atau informasi mengenai temuan *bug* seperti judul, tanggal, bukti-bukti, hingga kepada siapa tugas perbaikan *bug* tersebut ditujukan dapat dicatat atau disimpan melalui *website* ini. Data tersebut akan disimpan pada sistem basis data yang telah terhubung dengan sistem. Tampilan *website* yang sederhana dan mudah dimengerti membuat semua pengguna dapat dengan cepat dan nyaman saat mengakses situs *web* ini.

Kata kunci: pencatatan bug (*bug report*), *bug*, perangkat lunak, sistem informasi.

GLOSARIUM

Dashboard	tampilan visual dari informasi penting yang diperlukan untuk mencapai suatu tujuan atau lebih dalam suatu sistem.
Database	tempat penyimpanan data yang telah dimasukkan ke dalam sebuah sistem.
Dropdown	elemen kontrol grafis yang mirip dengan kotak daftar, memungkinkan pengguna untuk memilih salah satu nilai dari daftar tersebut.
Email	sarana dalam mengirim surat yang dilakukan dengan media internet.
Form	salah satu bentuk halaman <i>web</i> yang digunakan untuk menerima <i>input</i> dari pengguna.
Input	semua data perintah yang dimasukkan ke dalam memori komputer untuk selanjutnya diproses lebih lanjut oleh prosesor.
Link	sebuah acuan dalam dokumen hiperteks ke dokumen atau sumber lain.
Login	proses masuk ke dalam jaringan atau suatu sistem dengan memasukkan identitas akun dan verifikasi akun.
Output	hasil yang ditampilkan atau dikeluarkan.
Password	kumpulan karakter yang digunakan pengguna untuk melakukan verifikasi identitas dirinya kepada sistem keamanan pada suatu jaringan atau sistem.
Primary key	suatu nilai dalam basis data yang digunakan untuk mengidentifikasi suatu baris dalam tabel.
Screenshot	tangkapan layar atau gambar digital yang menampilkan tampilan dari suatu layar komputer.
Search bar	sebuah elemen kontrol grafis untuk mengetik kata kunci apa saja yang ingin dicari.
Testing	proses pengujian yang dilakukan oleh seorang penguji terhadap suatu sistem.
User	subjek atau pengguna yang menggunakan komputer.
Username	nama pengguna atau ID pengguna.
Web browser	perangkat lunak yang digunakan untuk menerima atau menyajikan berbagai informasi dari internet.
Website	halaman informasi yang dapat diakses banyak orang di internet.

DAFTAR ISI

HALAMAN JUDUL.....	1
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTO.....	vi
KATA PENGANTAR.....	vii
SARI.....	viii
GLOSARIUM.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian.....	3
1.7 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Studi Literatur.....	5
2.2 Sistem Informasi.....	9
2.2.1 Definisi Sistem Informasi.....	10
2.2.2 Karakteristik Sistem Informasi.....	10
2.3 <i>Website</i>	11
2.4 <i>Testing dan Bug</i>	12
2.5 <i>Hypertext Markup Language (HTML)</i>	14
2.6 PHP.....	14
2.7 <i>JavaScript</i>	14
2.8 Basis Data.....	14
2.9 XAMPP.....	15

BAB III ANALISIS DAN PERANCANGAN SISTEM.....	16
3.1 <i>User</i>	16
3.1.1 Admin.....	17
3.1.2 <i>Project Manager</i>	17
3.1.3 <i>Project Manager Assistant</i>	17
3.1.4 <i>Developer atau Programmer</i>	17
3.1.5 <i>Tester</i>	17
3.2 <i>Bug Life Cycle</i>	18
3.3 <i>Use Case Diagram</i>	19
3.4 <i>Activity Diagram</i>	21
3.4.1 <i>Activity Diagram Login</i>	22
3.4.2 <i>Activity Diagram Mengubah Profil</i>	22
3.4.3 <i>Activity Diagram Menambah Data User</i>	23
3.4.4 <i>Activity Diagram Melihat Data User</i>	25
3.4.5 <i>Activity Diagram Mengubah Data User</i>	26
3.4.6 <i>Activity Diagram Menghapus Data User</i>	27
3.4.7 <i>Activity Diagram Mencari Data User</i>	29
3.4.8 <i>Activity Diagram Menambah Data Proyek</i>	30
3.4.9 <i>Activity Diagram Melihat Data Proyek</i>	31
3.4.10 <i>Activity Diagram Mengubah Data Proyek</i>	32
3.4.11 <i>Activity Diagram Menghapus Data Proyek</i>	33
3.4.12 <i>Activity Diagram Mencari Data Proyek</i>	35
3.4.13 <i>Activity Diagram Menambah Data Bug</i>	36
3.4.14 <i>Activity Diagram Melihat Data Bug</i>	38
3.4.15 <i>Activity Diagram Mengubah Data Bug</i>	38
3.4.16 <i>Activity Diagram Menghapus Data Bug</i>	40
3.4.17 <i>Activity Diagram Mencari Data Bug</i>	41
3.4.18 <i>Activity Diagram Melihat Data Progres Bug</i>	42
3.5 Rancangan Basis Data.....	43
3.5.1 Relasi Tabel.....	43
3.5.2 Struktur Tabel.....	43
3.6 Rancangan Antar Muka.....	47
3.6.1 Halaman <i>Login</i>	47
3.6.2 Halaman Utama (<i>Dashboard</i>).....	47

3.6.3 Halaman Profil.....	48
3.6.4 Halaman Data <i>User Admin</i>	50
3.6.5 Halaman Data Proyek.....	52
3.6.6 Halaman Data <i>Bug</i>	55
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	62
4.1 Implementasi Sistem.....	62
4.1.1 Halaman <i>Login User</i>	63
4.1.2 Halaman Utama (<i>dashboard</i>).....	64
4.1.3 Halaman <i>Profile</i>	66
4.1.4 Halaman Data <i>User</i>	67
4.1.5 Halaman <i>Project</i>	69
4.1.6 Halaman <i>My task</i> atau <i>Bug</i>	73
4.2 Pengujian <i>Black Box</i>	82
4.3 <i>User Experience Testing (UX Testing)</i>	91
4.3.1 Hasil <i>User Experience Testing (UX Testing)</i>	101
BAB V KESIMPULAN DAN SARAN	103
5.1 Kesimpulan.....	103
5.2 Saran.....	103
DAFTAR PUSTAKA	104
LAMPIRAN	106

DAFTAR TABEL

Tabel 3. 1 Struktur tabel <i>bug</i>	44
Tabel 3. 2 Struktur tabel <i>progress</i>	44
Tabel 3. 3 Struktur tabel proyek.....	45
Tabel 3. 4 Struktur tabel proyek_ <i>user</i>	45
Tabel 3. 5 Struktur tabel <i>user</i>	46
Tabel 3. 6 Struktur tabel foto_ <i>bug</i>	46
Tabel 3. 7 Struktur tabel video_ <i>bug</i>	46
Tabel 4. 1 <i>User Acceptance Test</i>	82
Tabel 4. 2 <i>Test case summary Admin</i>	85
Tabel 4. 3 <i>Test case summary Project Manager</i>	86
Tabel 4. 4 <i>Test case summary Project Manager Assistant</i>	88
Tabel 4. 2 <i>Test case summary Developer</i>	89
Tabel 4. 3 <i>Test case summary Tester</i>	90



DAFTAR GAMBAR

Gambar 2. 1 Siklus pelaporan <i>bug</i> (Rex Black, 2009).....	6
Gambar 2. 2 Siklus hidup <i>bug</i> pada <i>Bugzilla</i>	7
Gambar 2. 3 Tampilan halaman pelaporan <i>bug</i> pada <i>Bugzilla</i>	8
Gambar 2. 4 Aktivitas sistem informasi.....	11
Gambar 3. 1 Struktur <i>user</i>	16
Gambar 3. 2 <i>Bug life cycle</i>	18
Gambar 3. 3 <i>Use case diagram</i>	20
Gambar 3. 4 <i>Activity diagram login</i>	22
Gambar 3. 5 <i>Activity diagram</i> mengubah profil.....	23
Gambar 3. 6 <i>Activity diagram</i> menambah data <i>user</i>	25
Gambar 3. 7 <i>Activity diagram</i> melihat data <i>user</i>	26
Gambar 3. 8 <i>Activity diagram</i> mengubah data <i>user</i>	27
Gambar 3. 9 <i>Activity diagram</i> menghapus data <i>user</i>	28
Gambar 3. 10 <i>Activity diagram</i> mencari data <i>user</i>	30
Gambar 3. 11 <i>Activity diagram</i> menambah data proyek.....	31
Gambar 3. 12 <i>Activity diagram</i> menampilkan data proyek.....	32
Gambar 3. 13 <i>Activity diagram</i> mengubah data proyek.....	33
Gambar 3. 14 <i>Activity diagram</i> menghapus data proyek.....	34
Gambar 3.15 <i>Activity diagram</i> mencari data proyek.....	36
Gambar 3. 16 <i>Activity diagram</i> menambah data <i>bug</i>	37
Gambar 3. 17 <i>Activity diagram</i> melihat data <i>bug</i>	38
Gambar 3. 18 <i>Activity diagram</i> mengubah data <i>bug</i>	39
Gambar 3. 19 <i>Activity diagram</i> menghapus data <i>bug</i>	40
Gambar 3. 20 <i>Activity diagram</i> mencari data <i>bug</i>	41
Gambar 3. 21 <i>Activity diagram</i> melihat data progress <i>bug</i>	42
Gambar 3. 22 Relasi tabel.....	43
Gambar 3. 23 Rancangan antar muka halaman <i>login</i>	47
Gambar 3. 24 Rancangan antar muka halaman <i>dashboard Admin</i>	48
Gambar 3. 25 Rancangan antar muka halaman <i>dashboard user</i> selain Admin.....	48
Gambar 3. 26 Rancangan antar muka halaman data profil.....	49
Gambar 3. 27 Rancangan antar muka halaman ubah profil.....	49
Gambar 3. 28 Rancangan antar muka halaman data <i>user</i>	50

Gambar 3. 29 Rancangan antar muka halaman tambah <i>user</i>	51
Gambar 3. 30 Rancangan antar muka halaman ubah data <i>user</i>	51
Gambar 3. 32 Rancangan antar muka halaman tambah data projek.....	53
Gambar 3. 33 Rancangan antar muka halaman ubah data projek.....	53
Gambar 3. 34 Rancangan antar muka halaman data projek Admin.....	54
Gambar 3. 35 Rancangan antar muka halaman data projek <i>Project Manager Assistant, Tester, dan Developer</i>	55
Gambar 3. 36 Rancangan tabel <i>bug</i>	56
Gambar 3. 37 Rancangan antar muka halaman data <i>bug</i> Admin.....	56
Gambar 3. 38 Rancangan antar muka halaman data <i>bug Project Manager, Project Manager Assistant dan Tester</i>	57
Gambar 3. 39 Rancangan antar muka halaman tambah data <i>bug Project Manager, Project Manager Assistant dan Tester</i>	58
Gambar 3. 40 Rancangan antar muka halaman ubah data <i>bug Project Manager, Project Manager Assistant dan Tester</i>	59
Gambar 3. 41 Rancangan antar muka halaman data <i>progress bug</i>	59
Gambar 3. 42 Rancangan antar muka halaman data <i>bug Developer</i>	60
Gambar 3. 43 Rancangan antar muka halaman ubah data <i>bug Developer</i>	61
Gambar 4. 1 Bagan alur sistem.....	63
Gambar 4. 2 Halaman <i>login user</i>	64
Gambar 4. 3 Halaman utama (<i>dashboard</i>) <i>Project Manager</i>	64
Gambar 4. 4 Halaman utama (<i>dashboard</i>) Admin.....	65
Gambar 4. 5 Halaman data profil <i>Project Manager</i>	66
Gambar 4. 6 Halaman ubah data profil <i>Project Manager</i>	67
Gambar 4. 7 Halaman data semua <i>user</i>	68
Gambar 4. 8 Halaman tambah data <i>user</i>	69
Gambar 4. 9 Halaman data projek <i>Project Manager</i>	70
Gambar 4. 10 Halaman tambah data projek <i>Project Manager</i>	71
Gambar 4. 11 Halaman data projek <i>Project Manager Assistant</i>	72
Gambar 4. 12 Halaman data projek Admin.....	73
Gambar 4. 13 Halaman data <i>bug Project Manager</i>	74
Gambar 4. 14 Halaman tambah data <i>bug Project Manager</i>	75
Gambar 4. 15 Halaman ubah data <i>bug Project Manager (a)</i>	76
Gambar 4. 16 Halaman ubah data <i>bug Project Manager (b)</i>	76

Gambar 4. 17 Halaman ubah status <i>bug Project Manager</i>	77
Gambar 4. 18 Halaman data progress <i>bug Project Manager</i>	77
Gambar 4. 19 Halaman bukti foto <i>bug</i>	78
Gambar 4. 20 Halaman bukti <i>video bug</i>	78
Gambar 4. 21 Halaman data <i>bug Tester</i>	79
Gambar 4. 22 Halaman data <i>bug Developer</i>	80
Gambar 4. 23 Halaman data <i>bug Admin</i>	81
Gambar 4. 24 Skenario <i>UX testing Admin</i>	92
Gambar 4. 25 Skenario <i>UX testing Project Manager</i>	93
Gambar 4. 26 Skenario <i>UX testing Tester</i>	93
Gambar 4. 27 Skenario <i>UX testing Developer</i>	94
Gambar 4. 28 Hasil penilaian pemilihan <i>font</i> yang digunakan.....	95
Gambar 4. 29 Hasil penilaian pilihan menu.....	95
Gambar 4. 30 Hasil penilaian penggunaan <i>web</i>	96
Gambar 4. 31 Kesulitan yang ditemui pengguna.....	96
Gambar 4. 32 Bagian <i>web</i> yang disukai penguji.....	97
Gambar 4. 33 Pendapat penguji mengenai UI.....	98
Gambar 4. 34 Pendapat penguji mengenai cara penyajian informasi.....	99
Gambar 4. 35 Kemungkinan penguji mengunjungi kembali <i>web CDS Bug Base</i>	100
Gambar 4. 36 Tingkat kemudahan penggunaan <i>web CDS Bug Base</i>	100
Gambar 4. 37 Tingkat kemudahan menyelesaikan skenario kasus.....	101

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perangkat lunak merupakan perintah program komputer yang akan berfungsi dan bekerja sesuai keinginan *user* apabila dieksekusi. Perangkat lunak sebagai pengatur aktivitas kerja komputer bertugas untuk menjembatani interaksi *user* dengan komputer yang hanya memahami bahasa mesin. Keberhasilan suatu perangkat lunak dapat diukur dari rencana yang telah dibuat, seperti kesesuaian kerja dan tujuan yang diinginkan. Dalam fase pemeliharaan sistem, pengembang akan menerima berbagai laporan seperti gagalnya sistem, ketidaksesuaian fungsi, atau kerusakan lainnya. Salah satu kerusakan yang sering dijumpai adalah *bug*.

Bug perangkat lunak merupakan suatu kesalahan atau kecacatan yang terjadi pada sebuah perangkat lunak. *Bug* adalah suatu hal yang seharusnya tidak terjadi pada perangkat lunak atau seharusnya tidak dilakukan oleh perangkat lunak. Pendeteksian adanya *bug* sangat penting dilakukan sehingga dapat diperbaiki oleh pengembang agar perangkat lunak berhasil mencapai keinginan atau tujuannya. Untuk itu, diperlukan suatu sistem informasi yang bertujuan untuk mencatat semua *bug* yang ditemukan secara terperinci. Sistem informasi merupakan suatu sistem pada sebuah organisasi yang mempertemukan kebutuhan transaksi harian yang berperan sebagai pendukung fungsi manajerial dalam kegiatan strategis organisasi untuk menyediakan laporan-laporan tertentu yang dibutuhkan oleh pihak lain. Adanya sistem informasi pencatatan *bug* ini akan mempermudah pengembang untuk mengetahui kesalahan atau kerusakan apa saja yang perlu mereka perbaiki demi keberhasilan suatu perangkat lunak.

Center of Data Science (CDS) UII merupakan sebuah pusat studi mengenai sains data dibawah naungan Jurusan Teknik Informatika Universitas Islam Indonesia. Tim CDS UII juga mengerjakan proyek-proyek seperti pembangunan sistem dan lain sebagainya. Tak jarang mereka juga kerap menemukan *bug* pada proyek-proyek tersebut. Umumnya, *bug* akan ditemukan oleh *tester* atau pengguna akhir perangkat lunak, lalu mereka akan melaporkannya pada pengembang. Tim CDS UII selama ini menggunakan media seperti *message*, *email*, atau catatan biasa untuk melaporkan adanya penemuan *bug*. Selanjutnya, barulah dilakukan pengecekan dan tindak lanjut pada temuan *bug* tersebut oleh tim pengembang atau anggota

tim teknis lainnya yang dipilih untuk mengerjakan *bug*. Pelaporan *bug* menggunakan media-media seperti yang telah disebutkan akan membuat pencatatan temuan *bug* kurang rapi dan kurang efisien karena dapat menyulitkan pengembang dalam proses pencarian data atau riwayat pelaporan *bug*. Dalam kasus lain, pengembang juga kesulitan memahami maksud pelapor karena kurangnya *detail* pada pelaporan.

Berdasarkan uraian di atas, untuk membantu para pengguna perangkat lunak dalam hal pelaporan dan membantu pengembang untuk melakukan pemantauan *bug* maka penulis akan membuat sebuah sistem informasi pencatatan *bug* berbasis *web* lebih yang rapi dan terperinci. Laporan *bug* yang rapi dan terperinci akan memberikan gambaran umum yang lebih jelas, seperti tanggal penemuan sampai penyelesaian, efek *bug* pada sistem, status prioritas perbaikan, *screenshot* atau *video* saat muncul *bug*, status progres perbaikan *bug* dari awal pelaporan hingga penyelesaian akhir, dan kepada siapa perbaikan *bug* ditugaskan.

1.2 Rumusan Masalah

Berdasarkan dari uraian pada latar belakang, maka dapat dirumuskan beberapa permasalahan sebagai berikut:

- a. Bagaimana membangun sistem informasi pencatatan *bug* berbasis *web* yang lebih rapi dan *detail*?
- b. Bagaimana membuat *web* pencatatan *bug* yang dinamis dan interaktif?

1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini tidak meluas, maka pada pembuatan sistem informasi pencatatan *bug* ini diberikan beberapa batasan masalah sebagai berikut:

- a. *User* dibatasi hanya sebanyak 5 meliputi admin, *project manager*, manajer, pengembang atau *programmer*, dan *tester*.
- b. Sistem informasi yang dibangun berbasis *web* dengan tampilan yang rapi, dinamis, dan interaktif.
- c. Menggunakan bahasa pemrograman *PHP*, *HTML*, *JavaScript* dan menggunakan *MySQL* sebagai basis datanya.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat sistem informasi pencatatan *bug* yang berbasis *web* sebagai wadah untuk pelaporan *bug*, penyimpanan data *bug*, dan pemantauan status *bug* yang ditemukan pada suatu perangkat lunak.

1.5 Manfaat Penelitian

Adapun manfaat dari dilakukannya penelitian ini ialah sebagai berikut:

- a. Memudahkan *user* untuk melakukan pelaporan adanya *bug*.
- b. Pencatatan *bug* menjadi lebih rapi dan terstruktur.
- c. Pencatatan *bug* menjadi lebih rinci sehingga memudahkan pengembang atau *programmer* untuk mengerti pokok permasalahannya.
- d. Mempermudah *user* dalam pencarian *history bug* yang sudah pernah terjadi pada suatu proyek.

1.6 Metodologi Penelitian

Adapun metodologi dari penelitian ini ialah sebagai berikut:

- a. Studi Pustaka
- b. Identifikasi Masalah
- c. Perancangan Sistem
- d. Implementasi
- e. Pengujian

1.7 Sistematika Penulisan

Dalam penulisan laporan penelitian ini, penulis akan membaginya menjadi enam bagian pokok. Setiap bagian akan memiliki sub bagian yang saling berkaitan. Berikut enam bagian pokok tersebut:

- a. Pendahuluan

Pada bagian ini penulis menguraikan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, serta manfaat dari penelitian yang dilakukan.

- b. Tinjauan Pustaka

Pada bagian ini berisikan tentang teori-teori terkait dengan pencatatan *bug* yang didapatkan dari membaca buku-buku serta jurnal-jurnal penelitian serupa.

c. Analisis dan Pemodelan Sistem

Analisis dan pemodelan sistem ini berisikan pembahasan mengenai sistem yang akan dibuat, seperti aktor dan strukturnya, penggunaan aplikasi, dan bahasa pemrograman yang akan digunakan.

d. Perancangan Sistem

Rancangan sistem dibuat berdasarkan analisis masalah yang didapat. Rancangan sistem ini berisikan pembahasan mengenai hal-hal yang dibutuhkan terkait sistem, seperti aktivitas apa saja yang terjadi pada sistem, diagram sistem, struktur sistem, basisdata, hingga antarmuka.

e. Implementasi dan Pengujian Sistem

Setelah rancangan sudah sesuai dan disepakati kemudian dilakukan implementasi sistem. Pada bagian ini berisikan implementasi sistem, spesifikasi sistem, instalasi aplikasi yang digunakan, serta konfigurasi sistem. Tujuan dari pengujian ini ialah untuk mengetahui apakah fungsi sistem sudah sesuai dengan yang diinginkan pengguna. Apabila terjadi kesalahan dan atau belum sesuai dengan fungsi dan tujuan yang diinginkan, maka akan dilakukan implementasi ulang.

f. Kesimpulan dan saran

Bagian ini berisikan kesimpulan yang diperoleh dari hasil penelitian dan saran yang bersifat membangun agar sistem bisa semakin sempurna.

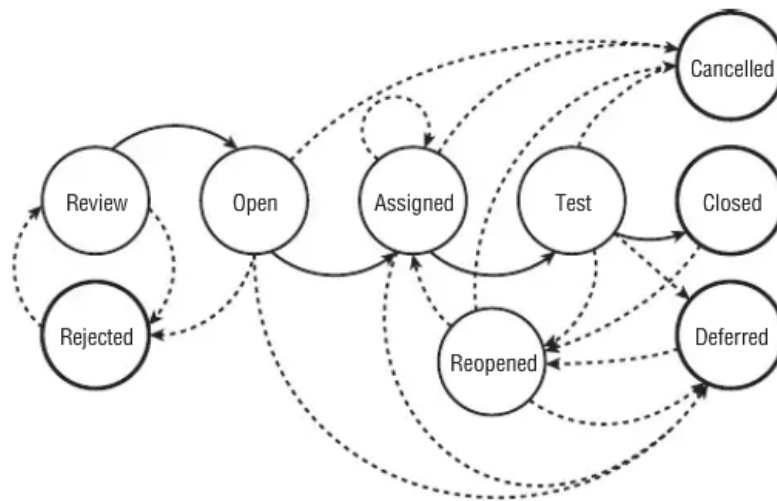
BAB II TINJAUAN PUSTAKA

2.1 Studi Literatur

Pelacakan *bug* merupakan hal yang penting dilakukan dalam rangka perbaikan atau peningkatan performa atau kualitas suatu sistem yang sedang dikembangkan atau digunakan. Sistem pencatatan *bug* merupakan alat yang tepat untuk memfasilitasi proses perbaikan suatu *bug* pada sistem yang lebih cepat, mudah, dan efektif. Menurut Kolluri, A. B., dkk (2012), sistem pelacakan *bug* yang efektif memiliki empat orientasi yang saling berkesinambungan, yaitu sebagai berikut.

- a. *Tool oriented*, yaitu menyediakan media yang bisa menyediakan suatu media untuk mengumpulkan temuan *bugs* beserta detailnya sehingga informasi mengenai suatu *bug* akan lebih lengkap. Orientasi ini akan mengacu pada orientasi berikutnya yaitu *information oriented*.
- b. *Information oriented*, yaitu pengumpulan informasi yang selengkap-lengkapnyanya dan sedetail mungkin akan sangat mempermudah dan mempercepat *developer* atau pihak yang ditugaskan untuk menangani *bug* tersebut untuk bisa memperbaikinya. Misalnya mencantumkan deskripsi *bug* dengan detail, lengkap beserta media pendukung lainnya seperti foto, *screenshot*, atau *video*.
- c. *Process oriented*, yaitu semua alur aktivitas atau langkah-langkah yang dilakukan dalam menangani suatu *bug*, mulai dari pengumpulan hingga penyelesaiannya.
- d. *User oriented*, yaitu petunjuk yang ditujukan untuk *user* yang melaporkan adanya *bug* dan *user* yang ditugaskan untuk menanganinya sehingga akan memudahkan mereka dalam menggunakan media pengumpulan atau pencatatan *bug* seperti yang disebutkan pada *tool oriented*.

Bug memiliki siklus hidup yang terdiri dari beberapa tahapan dari mulai *bug* ditemukan hingga tahap akhir tindak lanjut dalam mengatasi *bug* tersebut. Menurut Rex Black (2009), berikut ini ialah siklus dari pelaporan sebuah *bug*.

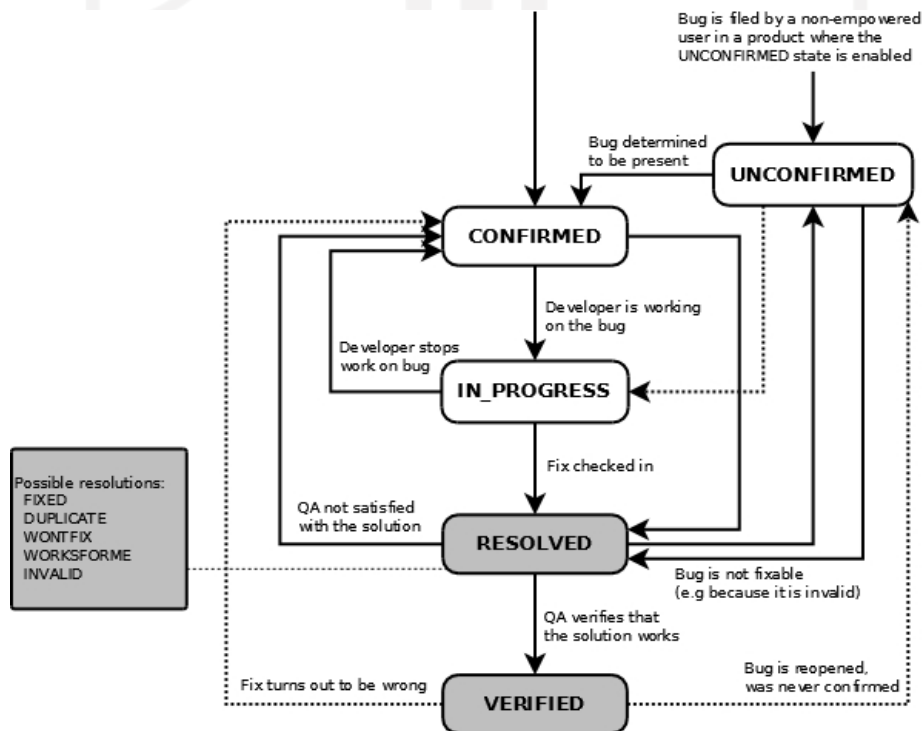


Gambar 2. 1 Siklus pelaporan *bug* (Rex Black, 2009)

- a. *Review*, yaitu peninjauan yang dilakukan oleh tim *tester* saat menemukan sebuah *bug* untuk memastikan *bug* tersebut memang benar adanya dan perlu ditindaklanjuti dengan melaporkannya.
- b. *Rejected*, yaitu saat laporan *bug* yang sudah ditinjau dan dianggap perlu dicek lebih lanjut memerlukan dites atau *research* ulang untuk nantinya dilaporkan kembali, atau laporan *bug* tersebut ditolak karena kurangnya informasi mengenai *bug* tersebut.
- c. *Open*, yaitu tahap atau status yang menyatakan bahwa laporan *bug* sudah ditinjau dan benar adanya serta butuh tindak lanjut.
- d. *Assigned*, yaitu penugasan yang dilakukan oleh tim *tester* kepada bagian *development*.
- e. *Test*, pada tahap ini akan ada dua tes yaitu tes konfirmasi, apakah *bug* yang diperbaiki sudah bisa berjalan dengan baik sesuai dengan fungsi yang diharapkan, dan tes regresi yang melihat apakah setelah diperbaiki, *bug* tersebut menimbulkan masalah lain atau memberikan efek lain terhadap sistem.
- f. *Reopened*, setelah menjalani tahap *test*, akan dilihat lagi apakah perbaikan *bug* tersebut sudah berhasil memenuhi tes konfirmasi dan tes regresi. Jika hanya memenuhi salah satu atau bahkan tidak memenuhi sama sekali, akan dibuat kembali laporan *bug*.
- g. *Cancelled*, yaitu kondisi saat tim menilai bahwa yang dilaporkan tidak *real* atau *false positive*, maka tim akan membatalkan laporan tersebut.

- h. *Closed*, yaitu kondisi saat *bug* sudah berhasil diperbaiki dan sudah memenuhi tahap tes, maka laporan ditutup.
- i. *Deffered*, yaitu saat tim menilai bahwa *bug* yang ditemukan memang benar adanya, akan tetapi kurang begitu penting atau berpengaruh dan dianggap bukan sebuah prioritas untuk diperbaiki, sehingga memungkinkan untuk ditunda perbaikannya.

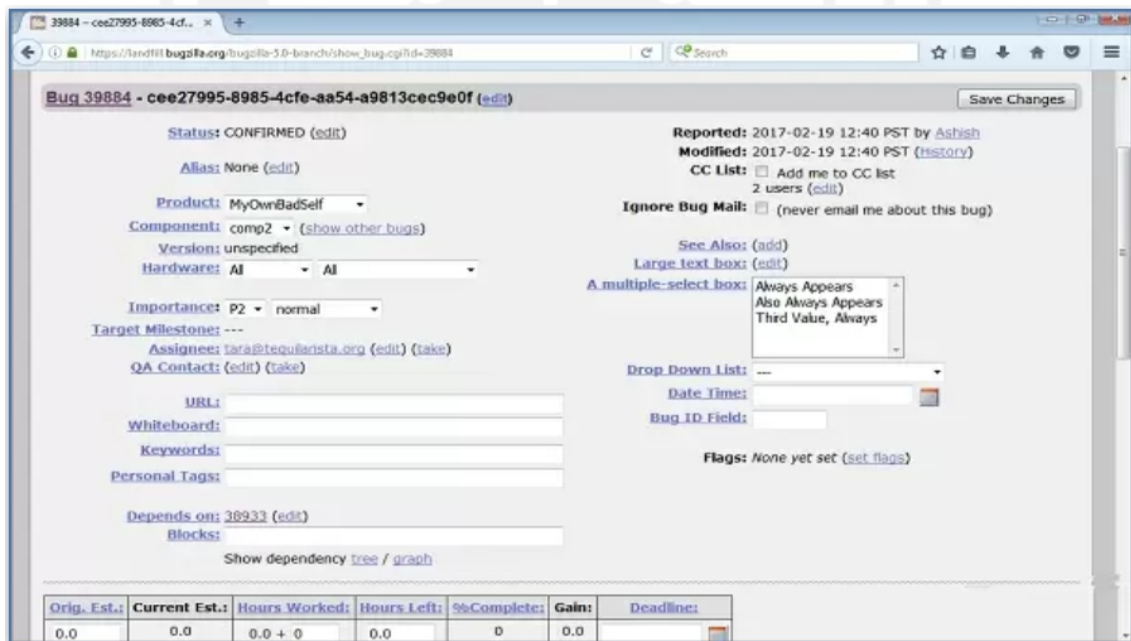
Salah satu contoh media pencatatan *bug* berbasis *web* yang sudah cukup terkenal ialah *Bugzilla* yang dibangun oleh *Mozilla Foundation*. Berdasarkan penjelasan pada *website* resmi *Bugzilla*, awalnya *Bugzilla* ini dibuat hanya untuk proyek-proyek *Mozilla* saja, namun hingga saat ini *Bugzilla* juga telah digunakan oleh proyek-proyek besar seperti *Gnome*, *Linux Kernel*, *Apache*, *Open Office*, dan masih banyak lagi proyek, perusahaan, dan organisasi lainnya. Seperti sistem informasi lainnya, sistem utama *Bugzilla* berfungsi untuk memasukkan dan mengakses data. Pada sistem *Bugzilla* ini, pengguna dapat memasukkan *bug* yang baru dan mencari atau mengedit *bug* yang sudah ada. Untuk menginstal dan menggunakan *Bugzilla*, persyaratan utamanya adalah menginstal *Perl*. *Bugzilla* dapat diinstal pada *platform* apa saja asalkan *Perl* dapat diinstal, seperti pada *Linux* sebagai rekomendasi utama mereka, *Windows*, dan *Mac OS X*. Gambar 2.2 merupakan representasi grafis dari *default* siklus hidup sebuah *bug* pada *Bugzilla*.



Gambar 2. 2 Siklus hidup *bug* pada *Bugzilla* (BMO Documentation)

- Confirmed* atau *unconfirmed*, yaitu status awal *bug* dalam *Bugzilla*. Pada tahap ini, data sebuah *bug* sudah tidak bisa dihapus atau diperbarui lagi.
- In Progress*, yaitu status *bug* dalam *Bugzilla* dimana pengembang masih dalam proses pengerjaan tugas yang tercantum dalam deskripsi di *Bugzilla*.
- Resolved*, yaitu status *bug* dalam *Bugzilla* yang biasanya diubah oleh pihak *tester* atau *quality assurance* apabila masalah yang ada sudah dites dan tidak lagi menemui kendala.
- Verified*, yaitu status *bug* dalam *Bugzilla* yang diubah oleh pihak *tester* apabila sudah tidak menemui kesalahan selama fase *testing*.

Gambar 2.3 berikut ini merupakan salah satu contoh tampilan dari halaman pelaporan *bug* pada *Bugzilla*.



Gambar 2. 3 Tampilan halaman pelaporan *bug* pada *Bugzilla*

- Status, yaitu keterangan status dari sebuah *bug*.
- Alias, yaitu nama unik suatu *bug* dan dapat digunakan sebagai pengganti nomor *bug*.
- Product* dan *component*, *bug* dibagi berdasarkan produk dan komponen, produk yang memiliki satu atau lebih komponen di dalamnya.
- Version*, berisi nomor atau nama versi produk yang dirilis.
- Hardware*, menunjukkan lingkungan komputasi tempat *bug* ditemukan.

- f. *Importance*, berisi mengenai tingkat prioritas dan *severity*.
- g. *Target milestone*, yaitu versi masa depan dimana *bug* akan diperbaiki.
- h. *Assignee*, yaitu orang yang bertanggung jawab untuk memperbaiki *bug*.
- i. *QA contact*, yaitu orang yang bertanggung jawab untuk jaminan kualitas pada *bug*.
- j. *URL*, berisikan URL yang terkait dengan *bug*, jika ada.
- k. *Whiteboard*, yaitu *text area* untuk menambahkan catatan pendek dan *tag* ke *bug*.
- l. *Keywords*, berguna untuk menandai dan mengkategorikan *bug*.
- m. *Personal tags*, bersifat pribadi dan hanya dapat dilihat dan diedit oleh pembuatnya. Digunakan untuk menandai dan melacak kumpulan *bug* tertentu secara pribadi dan menggunakan sistem klasifikasi sendiri.
- n. *Depends on*, kondisi dimana suatu *bug* tidak dapat diperbaiki kecuali *bug* lain diperbaiki (*depends on*).
- o. *Blocks*, kondisi dimana suatu *bug* menghentikan *bug* lain yang sedang diperbaiki (memblokir).
- p. *Reported*, yaitu orang yang mengajukan *bug*, serta tanggal dan waktu mereka melakukannya.
- q. *Modified*, yaitu tanggal dan waktu *bug* terakhir diubah.
- r. *CC list*, yaitu daftar orang yang menerima *email* saat *bug* mengalami perubahan.
- s. *Ignore bug mail*, setelah ini jika Anda tidak ingin mendapatkan *bug mail* dari *bug* ini lagi.
- t. *See also*, *bug* di *Bugzilla* ini, *Bugzilla* lain, atau pelacak *bug* lainnya, yang terkait dengan yang satu ini.
- u. *Flags*, semacam status yang dapat diatur pada *bug* atau lampiran untuk menunjukkan bahwa *bug* atau lampiran berada dalam keadaan tertentu.

2.2 Sistem Informasi

Menurut Romney dan Steinbart (2015), sistem merupakan suatu kesatuan dari dua atau lebih komponen yang saling berkaitan untuk mencapai tujuan tertentu. Informasi menurut Agus Mulyanto (2009) ialah suatu data yang mengalami proses pengolahan sehingga menjadi lebih berguna dan berarti bagi penerimanya, dan data merupakan sumber informasi yang menggambarkan kejadian yang nyata.

2.2.1 Definisi Sistem Informasi

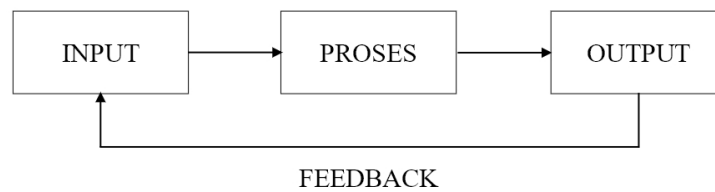
Pengertian sistem informasi menurut Kadir (2014) ialah suatu rangkaian pengelompokan data formal, kemudian data diproses menjadi sebuah informasi yang akan didistribusikan pada pengguna. Berbeda dengan Krismiaji (2005), menurutnya sistem informasi merupakan beberapa cara yang digunakan untuk mengumpulkan, memasukkan, mengolah serta menyimpan, dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan suatu informasi sehingga organisasi dapat mencapai tujuannya. Sistem Informasi adalah sekumpulan data yang berintegritas dan bersifat melengkapi satu sama lainnya untuk menghasilkan *output* yang dapat digunakan untuk memecahkan suatu permasalahan atau pengambilan keputusan.

2.2.2 Karakteristik Sistem Informasi

Terdapat empat karakteristik pokok sistem menurut Ardana dan Lukman (2016) yaitu sebagai berikut :

- a. Satu-kesatuan : satu-kesatuan organisasi.
- b. Bagian-bagian : ada manajemen, karyawan, pemangku kepentingan (*stakeholders*) lainnya, gedung kantor, sub-sistem komputer (perangkat keras, perangkat lunak, perangkat jaringan, sumber daya manusia, basis data, dan informasi)
- c. Terjalin erat : tercermin dalam hubungan, interaksi, prosedur kerjasama antar manajemen, karyawan dan subsistem komputer yang diatur dalam bentuk berbagai prosedur dan instruksi kerja.
- d. Mencapai tujuan : menghasilkan informasi yang berkualitas bagi manajemen dan pemangku kepentingan lainnya.

Pada sistem informasi, terdapat tiga aktivitas yang dibutuhkan yaitu masukan (*input*), proses, dan keluaran (*output*). *Input*, pengumpulan data mentah yang berasal dari dalam maupun luar organisasi. Proses, perubahan data *input* mentah menjadi sesuatu yang lebih berguna. *Output*, pengiriman informasi yang sudah melalui pemrosesan kepada pengguna atau kepada aktivitas lain yang akan menggunakan informasi tersebut. Aktivitas-aktivitas ini divisualkan seperti pada Gambar 2.2 berikut.



Gambar 2. 4 Aktivitas sistem informasi

2.3 Website

Pengertian *web* secara fisik menurut Oetomo (2007) ialah kumpulan dari komputer pribadi, *web browser*, koneksi ke ISP, komputer *server*, *router*, dan *switch* yang digunakan sebagai media untuk menyebarkan informasi. Menurut Laudon K. C. dan Laudon J. P. (2007), *world wide web* atau yang biasa disebut dengan *web* adalah suatu sistem dengan suatu standar tertentu yang digunakan untuk menyimpan, mencari, mengelola, serta menampilkan informasi dengan menggunakan arsitektur *client server*. Menurut Supriyanto (2007), ada tiga mekanisme kerja *website* yaitu sebagai berikut.

- a. Protokol standar, yaitu *Hypertext Transfer Protocol* (HTTP)
- b. *Address*, yaitu alamat web atau *Uniform Resource Locator* (URL)
- c. *Hypertext Markup Language* (HTML), bahasa yang digunakan untuk membuat suatu dokumen agar dapat diakses dengan *website*.

Menurut Ali Zaki (2009), sebuah *web page* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website*. Contoh *website* statis adalah berisi profil perusahaan, sedangkan *website* dinamis adalah seperti *Friendster*, *Facebook*, *Zalora*, dan masih ada banyak lagi contoh lainnya. Dalam sisi pengembangannya, *website* statis hanya bisa diupdate oleh pemiliknya saja, sedangkan *website* dinamis bisa di-*update* oleh pengguna maupun pemilik.

2.4 Testing dan Bug

Testing atau pengujian pada perangkat lunak merupakan salah satu tahap yang penting dalam penentuan keberhasilan dan menjaga kualitas suatu perangkat lunak. Galin (2004) mengartikan *software testing* atau pengujian pada perangkat lunak sebagai suatu proses pengujian yang dilakukan secara formal dengan menjalankan prosedur dan kasus tertentu pada perangkat lunak tersebut di dalam komputer. Galin (2004) juga menyatakan bahwa proses pengujian pada suatu perangkat lunak juga mempunyai peranan yang erat kaitannya dengan kualitas perangkat lunak tersebut, sehingga menurutnya proses ini sangat penting dalam siklus pengembangan perangkat lunak. Proses pengujian ini juga bertujuan untuk menemukan kesalahan atau *bug* yang ada pada suatu perangkat lunak. Perangkat Lunak sendiri dapat diuji dengan dua cara yaitu *White Box Testing* dan *Black Box Testing*. *White Box Testing* ialah pengujian yang dilakukan dengan mengenakan data uji untuk menguji semua elemen program atau struktur internal berupa *source code* dari suatu perangkat lunak, Pengujian ini adalah pengujian yang paling umum digunakan. *Black Box Testing* ialah salah satu metode pengujian yang dilakukan dengan cara mengeksekusi data uji tanpa memperhatikan struktur logika internal (*source code*) perangkat lunak dan mengecek apakah fungsional perangkat lunak bekerja dengan baik dan sesuai tujuan. Teknik pengujian yang akan digunakan oleh peneliti pada pembangunan sistem kali ini ialah metode *Black Box Testing*. Pengujian *Black Box* berusaha menemukan kesalahan dalam lima kategori yaitu sebagai berikut.

- a. Fungsi-fungsi yang tidak benar atau hilang.
- b. Kesalahan interface.
- c. Kesalahan dalam struktur data atau akses database eksternal.
- d. Kesalahan kinerja.
- e. Inisialisasi dan kesalahan terminasi

Black Box Testing memiliki cabang pengujian lain didalamnya, salah satunya yaitu *User Experience (UX) Testing*, dimana metode ini juga akan digunakan setelah pembangunan sistem ini. *UX Testing* ini lebih berkaitan dengan rasa dan pengalaman yang dirasakan dan dialami oleh pengguna dengan produk yang digunakan. Perasaan yang dapat pengguna berikan sebagai hasil pengujian adalah memuaskan, nyaman, menghibur, membantu, indah, kreatif, stimulasi kognitif, menyenangkan, lucu, mengejutkan, menantang, membosankan, membuat frustrasi, atau mengganggu. Berikut ini merupakan enam tujuan dari *UX Testing*.

- a. Mengukur kepuasan pengguna dengan antarmuka.

- b. Mencari tahu apakah pengguna dapat menggunakan antarmuka dan mencapai tujuan mereka.
- c. Menemukan kemungkinan kelemahan produk.
- d. Menemukan apa yang paling disukai pengguna tentang situs *web*.
- e. Menerima *feedback* yang berguna sehingga kualitas produk dapat ditingkatkan.
- f. Meningkatkan retensi pelanggan

Berikut ini merupakan langkah-langkah yang dapat dilakukan untuk melaksanakan UX

Testing.

- a. Penguji dipilih.
- b. Penguji kemudian diberi tugas (*test case*) untuk diselesaikan.
- c. Penguji diminta untuk mengomentari tindakan mereka.
- d. Penguji menjawab serangkaian pertanyaan mengenai pengujian yang dilakukan.
- e. Pengamat dengan hati-hati memperhatikan gerakan mereka, mendengarkan komentar dan jawaban mereka sambil mencatat.

Bug adalah sesuatu yang seharusnya tidak dilakukan oleh perangkat lunak atau perangkat lunak yang tidak melakukan seperti yang seharusnya dilakukan. Setiap *bug* memiliki atribut data yang digunakan untuk identifikasi, seperti nama *bug*, tipe *bug*, tanggal ditemukan, hingga cara bagaimana *bug* tersebut bisa muncul dan ditemukan. *Bug* mempunyai empat tipe, yaitu :

- a. *Trivia*, yaitu tipe *bug* dengan tingkatan kesalahan yang paling rendah seperti kesalahan pada penulisan, label, warna, dan segala sesuatu yang berkaitan dengan tampilan atau *user interface*.
- b. *Minor*, yaitu tipe *bug* dengan tingkat urgensi menengah (*medium*). *Bug minor* ini merupakan *bug* yang tidak sesuai dengan ekspektasi namun tidak berpengaruh besar pada fungsionalitasnya, misalnya *user* melakukan penyimpanan data lalu sistem menampilkan notifikasi berhasil tersimpan, namun data tersebut tidak tampil pada sistem.
- c. *Major*, yaitu tipe *bug* yang memiliki tingkat urgensi menengah keatas. *Bug major* ini memiliki pengaruh yang cukup besar terhadap fungsionalitas atau jalannya sistem, misalnya kegagalan pada saat menambahkan data baru pada sistem sehingga memunculkan pesan *error* yang dapat menimbulkan kepanikan *user*.

- d. *Critical*, yaitu tipe *bug* yang memiliki tingkat urgensi yang tertinggi, misalnya sistem yang tidak dapat diakses oleh *user* sehingga sistem benar-benar tidak dapat berjalan sama sekali seperti semestinya.

2.5 *Hypertext Markup Language (HTML)*

Pengertian *Hypertext Markup Language (HTML)* menurut Madcoms (2008) ialah teks bahasa kode yang digunakan untuk membuat suatu halaman yang dapat diakses dengan *web* dan dikenal dengan ekstensi *.htm* atau *.html*. HTML mempunyai sifat yang fleksibel karena terdapat beberapa kode seperti Java Script, VBScript, bahasa pemrograman seperti PHP bisa disisipkan di dalamnya. Supriyanto (2007) mengatakan bahwa terdapat dua fungsi HTML, yaitu sebagai berikut.

- a. Membentuk tata letak dokumen (jenis huruf, gambar, dan komponel lainnya)
- b. Menentukan hubungan dengan dokumen yang lainnya.

2.6 PHP

Pada penelitian ini, penulis menggunakan bahasa pemrograman PHP. PHP *Hypertext Preprocessor* ini merupakan bahasa pemrograman yang biasa digunakan untuk membangun suatu *website* yang dinamis. Menurut Kadir (2008), PHP dapat digunakan secara *command line* yang maksudnya adalah tetap bisa dioperasikan tanpa perlu *web server* dan *browser*.

2.7 *JavaScript*

Pengertian *JavaScript* menurut Zaki (2008) ialah suatu skrip pada program yang berbasis *client* dan dieksekusi oleh *browser* dengan tujuan untuk membuat sebuah laman *web* dapat melakukan perintah tambahan saat skrip HTML biasa tidak dapat melakukannya. Menurut Pohan (2012), *JavaScript* merupakan bahasa C++ yang telah dimodifikasi sehingga penulisannya menjadi lebih sederhana.

2.8 Basis Data

Basis data atau *database* menurut Hesnanda (2017) ialah suatu wadah untuk menampung data yang ada pada suatu sistem. Menurut Rosa dan Salahuddin (2013), basis data merupakan suatu media penyimpanan data sehingga data tersebut dapat dengan mudah dan cepat untuk diakses. Berbeda dengan Hirin dan Virgi (2011), menurut mereka basis data

ialah sekumpulan informasi atau data yang sistematis sehingga bisa digunakan oleh suatu program komputer dalam mendapatkan atau mengakses informasi atau data tersebut.

MySQL merupakan suatu jenis basis data yang bersifat *open source* yang dilengkapi dengan *source code* yang dapat dijalankan langsung pada sebuah sistem operasi. Menurut Madcoms (2008), ada empat instruksi dasar yang dimiliki oleh MySQL, yaitu *select*, *insert*, *update*, dan *delete*. *Select* untuk menampilkan data, *insert* untuk memasukkan data, *update* untuk mengubah data, dan *delete* untuk menghapus data pada basis data.

2.9 XAMPP

XAMPP merupakan sebuah paket perangkat lunak yang dikembangkan oleh *Apache Friends*. XAMPP ini memiliki fungsi sebagai *server* lokal untuk menunjang data *website* yang sedang dibangun atau dikembangkan. Selain itu XAMPP juga dapat digunakan untuk menguji sistem atau fitur pada sistem berbasis *website* tanpa perlu terhubung dengan internet. XAMPP merupakan akronim dari :

X : Cross platform, maksudnya dapat digunakan pada berbagai sistem operasi.

A : Apache, yaitu web server yang dapat digunakan oleh banyak orang secara cuma-cuma (*open source*).

M : MySQL atau MariaDb, yaitu *server* basis data yang berperan sebagai pengolah data.

P : PHP, yaitu bahasa pemrograman yang biasanya digunakan untuk pembuatan *website* dinamis.

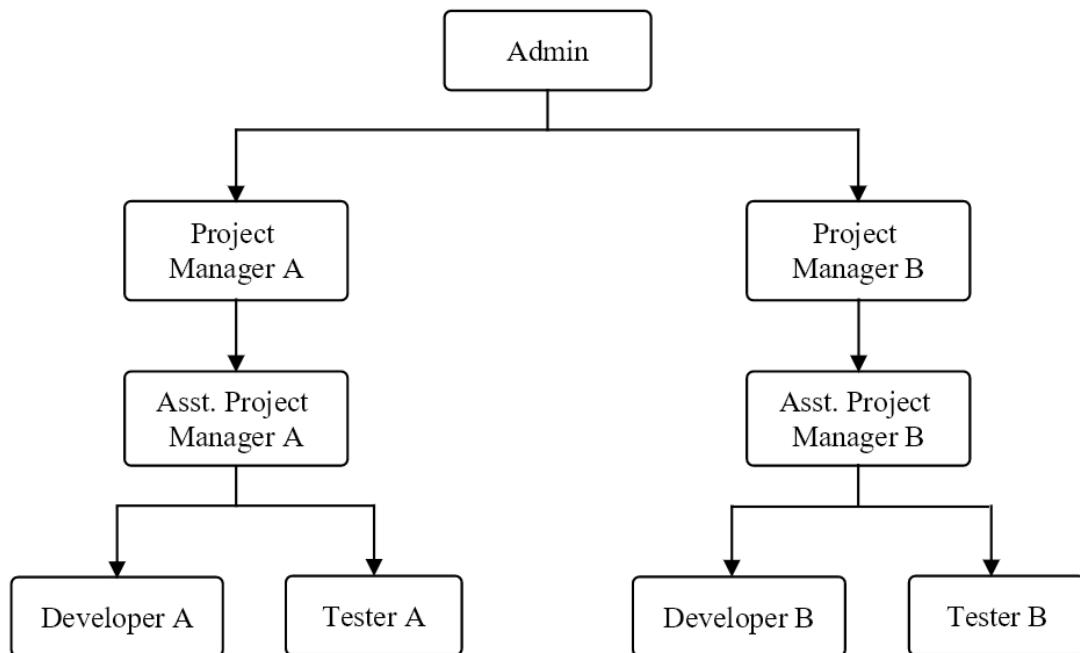
P : *Perl*, yaitu bahasa pemrograman yang dapat digunakan pada banyak sistem operasi dan bersifat lebih fleksibel.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 User

Penelitian ini diawali dengan melakukan analisis siapa saja yang akan bertindak sebagai *user* atau pengguna serta menentukan perannya masing-masing. Terdapat lima jenis *user* yang akan menggunakan sistem ini yaitu Admin, *Project Manager*, *Project Manager Assistant*, *Developer*, dan *Tester*. Semua pengguna Sistem Informasi Pencatatan Bug ini merupakan anggota dari tim *Center of Data Science* UII. Admin menempati posisi teratas pada struktur *user* ini. Kemudian Admin membawahi *Project Manager* untuk setiap proyek yang ada. Tiap *Project Manager* membawahi *Project Manager Assistant* untuk masing-masing proyek. *Project Manager Assistant* membawahi dua jabatan *user* lainnya yaitu *Developer* dan *Tester*. Gambar 3.1 berikut ini merupakan struktur *user* dalam sistem.



Gambar 3. 1 Struktur *user*

Setiap pengguna mempunyai peran yang berbeda-beda. Kelima pengguna dan perannya masing-masing yaitu sebagai berikut:

3.1.1 Admin

Admin merupakan pengguna yang berperan sebagai pengelola semua data pengguna. Pada diagram struktur *user*, admin menempati posisi yang paling tinggi dibandingkan dengan *user* lainnya. Admin memiliki beberapa tugas yang dapat dikerjakan pada sistem, seperti mengelola data semua pengguna, mengelola data proyek, dan mengakses data *bug*.

3.1.2 Project Manager

Project manager merupakan salah satu pengguna sistem yang bertugas memimpin jalannya sebuah proyek. Di dalam diagram struktur *user*, *project manager* menempati posisi pada level kedua atau berada tepat dibawah admin. Ia akan memimpin timnya yang terdiri dari beberapa orang seperti *project manager assistant*, *developer*, dan *tester* yang terlibat pada proyek yang dipimpinya. Pada sistem, seorang *project manager* dapat mengakses data proyek, mengakses data *bug*, dan juga dapat menerima penugasan pengerjaan perbaikan *bug*.

3.1.3 Project Manager Assistant

Seperti pada gambar 3.1, *project manager assistant* atau asisten manajer proyek merupakan bawahan dari seorang *project manager*, dimana ia bertugas dalam pengelolaan data proyek, dan pengelolaan data *bug*. Pada sistem, selain dapat mengelola data *bug*, asisten manajer proyek juga dapat mengakses data progres *bug* dan menerima tugas dari *user* lain untuk memperbaiki suatu *bug*.

3.1.4 Developer atau Programmer

Developer atau *programmer* memiliki peran sebagai pembangun sistem pada suatu proyek serta melakukan perbaikan pada kesalahan atau kerusakan yang ditemukan dan ditugaskan oleh *user* lain. Di dalam diagram struktur *user*, *developer* berada tepat dibawah asisten manajer proyek dan berdampingan dengan *tester*. Pada sistem, *developer* juga dapat mengakses data proyek, mengelola data *bug* dan mengelola data progres *bug*. *Developer* akan melihat data-data *bug* pada sistem yang ditugaskan padanya dan perlu ditindaklanjuti, lalu melakukan *update* pada status perbaikan *bug*. Jika *bug* sudah berhasil diperbaiki, maka *developer* akan melakukan *update* kembali pada bagian status dan tanggal *bug* berhasil diperbaiki.

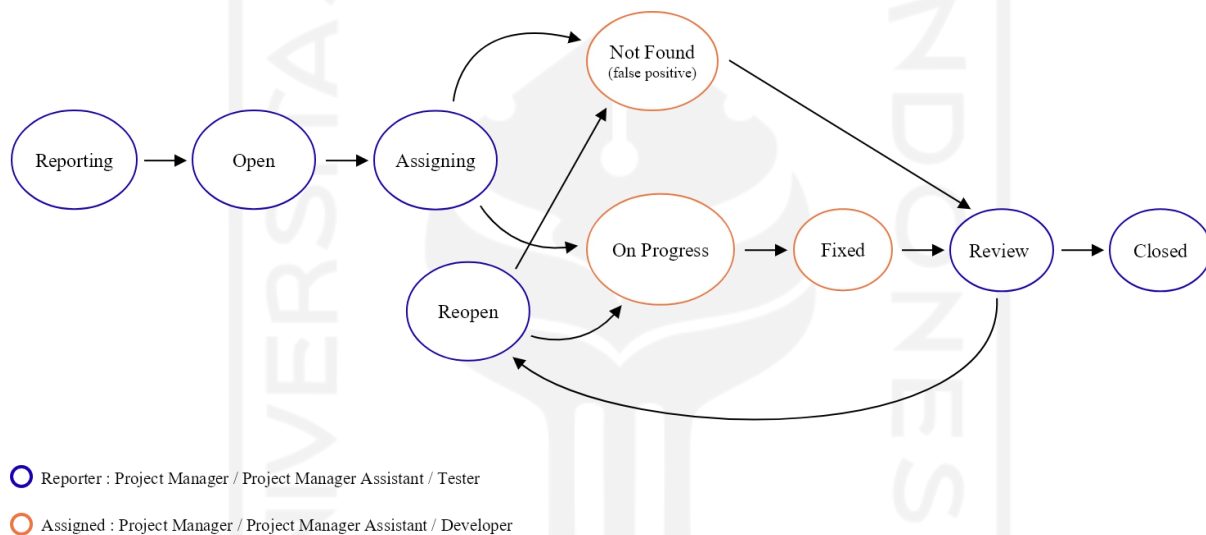
3.1.5 Tester

Berdasarkan diagram struktur *user* pada gambar 3.1, seorang *tester* dalam sebuah proyek menempati posisi sejajar dengan *developer*. Ia bertugas melakukan pengujian pada proyek, mencari kerusakan atau kesalahan seperti *bug* pada proyek, lalu mendokumentasikan dan mencatatnya pada sistem sebagai media pelaporan kepada *developer* atau *user* lain yang

selanjutnya akan ditindaklanjuti oleh mereka untuk diperbaiki. Pada sistem, *tester* dapat mengakses data proyek, mengelola data bug, dan mengakses data progres *bug*. Selain dapat menugaskan perbaikan *bug* pada *user* lain, seorang *tester* juga dapat menerima penugasan perbaikan *bug* dari *user* yang lainnya.

3.2 Bug Life Cycle

Setiap *bug* memiliki siklus hidupnya sendiri, mulai dari ditemukan hingga *bug* berhasil diperbaiki. Berikut ini merupakan gambar dari siklus hidup sebuah *bug* pada Sistem Informasi Pencatatan *Bug*.



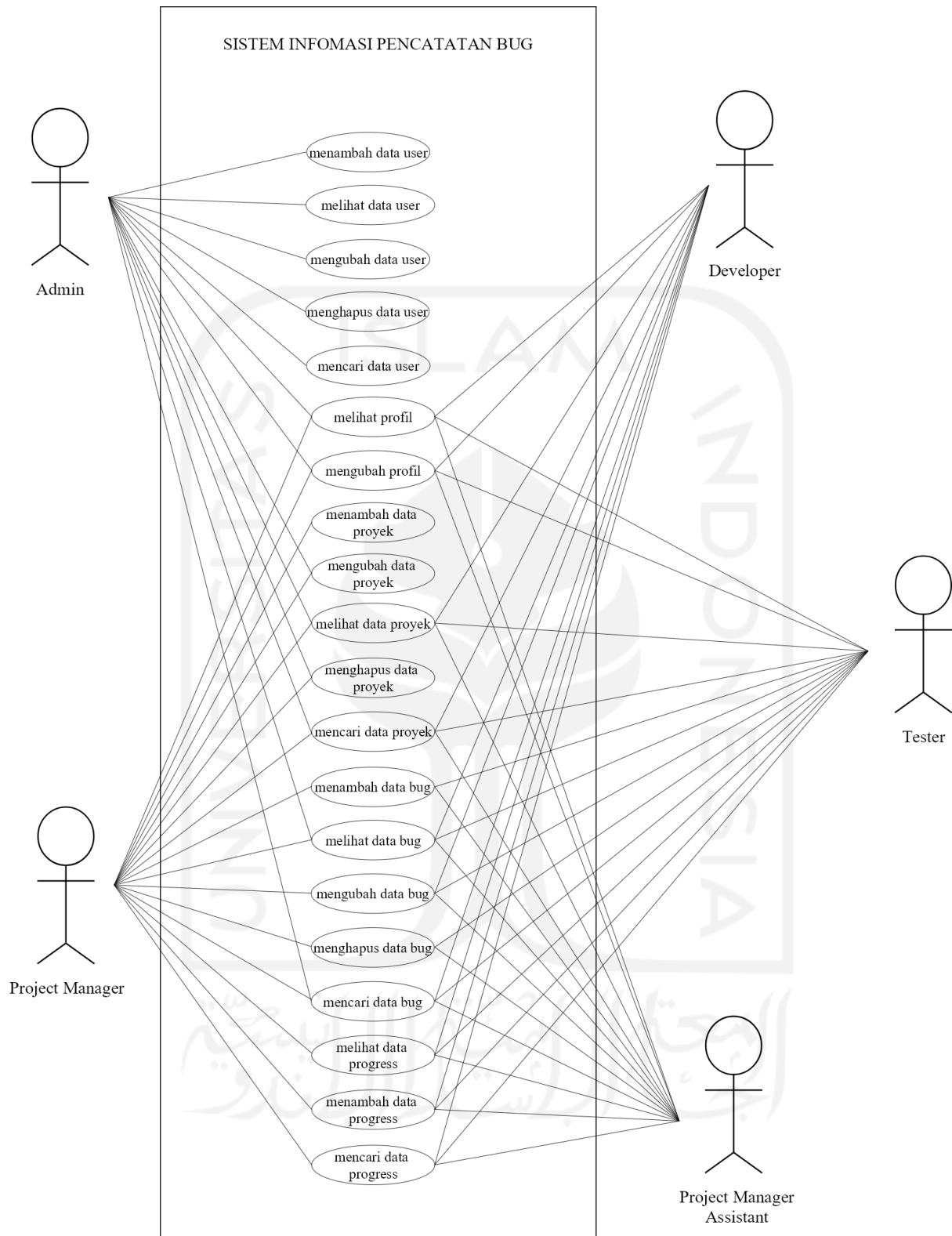
Gambar 3. 2 Bug life cycle

Berdasarkan Gambar 3.2, tahapan dengan *outline* biru menandakan kegiatan tersebut dilakukan oleh pelapor atau *reporter*, sedangkan yang berwarna *orange* dilakukan oleh pengguna yang diberi tugas atau *assigned*. Pengguna yang dapat menjadi seorang pelapor *bug* ialah *Project Manager*, *Project Manager Assistant*, atau *Tester*, sedangkan yang dapat diberi tugas untuk memperbaiki *bug* ialah *Project Manager*, *Project Manager Assistant*, atau *Developer*. Siklus hidup sebuah *bug* dimulai saat *tester* atau *user* lain menemukan *bug* pada sebuah proyek lalu melaporkannya atau mencatat kejadian tersebut pada Sistem Informasi Pencatatan *Bug*. Setelah melalui tahap *reporting*, status *bug* akan otomatis menjadi *open*. Kemudian masuk ke tahap *assigning*, yaitu penentuan kepada siapa perbaikan *bug* akan ditugaskan. Orang yang telah terpilih sebagai orang yang ditugaskan untuk memperbaiki *bug* ini kemudian mengubah status *bug* menjadi *on progress* jika ia mulai melakukan perbaikan,

atau mengubah status *bug* menjadi *not found* jika *bug* yang dilaporkan ternyata tidak ditemukan (*false positive*). Jika orang yang ditugaskan sudah mengubah status *bug* menjadi *not found*, maka pelapor akan melakukan pengecekan ulang terhadap *bug* yang dilaporkannya, jika memang tidak ada pelapor dapat menutup pelaporan *bug* dengan mengubah statusnya menjadi *closed*, atau bisa mengubah status *bug* menjadi *reopen* agar orang yang ditugaskan dapat melakukan pengecekan kembali terhadap *bug* tersebut. Selanjutnya, jika orang yang ditugaskan telah selesai mengerjakan atau memperbaiki sebuah *bug*, maka ia dapat mengubah status *bug* dari *on progress* menjadi *fixed*. Jika *bug* sudah berstatus *fixed*, maka selanjutnya pelapor *bug* melakukan *review* atau pengecekan kembali terhadap *bug* yang telah diperbaiki. Jika *bug* sudah tidak ditemukan dan tidak ada keluhan lain, maka ia dapat menutup pelaporan *bug* dengan mengubah status *bug* menjadi *closed*. Namun jika ia merasa *bug* masih ada dan masih perlu ditindaklanjuti, maka ia dapat mengubah status *bug* menjadi *reopen* sehingga orang yang ditugaskan dapat melakukan pengecekan ulang atau memperbaiki *bug* tersebut kembali.

3.3 Use Case Diagram

Use case diagram berfungsi untuk menggambarkan interaksi apa saja yang dapat dilakukan oleh satu atau lebih aktor terhadap sebuah sistem. Seperti yang sudah dijelaskan pada bab sebelumnya, terdapat lima aktor dalam sistem ini yaitu *Project Manager*, *Project Manager Assistant*, *Admin*, *Developer*, dan *Tester*. Berikut *use case diagram* tiap aktor pada Sistem Informasi Pencatatan *Bug* ini.



Gambar 3.3 Use case diagram

Sebagai admin pada sistem, berdasarkan Gambar 3.3, ada tiga belas aktivitas yang dapat dilakukan Admin setelah berhasil melakukan proses *login* sistem. Seorang Admin dapat

melihat dan mengubah data profil akun miliknya, melihat data semua *user*, menambahkan data *user* baru, melakukan perubahan, penghapusan, dan pencarian data *user* yang telah dimasukkan pada sistem, melihat data semua proyek, melakukan perubahan, penghapusan, dan pencarian data proyek, melihat semua data *bug*, serta melakukan pencarian data *bug* yang telah dimasukkan pada sistem.

Seorang *Project Manager* memiliki 15 aktivitas yang dapat dilakukannya pada sistem setelah berhasil melewati proses *login*. Pada sistem, *Project Manager* dapat melihat dan mengubah data profil akun miliknya, melihat data semua proyek, menambahkan data proyek baru, melakukan perubahan, penghapusan, dan pencarian data proyek, melihat data bug, menambahkan data *bug* baru, mengubah data *bug*, menghapus data bug, mencari data bug, menambah dan melihat data progres *bug*, serta mencari data progres *bug*.

Selanjutnya, *Project Manager Assistant* dapat melakukan 12 aktivitas pada sistem yaitu melihat dan mengubah data profil akun miliknya, melihat data proyek, mencari data proyek yang sudah dimasukkan ke sistem, menambahkan, melihat, mengubah, menghapus dan mencari data bug, serta menambah, melihat, dan mencari data progress. Aktivitas-aktivitas tersebut dapat dilakukan oleh asistem manajer proyek setelah berhasil melakukan tahap *login* sistem.

Setelah berhasil *login*, terdapat beberapa aktivitas yang dapat dilakukan oleh seorang *Developer*. Aktivitas-aktivitas tersebut ialah melihat dan mengubah data profil akun miliknya, melihat data proyek, mencari data proyek, melihat data *bug*, melakukan perubahan dan pencarian terhadap data *bug*, melihat data *progress*, serta melakukan penambahan dan pencarian terhadap data *progress*.

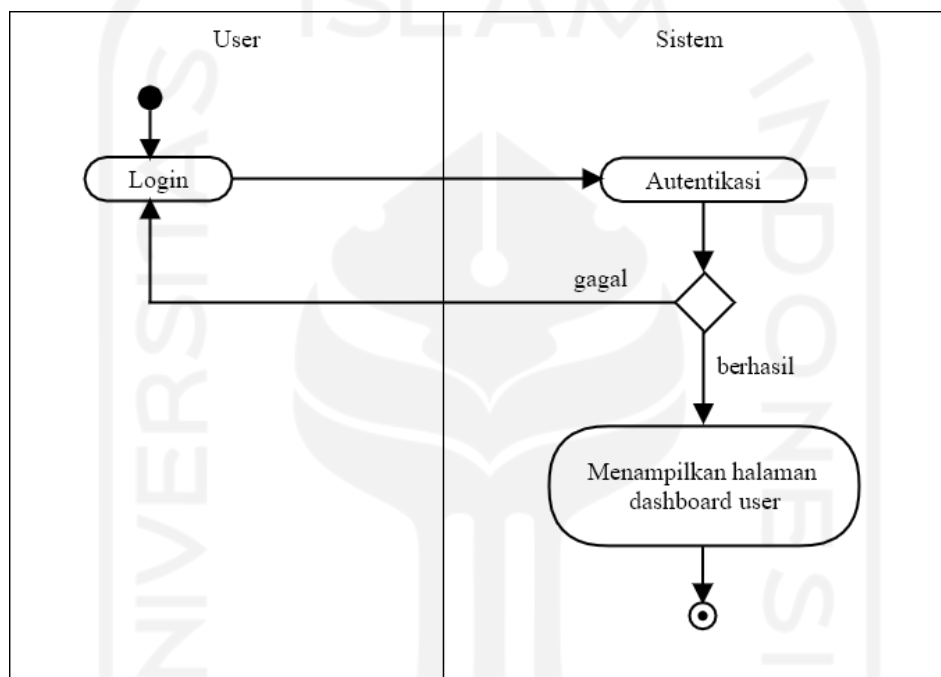
Terakhir untuk *Tester*, ada beberapa aktivitas yang dapat dilakukannya pada sistem setelah berhasil melakukan *login*, yaitu melihat dan mengubah data profil akun miliknya, melihat data proyek, mencari data proyek. melihat data *bug*, melakukan penambahan data *bug* baru, melakukan perubahan, penghapusan, dan pencarian terhadap data *bug*, menambahkan dan melihat data *progress*, serta melakukan pencarian data *progress*.

3.4 Activity Diagram

Activity diagram menggambarkan alur proses terjadinya tiap aktivitas yang terjadi pada Sistem Informasi Pencatatan Bug. Berbagai aktivitas tersebut yaitu *login*, perubahan profil, pengelolaan data user, pengelolaan data proyek, pengelolaan data *bug*, dan pengelolaan data progres *bug*.

3.4.1 Activity Diagram Login

Aktivitas *login* ini dilakukan oleh semua *user* agar dapat masuk ke sistem. Untuk masuk ke dalam sistem, tiap *user* harus melakukan proses *login* terlebih dahulu. *Login* dilakukan dengan menggunakan *google account*. *User* memasukkan *email* dan *password* yang sudah terdaftar sebagai akun *google*. Selanjutnya sistem akan menampilkan *dashboard user* jika *login* telah berhasil, namun bila *login* gagal maka *user* akan tetap berada pada halaman *login*. Aktivitas *login* ini digambarkan seperti pada gambar 3.4 berikut.

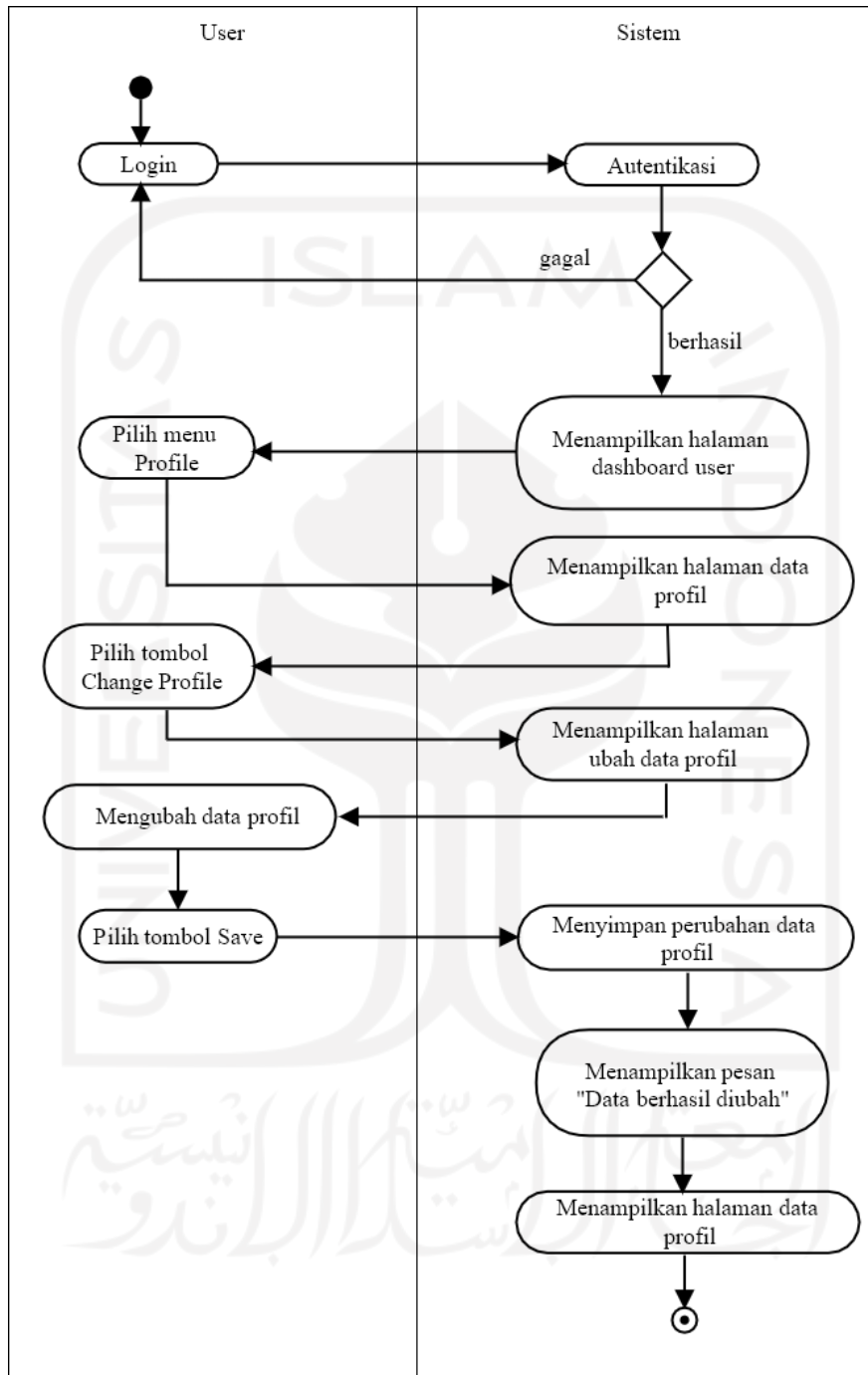


Gambar 3. 4 Activity diagram login

3.4.2 Activity Diagram Mengubah Profil

Aktivitas perubahan profil adalah kegiatan perubahan data profil seorang *user* jika terdapat data profil yang tidak sesuai dan aktivitas ini dapat dilakukan oleh semua *user*. Setelah berhasil melakukan proses *login*, *user* akan masuk ke halaman *dashboard*. Untuk melakukan perubahan profil, *user* memilih menu *Profile* yang berada pada bagian kiri halaman. Setelah memilih menu *Profile*, *user* akan masuk ke halaman data profil. Pada halaman tersebut terdapat tombol *Change Profile*, pilih tombol tersebut untuk masuk ke halaman ubah data profil. Lalu *user* dapat melakukan perubahan data profil miliknya. Selanjutnya *user* menekan tombol *Save* agar sistem dapat menyimpan perubahan data profil. Setelah perubahan data profil tersimpan, sistem akan kembali menampilkan halaman profil

dengan data yang terbaru. Aktivitas pengubahan data profil tersebut digambarkan seperti pada gambar 3.5 berikut ini.



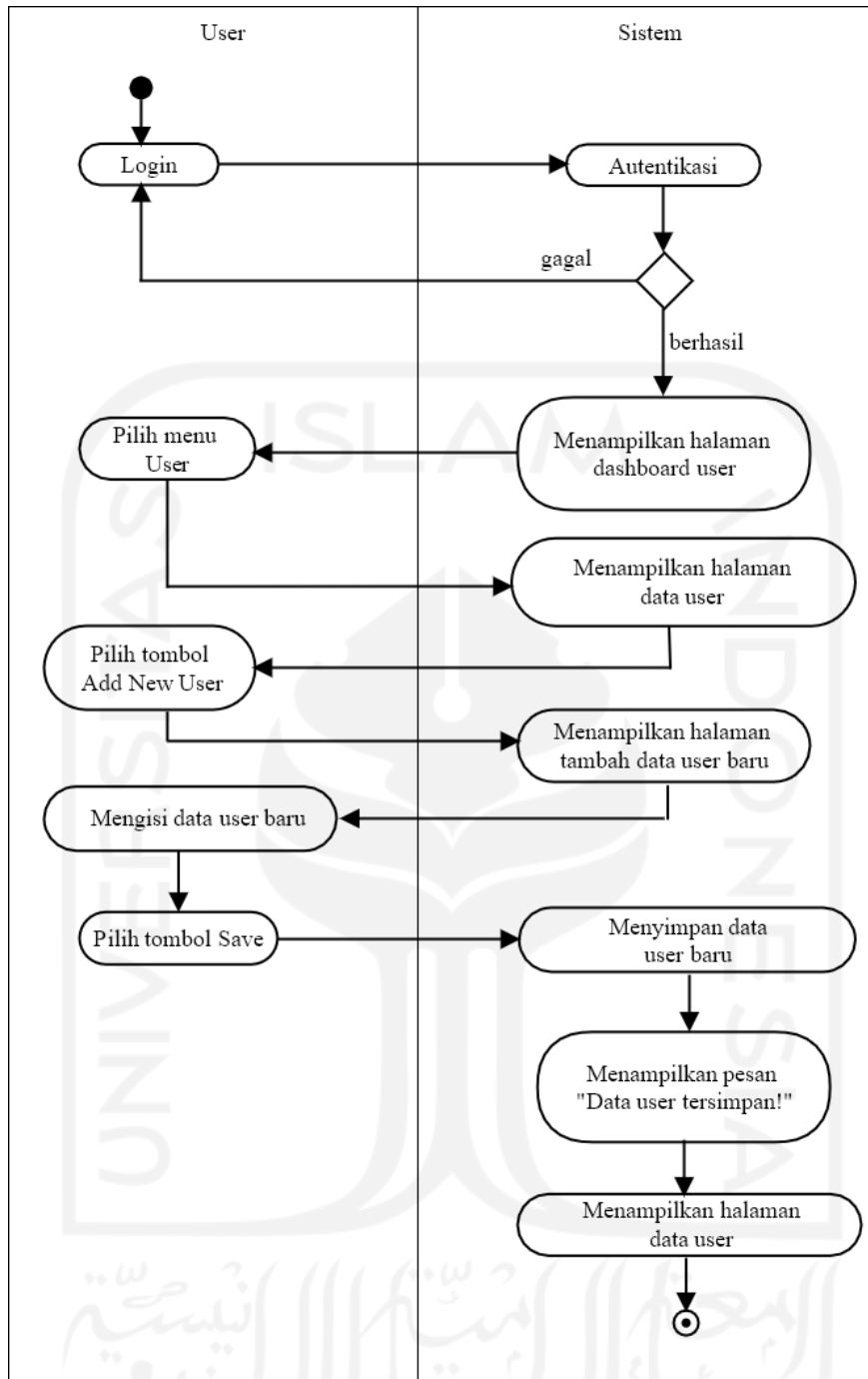
Gambar 3. 5 Activity diagram mengubah profil

3.4.3 Activity Diagram Menambah Data User

Aktivitas penambahann data *user* ini hanya dapat dilakukan oleh admin. Untuk manambahkan data *user*, seorang admin harus melakukan *login* terlebih dahulu. Jika proses

login gagal, maka admin akan tetap berada pada halaman *login*. Apabila admin berhasil melakukan proses *login*, maka admin akan masuk ke halaman *dashboard* admin. Pada halaman *dashboard* admin, admin memilih menu *User* pada bagian kiri halaman. Selanjutnya, sistem akan menampilkan halaman data *user*. Penambahan data *user* dilakukan dengan memilih tombol *Add New User* dibagian bawah pada halaman tersebut, sehingga sistem akan menampilkan halaman *form* untuk menambah data *user*. Pada halaman tambah *user*, admin diminta untuk mengisi data *user* yang ingin dimasukkan ke dalam sistem, lalu memilih tombol *Save* jika pengisian data sudah selesai. Sistem akan menyimpan data yang telah dimasukkan, lalu sistem akan menampilkan kembali halaman data *user* dengan data yang terbaru. Aktivitas ini digambarkan sesuai dengan gambar 3. 6 berikut ini.



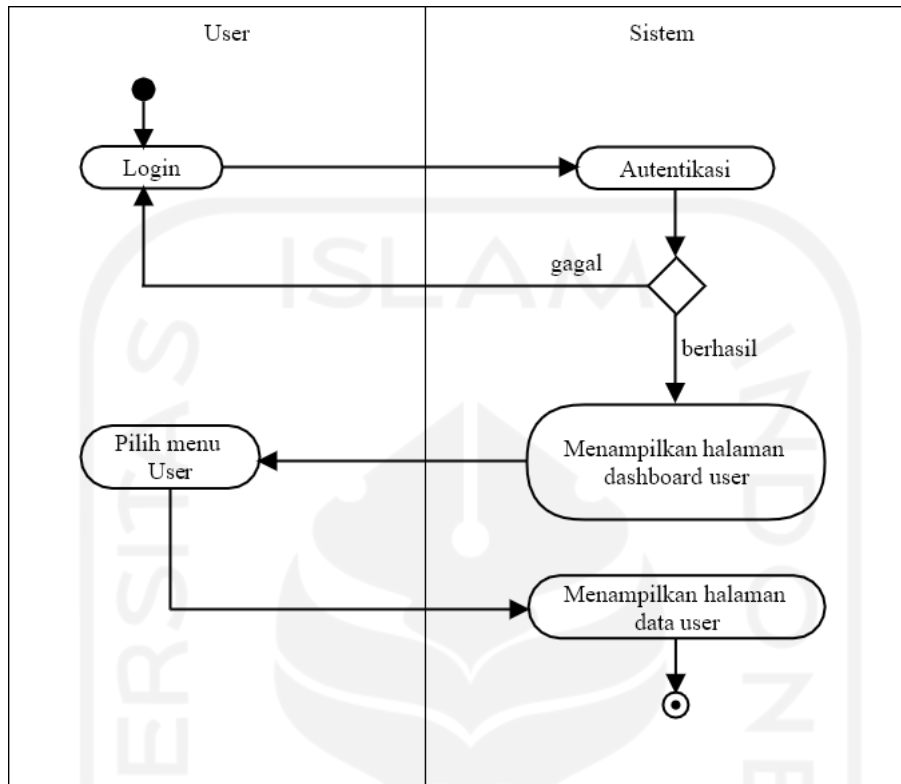


Gambar 3. 6 Activity diagram menambah data user

3.4.4 Activity Diagram Melihat Data User

Pengaksesan data semua *user* hanya dapat dilakukan oleh *user* dengan jabatan sebagai admin. Pertama, admin melakukan aktivitas *login* dengan memasukkan *email* dan *password* yang sesuai. Jika *login* gagal, maka admin akan tetap berada pada halaman *login*. Sebaliknya, jika *login* berhasil, maka sistem akan menampilkan *dashboard* admin. Untuk melihat data *user*, admin memilih menu *User* pada bagian *navbar* di kiri halaman, lalu sistem akan

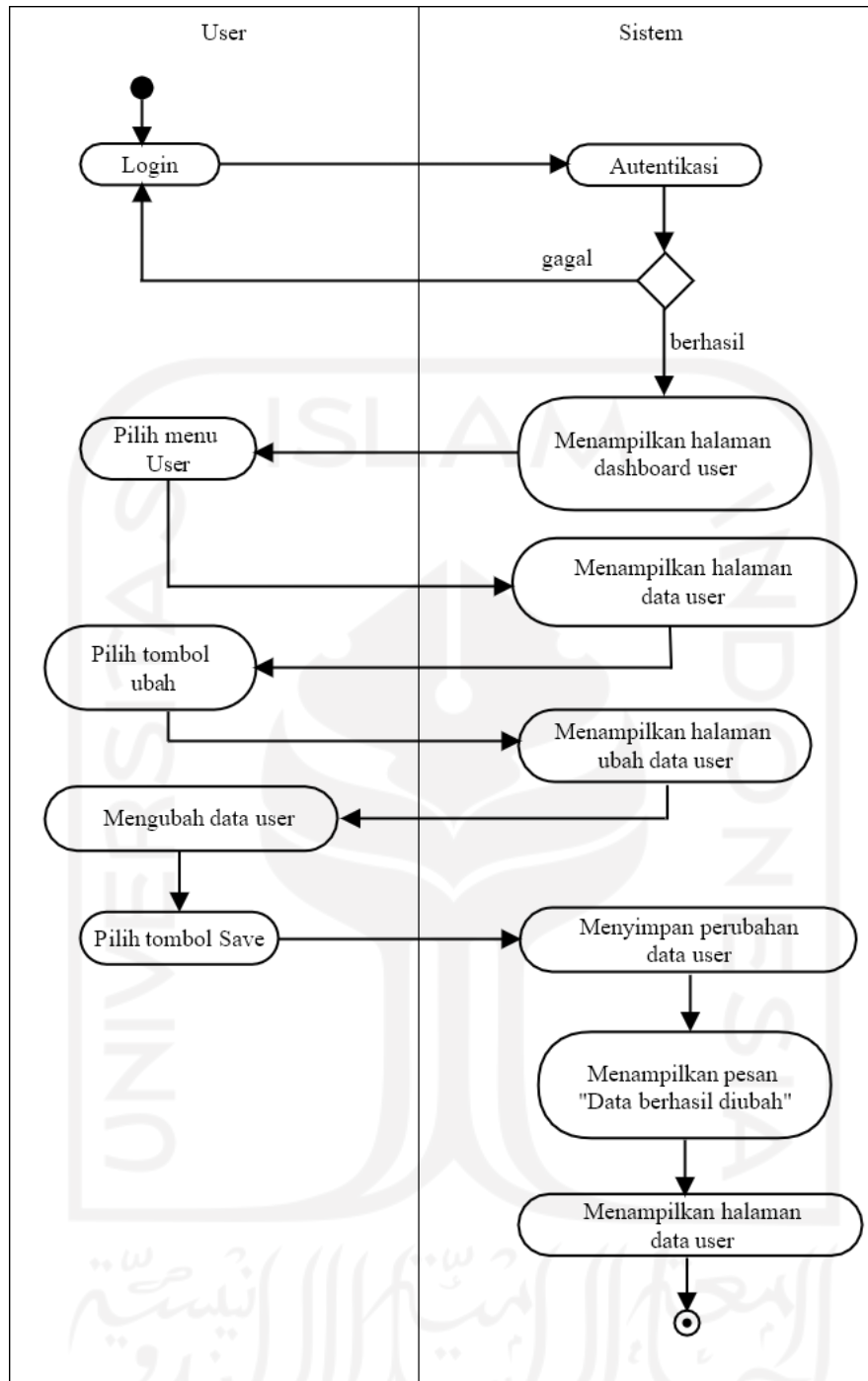
menampilkan halaman data *user*. Aktivitas melihat data *user* divisualkan seperti pada gambar 3.7 berikut ini.



Gambar 3. 7 Activity diagram melihat data *user*

3.4.5 Activity Diagram Mengubah Data User

Aktivitas pengubahan data *user* merupakan kegiatan pengubahan data *user* jika terdapat data yang salah atau tidak sesuai. Aktivitas ini hanya dapat dilakukan oleh seorang admin. Pada gambar 3.8 berikut ini menggambarkan alur dari aktivitas pengubahan data *user*. Pertama, admin melakukan *login* yang jika gagal, maka admin akan tetap berada di halaman *login*, namun jika *login* berhasil, maka sistem akan menampilkan halaman *dashboard* admin. Kemudian, admin memilih menu *User* dan sistem akan menampilkan halaman data *user*. Pada halaman data *user* tersebut terdapat tombol ubah berwarna *orange* di kolom *action*. Admin memilih tombol ubah tersebut, lalu sistem akan menampilkan halaman ubah data *user*. Admin melakukan pengubahan data *user* dan menekan tombol *Save* jika pengubahan sudah selesai. Sistem akan menyimpan data perubahan tersebut. Setelah itu, sistem akan menampilkan kembali halaman data *user* dengan data yang terbaru.

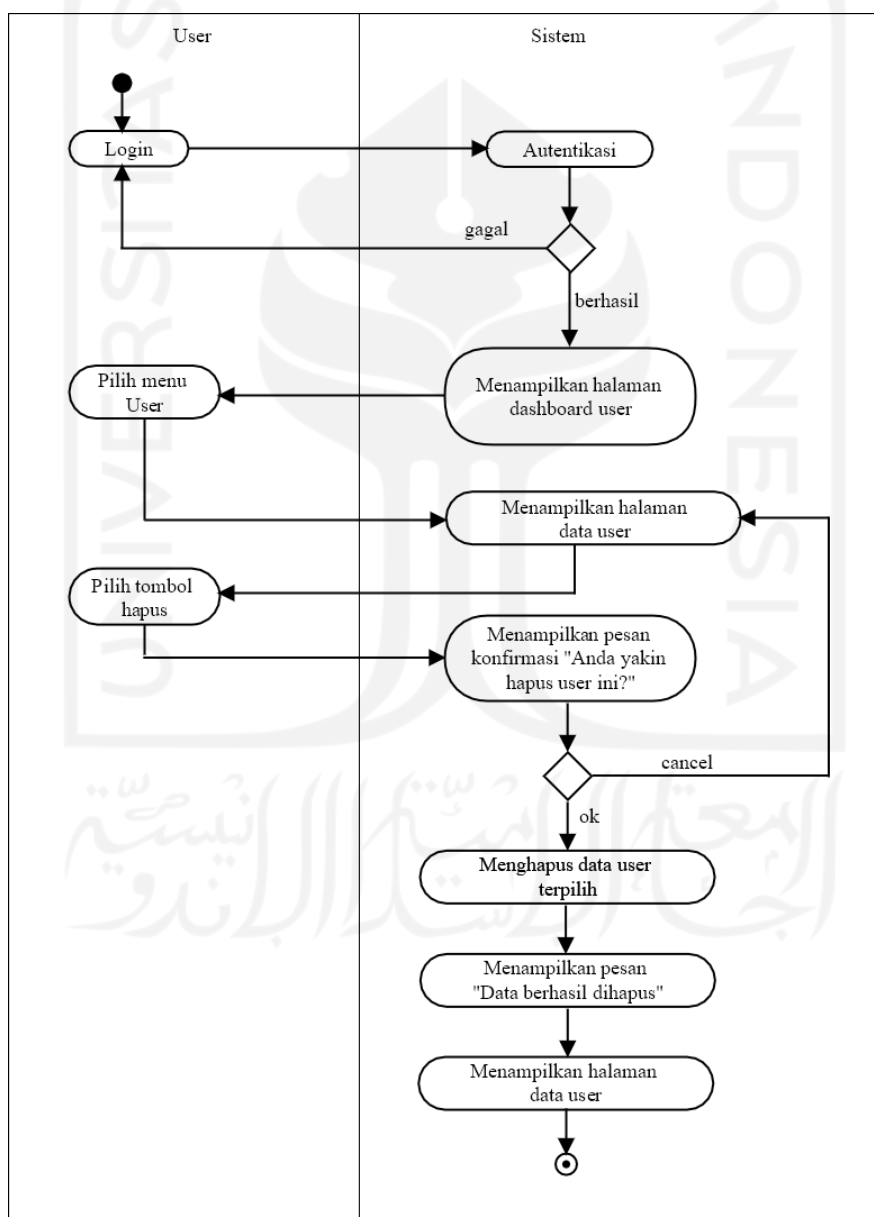


Gambar 3. 8 Activity diagram mengubah data user

3.4.6 Activity Diagram Menghapus Data User

Aktivitas hapus data *user* merupakan kegiatan penghapusan data *user* dari daftar data *user* yang telah dimasukkan ke sistem oleh admin. Kegiatan ini juga hanya dapat dilakukan oleh *user* dengan jabatan sebagai admin. Hal pertama yang harus dilakukan admin untuk menghapus data *user* adalah *login*. Jika *login* gagal, maka admin akan tetap berada pada

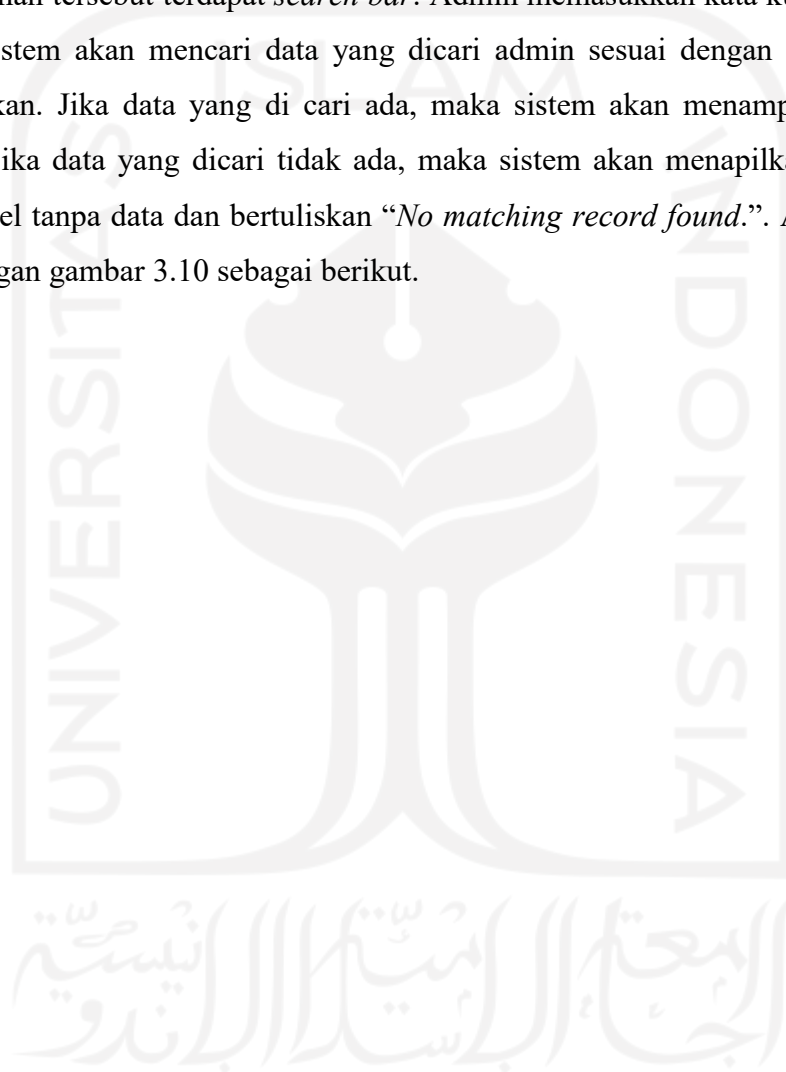
halaman *login*. Jika berhasil, maka sistem akan menampilkan halaman *dashboard* admin. Setelah itu admin memilih menu *User*, lalu sistem akan menampilkan halaman data *user*. Admin memilih tombol hapus berwarna merah pada kolom *action* yang sejajar dengan baris data *user* yang ingin dihapus. Kemudian sistem akan menampilkan pesan konfirmasi “Anda yakin hapus *user* ini?”. Jika admin memilih *cancel*, maka sistem akan menampilkan kembali halaman data *user* tanpa melakukan penghapusan data. Jika admin memilih *OK*, maka sistem akan menghapus data *user* tersebut lalu menampilkan kembali halaman data *user* dengan daftar data yang terbaru. Aktivitas ini sebagaimana divisualkan pada gambar 3.9 berikut.

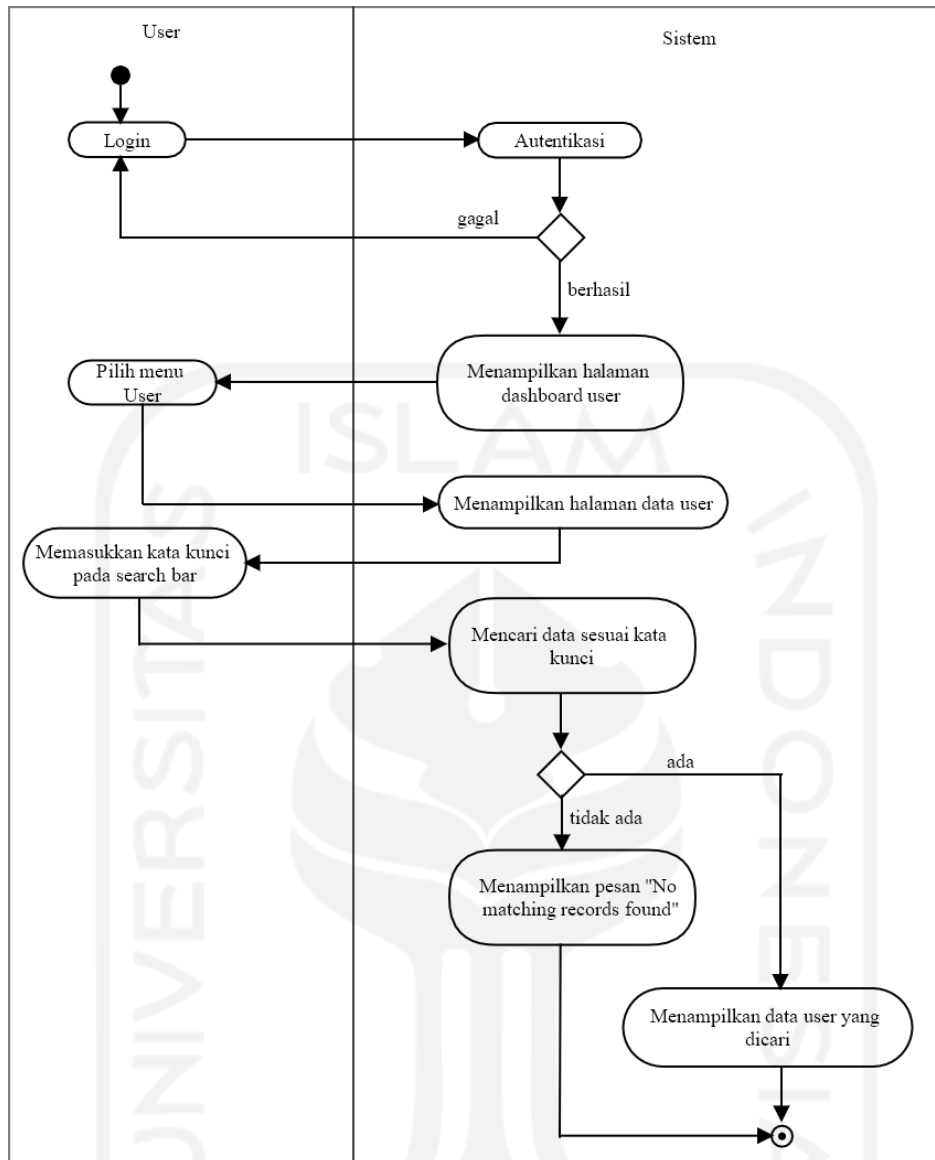


Gambar 3. 9 Activity diagram menghapus data user

3.4.7 Activity Diagram Mencari Data User

Aktivitas pencarian data *user* ini hanya dapat dilakukan oleh *user* dengan jabatan admin saja. Untuk mencari data *user* yang ada pada sistem, admin perlu melakukan *login* sistem terlebih dahulu. Jika *login* gagal, maka admin akan tetap berada pada halaman *login*. Namun jika *login* berhasil, maka sistem akan menampilkan halaman *dashboard* admin. Selanjutnya admin memilih menu *User* sehingga sistem akan menampilkan halaman data *user*. Pada halaman tersebut terdapat *search bar*. Admin memasukkan kata kunci pada *search bar* tersebut. Sistem akan mencari data yang dicari admin sesuai dengan kata kunci yang sudah dimasukkan. Jika data yang di cari ada, maka sistem akan menampilkan data yang dicari. Namun jika data yang dicari tidak ada, maka sistem akan menampilkan halaman data *user* dengan tabel tanpa data dan bertuliskan “*No matching record found.*”. Alur aktivitas ini divisualkan dengan gambar 3.10 sebagai berikut.



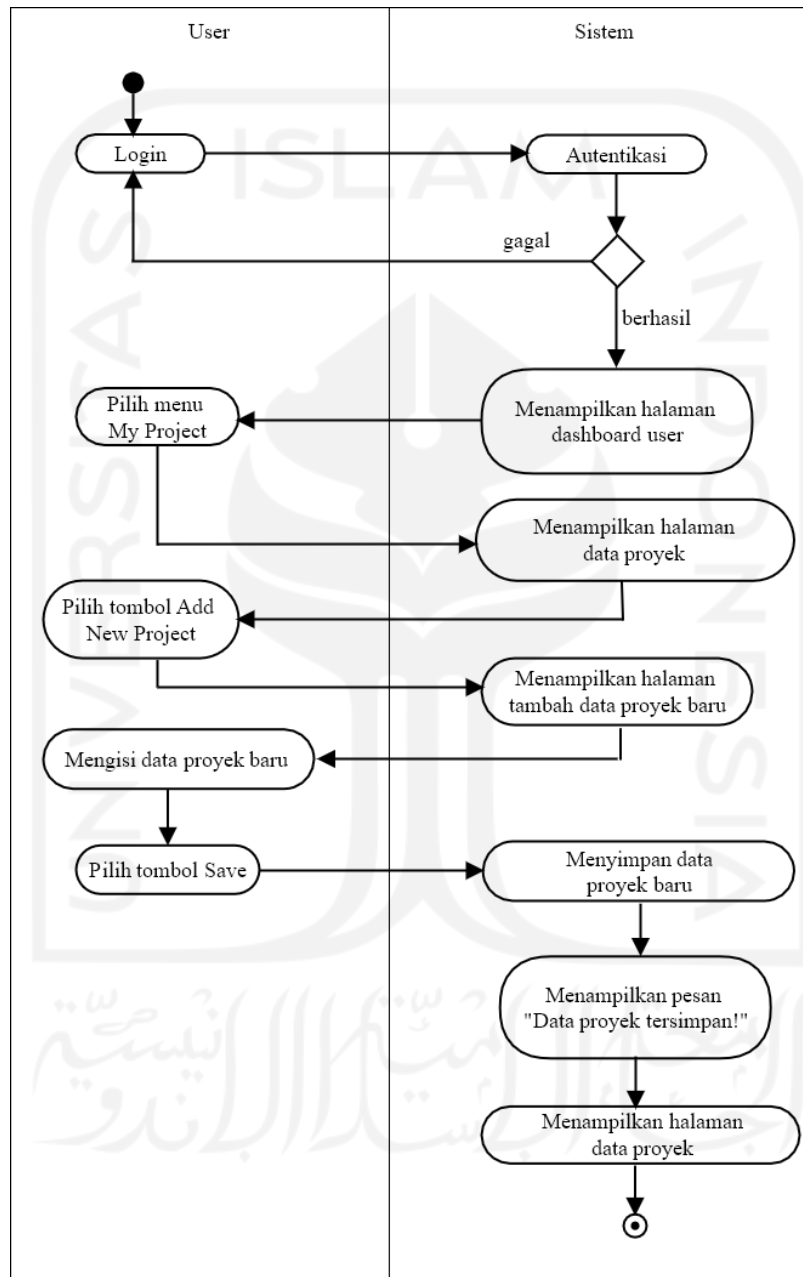


Gambar 3. 10 Activity diagram mencari data user

3.4.8 Activity Diagram Menambah Data Proyek

Aktivitas tambah data proyek merupakan kegiatan yang hanya dapat dilakukan oleh seorang *Project Manager* untuk memasukkan data proyek baru. Untuk menambah data proyek, seorang *Project Manager* harus melakukan *login* terlebih dahulu dengan memasukkan *email* dan *password* yang sudah terdaftar sebagai akun *google*. Jika *login* gagal, maka asisten manajer proyek akan tetap berada pada halaman *login*. Namun sebaliknya, jika *login* berhasil, maka sistem akan menampilkan halaman *dashboard Project Manager*. Selanjutnya *Project Manager* memilih menu *My Project* sehingga sistem akan menampilkan halaman data proyek. *Project Manager* menekan tombol *Add New Project*, lalu sistem menampilkan halaman halaman tambah data proyek. *Project Manager* memasukkan data

proyek yang ingin ditambahkan, kemudian menekan tombol *Save* jika sudah selesai. Sistem akan menyimpan data proyek tersebut, lalu sistem akan kembali menampilkan halaman data proyek namun dengan daftar data yang terbaru. Aktivitas penambahan data proyek ini divisualkan sebagaimana gambar 3.11 berikut ini.

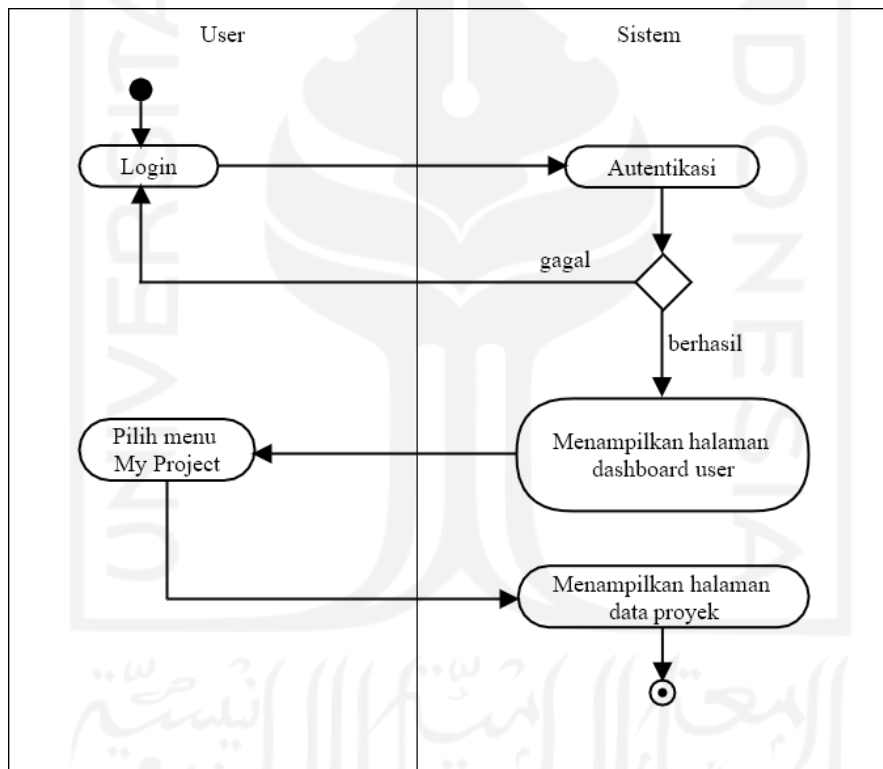


Gambar 3. 11 *Activity diagram* menambah data proyek

3.4.9 *Activity Diagram* Melihat Data Proyek

Aktivitas melihat data proyek merupakan kegiatan dimana *user* dapat melihat daftar data proyek. Daftar data proyek yang ditampilkan oleh sistem akan berbeda antara *user* dengan

jabatan admin dan *user* dengan jabatan lainnya. Seorang admin dapat melihat seluruh daftar data proyek yang telah dimasukkan ke sistem, sedangkan untuk *user* lain sistem hanya akan menampilkan daftar data proyek yang terlibat dengan dirinya saja. Untuk melihat data proyek, *user* harus *login* terlebih dahulu. *User* memasukkan *email* dan *password* yang sudah terdaftar sebagai pada *google*. Jika *login* gagal, maka *user* akan tetap berada pada halaman *login*. Jika *login* berhasil, maka *user* akan masuk ke dalam sistem dan sistem akan menampilkan halaman *dashboard user*. Kemudian, untuk admin, pilih menu *Project* pada *navbar*, sedangkan untuk *user* lain pilih menu *My Project* yang ada pada bagian *navbar* di kiri halaman. Dengan demikian, sistem akan menampilkan halaman data proyek. Aktivitas ini sama seperti pada gambar 3.12 berikut ini.

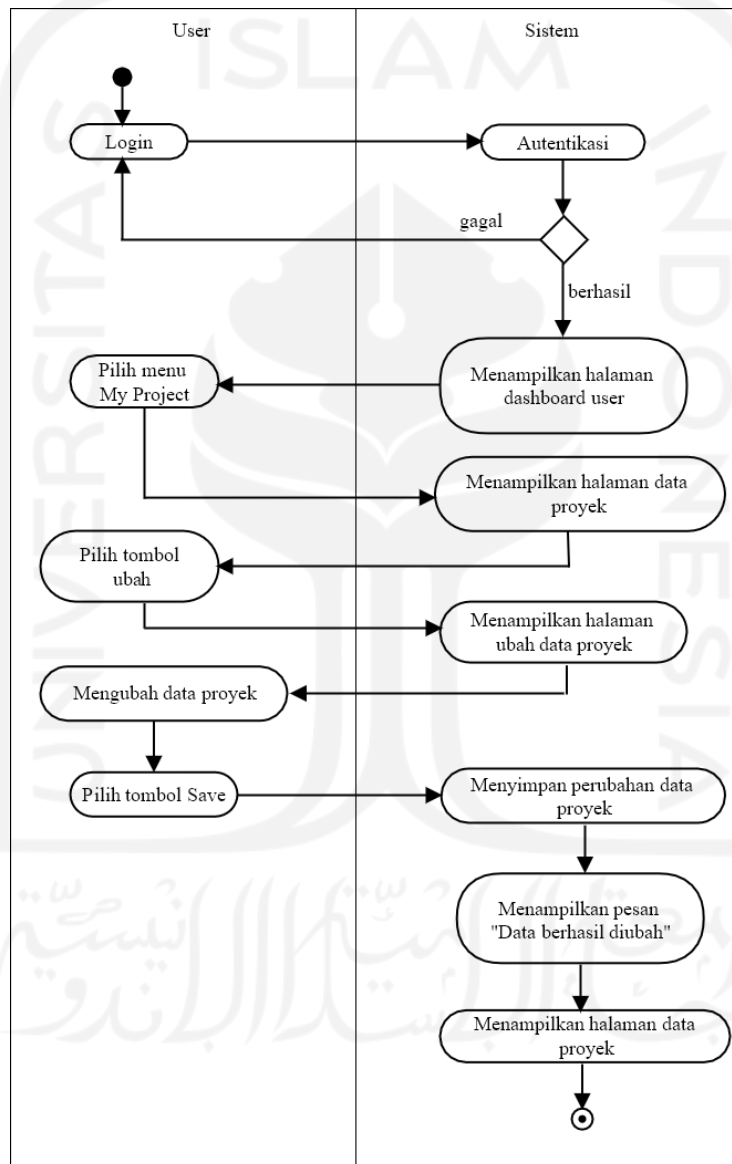


Gambar 3. 12 *Activity diagram* menampilkan data proyek

3.4.10 *Activity Diagram* Mengubah Data Proyek

Aktivitas ubah data proyek merupakan kegiatan perubahan terhadap data proyek yang salah atau tidak sesuai. Kegiatan perubahan data proyek ini hanya dapat dilakukan oleh seorang Admin dan *Project Manager*. *User* login dengan akun *google* terlebih dahulu. Jika proses *login* gagal, maka *user* akan tetap berada pada halaman *login*. Jika *login* berhasil, maka sistem akan menampilkan halaman *dashboard user*. Kemudian *user* memilih menu *My*

Project pada bagian *navbar* sehingga sistem akan menampilkan halaman data proyek. *User* menekan tombol ubah berwarna *orange* pada kolom *action* yang sejajar dengan proyek yang ingin diubah datanya. Sistem akan menampilkan halaman *form* ubah data proyek. *User* melakukan perubahan terhadap data proyek lalu menekan tombol *Save* bila sudah selesai. Selanjutnya sistem akan menyimpan perubahan tersebut lalu menampilkan kembali halaman data proyek yang terbaru. Aktivitas ini divisualkan seperti gambar 3.13 berikut ini.

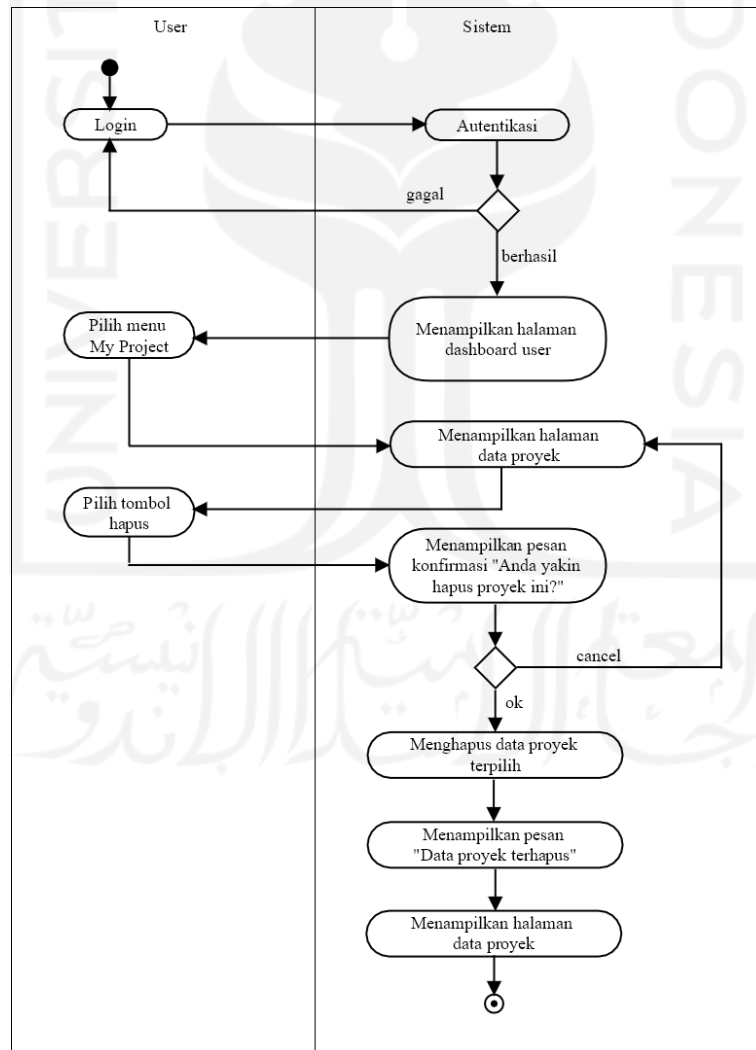


Gambar 3. 13 *Activity diagram* mengubah data proyek

3.4.11 *Activity Diagram* Menghapus Data Proyek

Aktivitas hapus data proyek merupakan kegiatan penghapusan terhadap data proyek yang sudah dimasukkan pada sistem. Kegiatan penghapusan data proyek ini hanya dapat

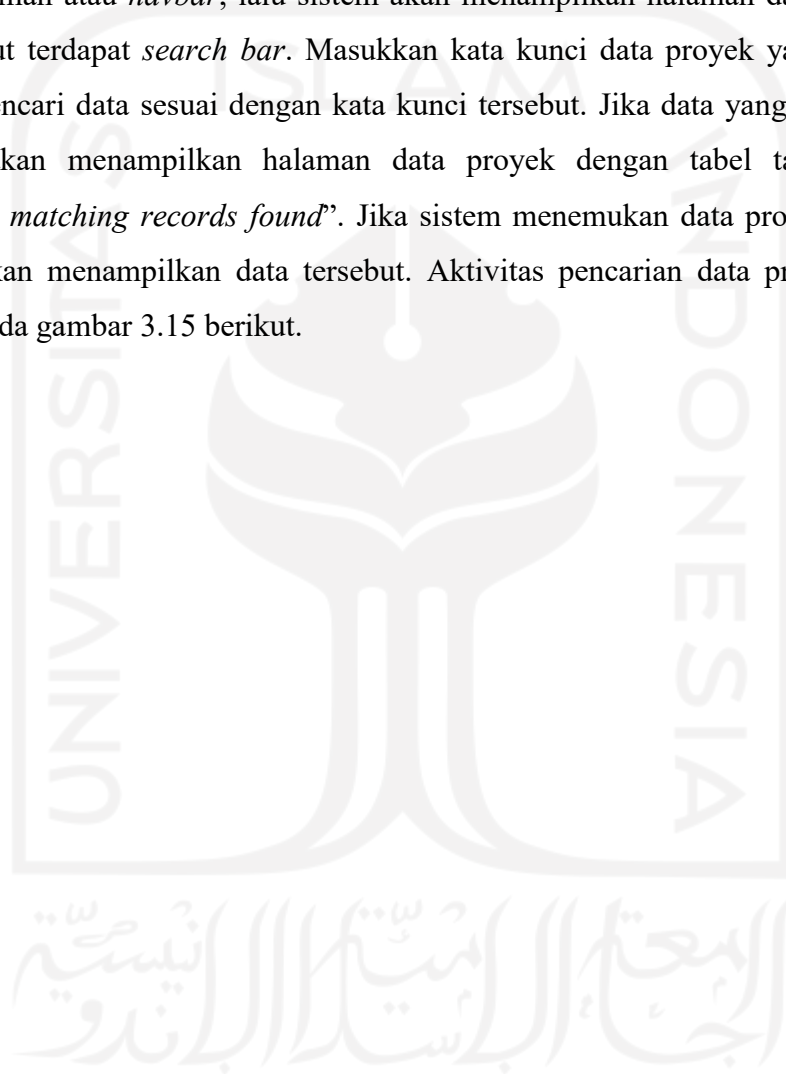
dilakukan oleh seorang Admin dan *Project Manager*. Pertama, *user* melakukan proses *login* terlebih dahulu. Jika gagal, maka *user* akan tetap berada pada halaman *login*. Jika berhasil, maka *user* akan masuk ke sistem. Sistem akan menampilkan *dashboard user* setelah proses *login* berhasil. Kemudian, *user* memilih menu *My Project* pada bagian *navbar* sehingga sistem akan menampilkan halaman data proyek. Pada halaman tersebut, terdapat tombol hapus berwarna merah di kolom *action*. *User* menekan tombol hapus tersebut yang sebaris dengan data yang ingin dihapus. Sistem akan menampilkan pesan konfirmasi penghapusan. Jika *user* memilih *cancel*, maka sistem akan menampilkan kembali halaman data proyek. Namun jika *user* memilih *OK*, maka sistem akan menghapus data proyek yang dimaksud lalu menampilkan halaman data proyek dengan daftar data yang terbaru. Aktivitas ini sebagaimana divisualkan seperti gambar 3.14 berikut.

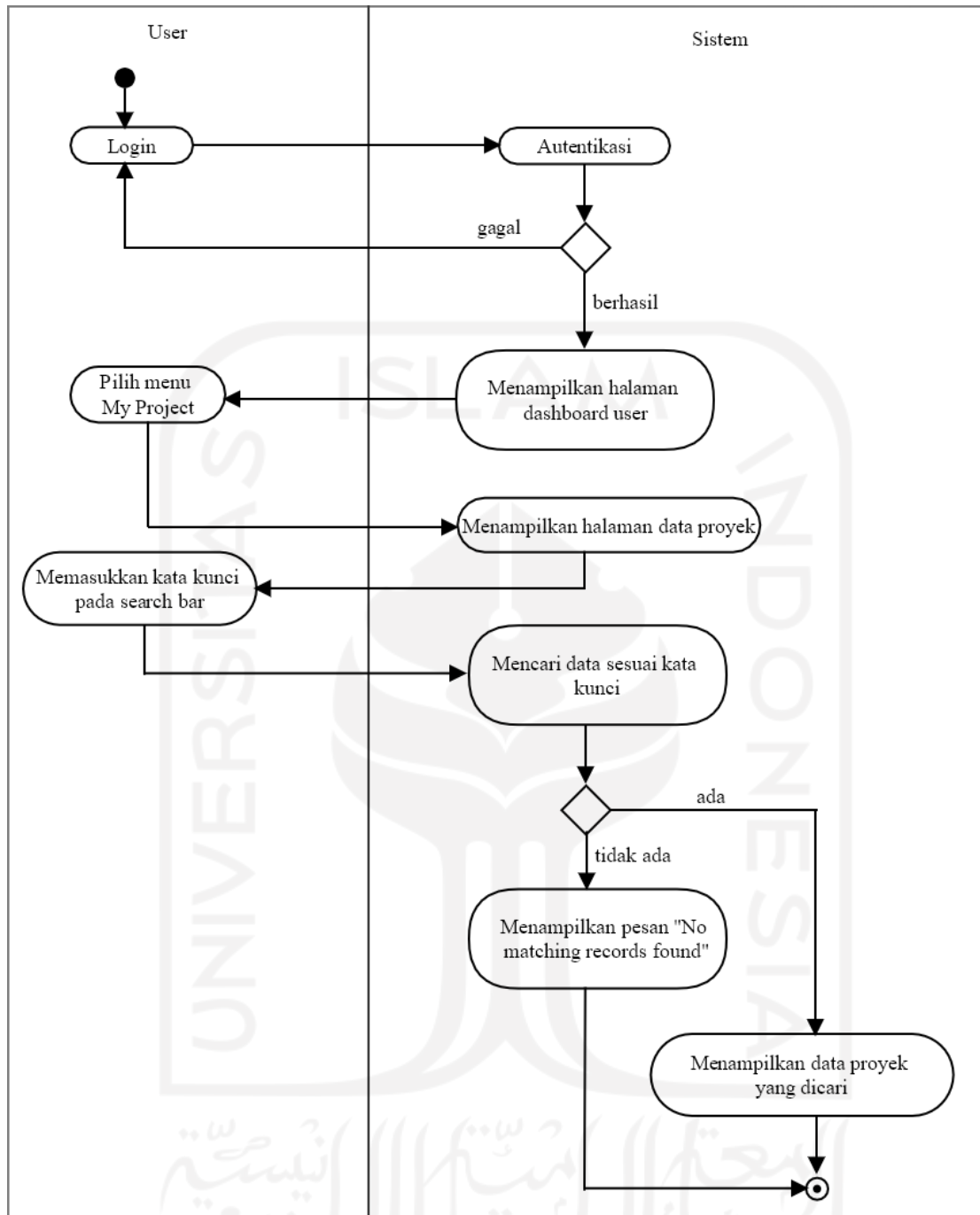


Gambar 3. 14 Activity diagram menghapus data proyek

3.4.12 Activity Diagram Mencari Data Proyek

Aktivitas pencarian data proyek ini dapat dilakukan oleh semua *user* untuk mencari data proyek dengan lebih cepat. Untuk melakukan pencarian data proyek, setiap *user* harus melakukan *login* terlebih dahulu. Jika *email* dan atau *password* tidak sesuai, maka *user* akan tetap berada pada halaman *login*. Jika *email* dan atau *password* sudah sesuai berarti *login* berhasil dan sistem akan menampilkan halaman *dashboard user*. Pilih menu *My Project* pada bagian kiri halaman atau *navbar*, lalu sistem akan menampilkan halaman data proyek. Pada halaman tersebut terdapat *search bar*. Masukkan kata kunci data proyek yang ingin dicari. Sistem akan mencari data sesuai dengan kata kunci tersebut. Jika data yang dicari tidak ada maka sistem akan menampilkan halaman data proyek dengan tabel tanpa data yang bertuliskan “*No matching records found*”. Jika sistem menemukan data proyek yang dicari, maka sistem akan menampilkan data tersebut. Aktivitas pencarian data proyek ini seperti digambarkan pada gambar 3.15 berikut.



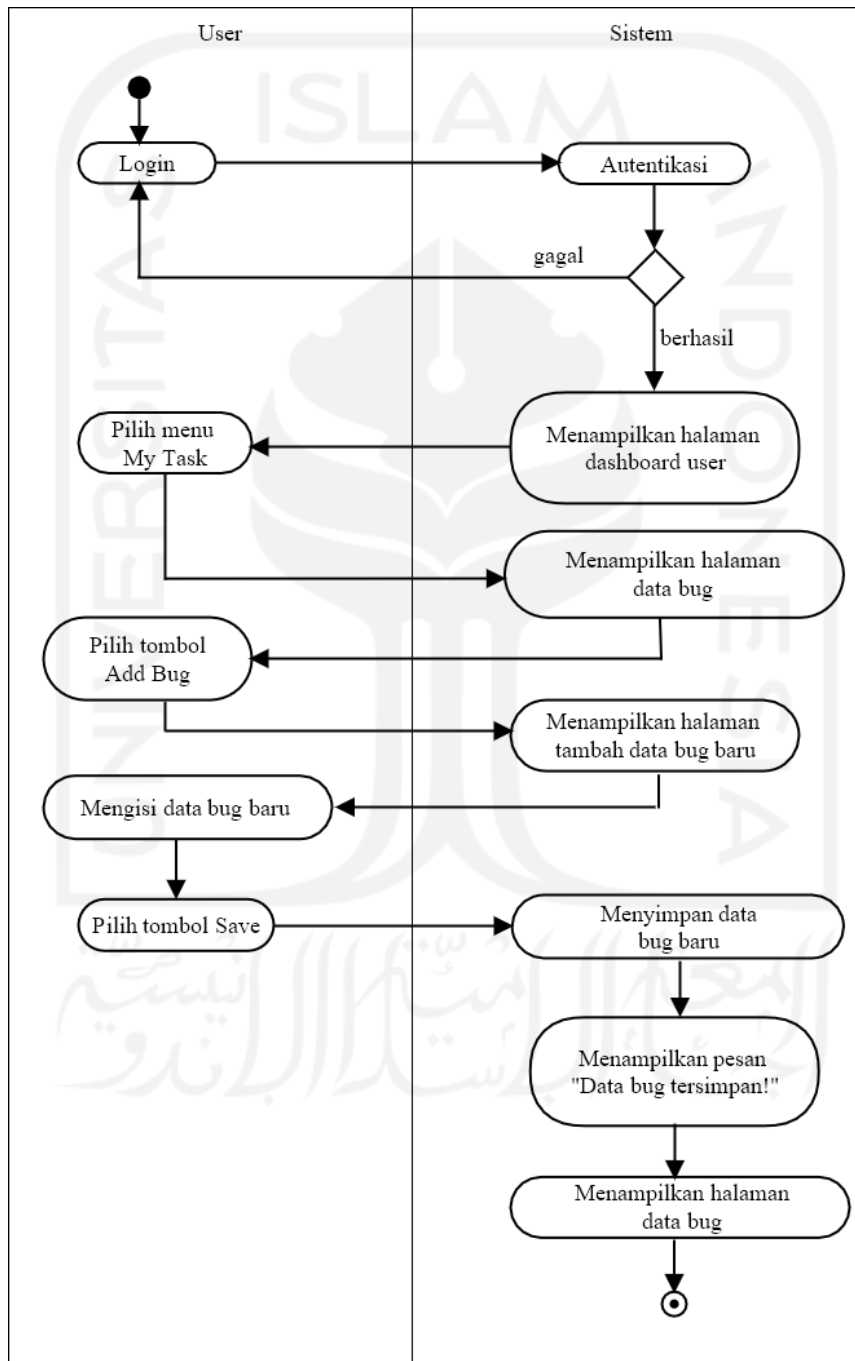


Gambar 3.15 Activity diagram mencari data proyek

3.4.13 Activity Diagram Menambah Data Bug

Aktivitas tambah data *bug* merupakan kegiatan yang dilakukan untuk memasukkan data bug baru pada sistem. Kegiatan ini hanya dapat dilakukan oleh *Project Manager*, *Project Manager Assistant*, dan *Tester*. Pertama-tama, *user* melakukan *login* dengan akun *google* terlebih dahulu. Jika *login* gagal maka *user* akan tetap berada pada halaman *login*. Jika *login* berhasil, maka sistem akan menampilkan *dashboard user*. Pilih menu *My Task* sehingga

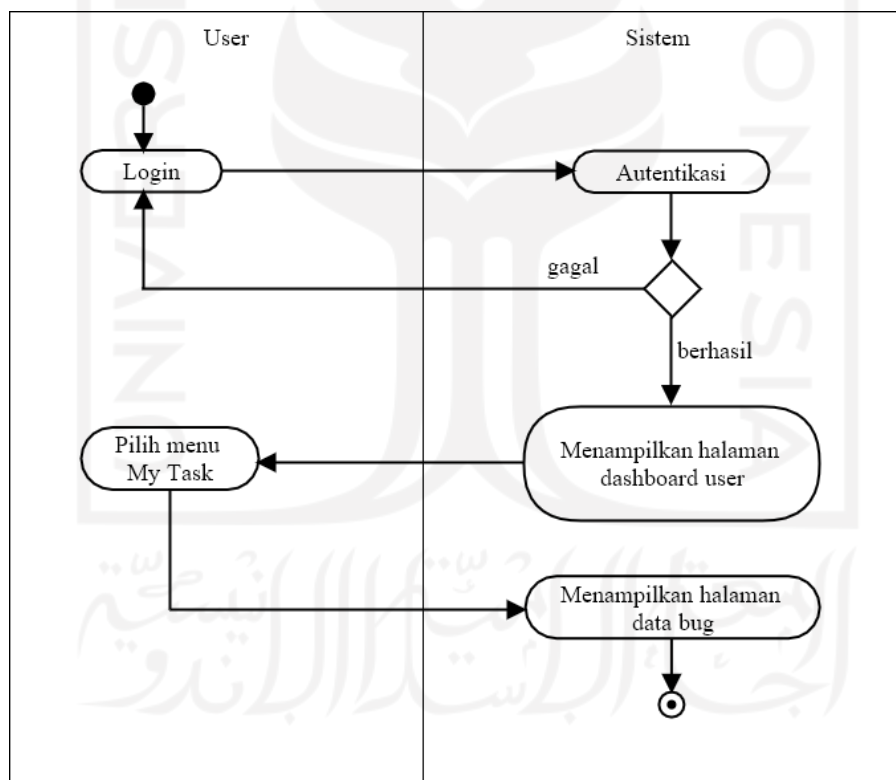
sistem akan menampilkan halaman data *bug*. Pada halaman tersebut terdapat tombol *Add Bug*, *user* menekan tombol tersebut, lalu sistem akan menampilkan halaman *form* tambah data *bug*. *User* melakukan pengisian *form* tambah data *bug* lalu menekan tombol *Save* bila sudah selesai. Sistem akan menyimpan data *bug* baru tersebut kemudian menampilkan halaman data *bug* dengan data yang terbaru. Aktivitas ini seperti pada gambar 3.16 berikut ini.



Gambar 3. 16 Activity diagram menambah data *bug*

3.4.14 Activity Diagram Melihat Data Bug

Aktivitas melihat data *bug* ini merupakan kegiatan untuk mengakses daftar data *bug* yang ada pada sistem. Semua *user* dapat melihat daftar data *bug* pada sistem, namun ada sedikit perbedaan antara data *bug* yang ditampilkan pada akun milik admin dengan data *bug* yang ditampilkan pada *user* lainnya. Admin dapat melihat semua daftar data *bug* yang telah dimasukkan pada sistem, sedangkan *user* lain hanya dapat melihat daftar data *bug* yang melibatkan dirinya saja. Untuk melihat data *bug*, *user* harus melakukan *login* terlebih dahulu. *User* memasukkan *email* dan *password* yang sudah terdaftar sebagai pada *google*. Jika gagal, maka *user* akan tetap berada pada halaman *login*. Jika berhasil, maka sistem akan menampilkan *dashboard user*. *User* memilih menu *My Task* pada bagian kiri halaman atau *navbar*, lalu sistem akan menampilkan halaman data *bug*. Aktivitas ini divisualkan dengan gambar 3.17 berikut ini.

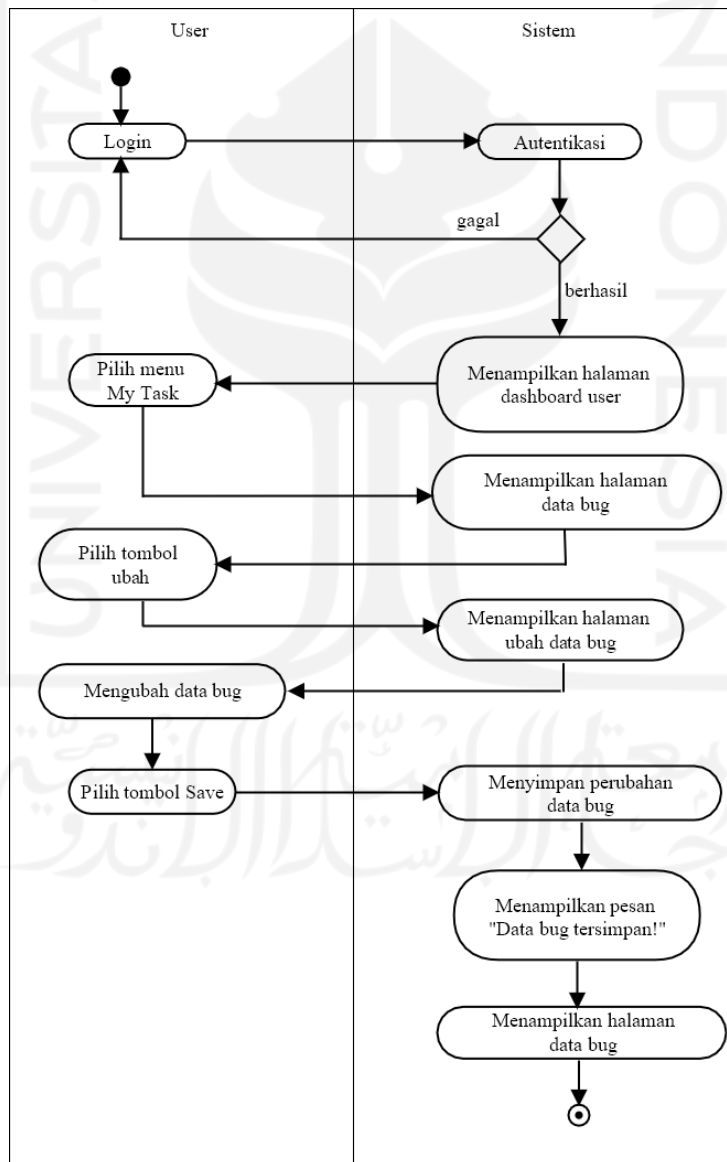


Gambar 3. 17 Activity diagram melihat data bug

3.4.15 Activity Diagram Mengubah Data Bug

Aktivitas ubah data *bug* merupakan kegiatan yang dilakukan untuk mengubah data *bug* yang sudah dimasukkan pada sistem. Kegiatan perubahan data *bug* ini dapat dilakukan oleh *Project Manager*, *Project Manager Assistant*, dan *Tester*. Pertama *user* melakukan *login*

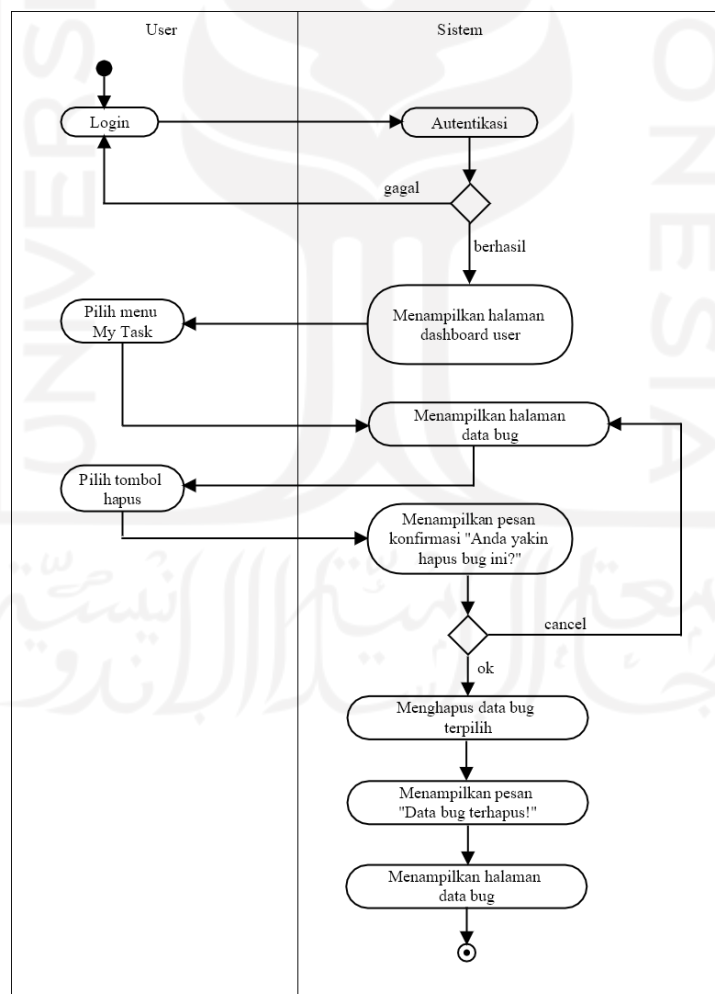
dengan akun *google*. Jika gagal, maka *user* akan tetap berada di halaman *login*. Jika *login* berhasil, maka sistem akan menampilkan *dashboard user*. Kemudian *user* memilih menu *My Task* sehingga sistem akan menampilkan halaman data *bug*. Pada halaman tersebut terdapat tombol ubah, pilih tombol ubah yang berada pada kolom *action* dan sejajar dengan data *bug* yang ingin diubah, lalu sistem akan menampilkan halaman *form* ubah data *bug*. Lakukanlah perubahan terhadap data *bug* lalu tekan tombol *Save* bila suda selesai. Sistem akan menyimpan perubahan data lalu menampilkan pesan “Data bug tersimpan!”. Selanjutnya sistem akan menampilkan kembali halaman data *bug* namun dengan data yang terbaru. Aktivitas ini sebagaimana divisualkan seperti gambar 3.18 berikut ini.



Gambar 3. 18 Activity diagram mengubah data bug

3.4.16 Activity Diagram Menghapus Data Bug

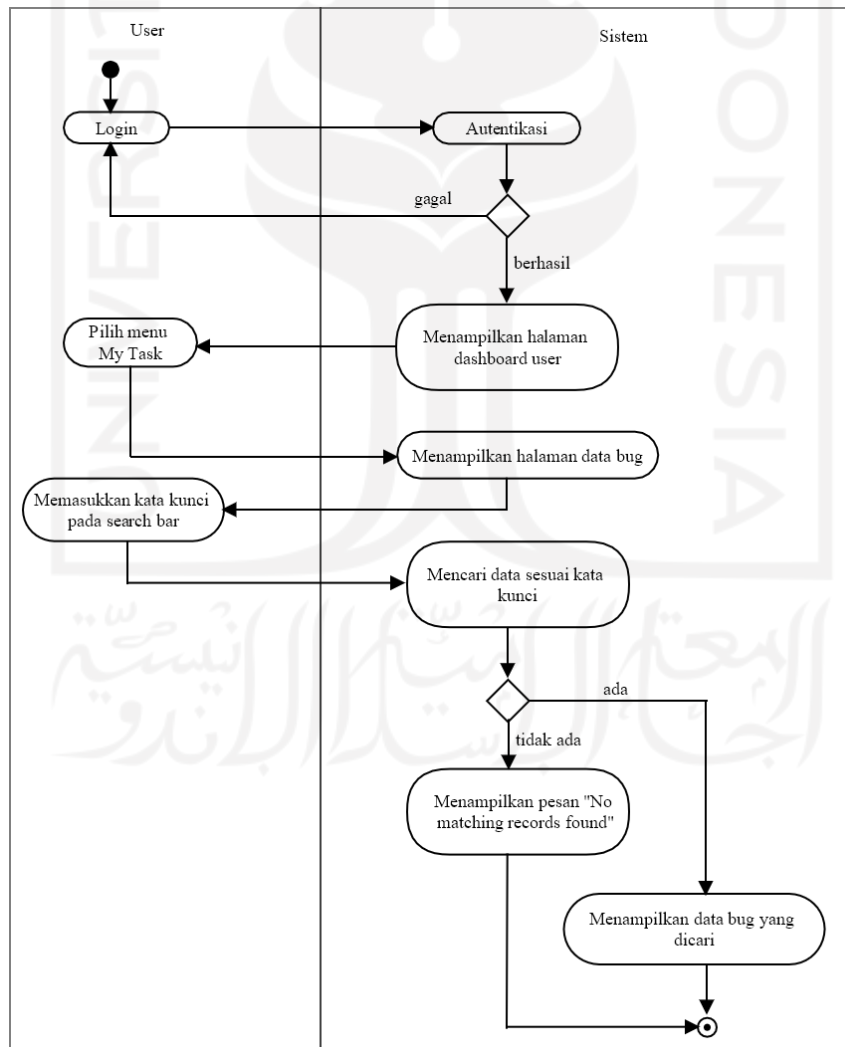
Aktivitas penghapusan data *bug* ini hanya dapat dilakukan oleh *Project Manager*, *Project Manager Assistant*, dan *Tester* yang diawali dengan *login* terlebih dahulu menggunakan akun *google*. Jika *login* gagal, maka *user* akan tetap berada pada halaman *login*. Jika *login* berhasil, maka sistem akan menampilkan halaman *dashboard user*. Kemudian, *user* memilih menu *My Task* sehingga sistem akan menampilkan halaman data *bug*. Pada halaman tersebut terdapat tombol hapus berwarna merah pada kolom *action*. Pilih tombol hapus sesuai dengan baris data *bug* yang ingin dihapus. Sistem akan menampilkan pesan konfirmasi penghapusan. Jika *user* memilih *cancel*, maka sistem akan kembali menampilkan halaman data *bug*. Jika *user* memilih *OK*, maka sistem akan menghapus data yang dimaksud lalu menampilkan halaman data *bug* dengan data yang terbaru. Aktivitas ini seperti gambar 3.19 berikut ini.



Gambar 3. 19 Activity diagram menghapus data bug

3.4.17 Activity Diagram Mencari Data Bug

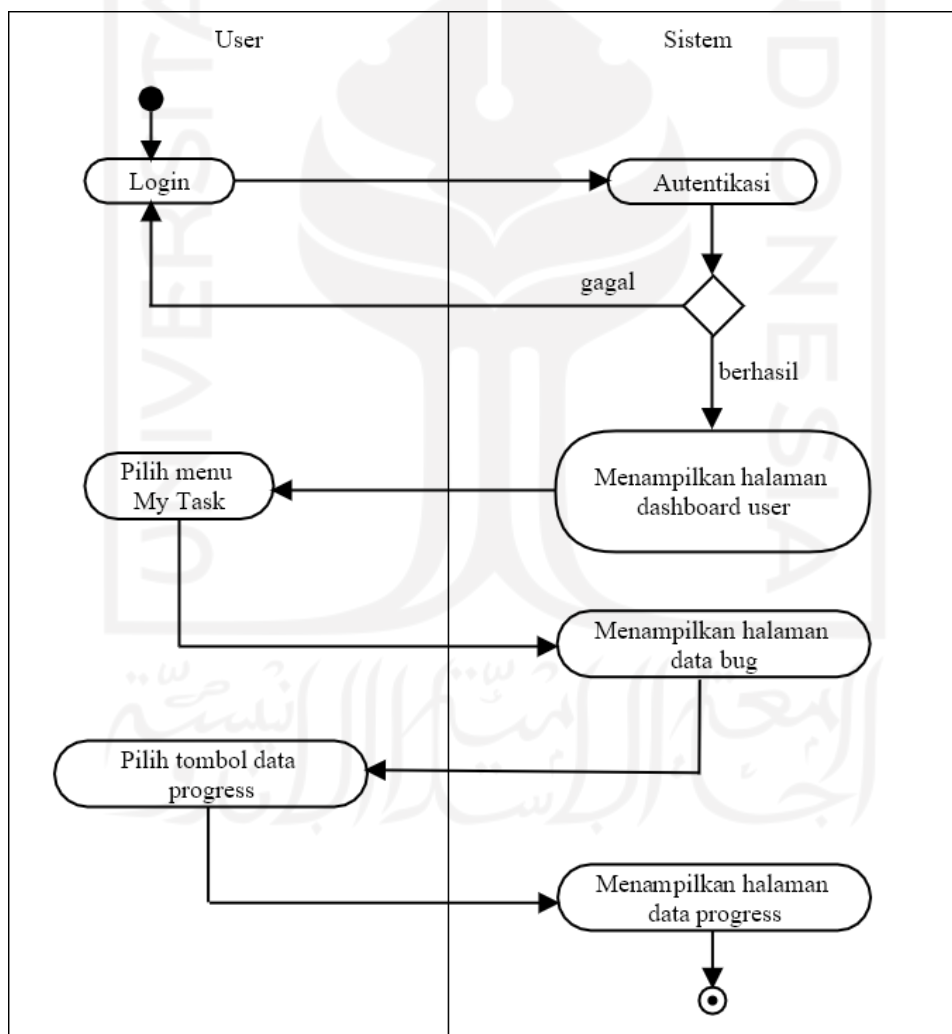
Aktivitas pencarian data *bug* ini dapat dilakukan oleh setiap *user*. Untuk mencari data *bug*, *user* harus melakukan *login* terlebih dahulu. Jika *login* gagal, maka *user* akan tetap berada pada halaman *login*. Jika *login* berhasil, maka *user* akan masuk ke sistem dan sistem akan menampilkan halaman *dashboard user*. *User* memilih menu *My Task* pada bagian *navbar*, lalu sistem akan menampilkan halaman data *bug*. Pada halaman tersebut terdapat *search bar*. Masukkan kata kunci pada *search bar*. Sistem akan mencari data *bug* sesuai dengan kata kunci yang dimasukkan. Jika data yang dicari tidak ada, maka sistem akan menampilkan pesan “*No matching records found.*”. Jika data yang dimaksud ada, maka sistem akan menampilkan data tersebut. Aktivitas ini divisualkan seperti gambar 3.20 berikut ini.



Gambar 3. 20 Activity diagram mencari data bug

3.4.18 Activity Diagram Melihat Data Progres Bug

Aktivitas melihat data *progress bug* merupakan kegiatan pengaksesan data progres suatu *bug* pada sistem. Semua *user* dapat melakukan kegiatan ini kecuali admin. Untuk melihat data *progress bug*, *user* harus berhasil melewati tahap *login* terlebih dahulu. Jika gagal, *user* akan tetap berada pada halaman *login*. Jika *login* berhasil, maka *user* berhasil masuk ke sistem dan sistem akan menampilkan halaman *dashboard user*. Pilih menu *My Task* sehingga sistem dapat menampilkan halaman data *bug*. Pada halaman tersebut terdapat tombol *progress* berwarna biru muda di kolom *action*. Pilih tombol *progress* yang sebaris dengan data *bug* yang ingin dilihat data progresnya. Selanjutnya sistem akan menampilkan halaman data *progress bug* yang dituju. Aktivitas ini divisualkan dengan gambar 3.21 berikut ini.



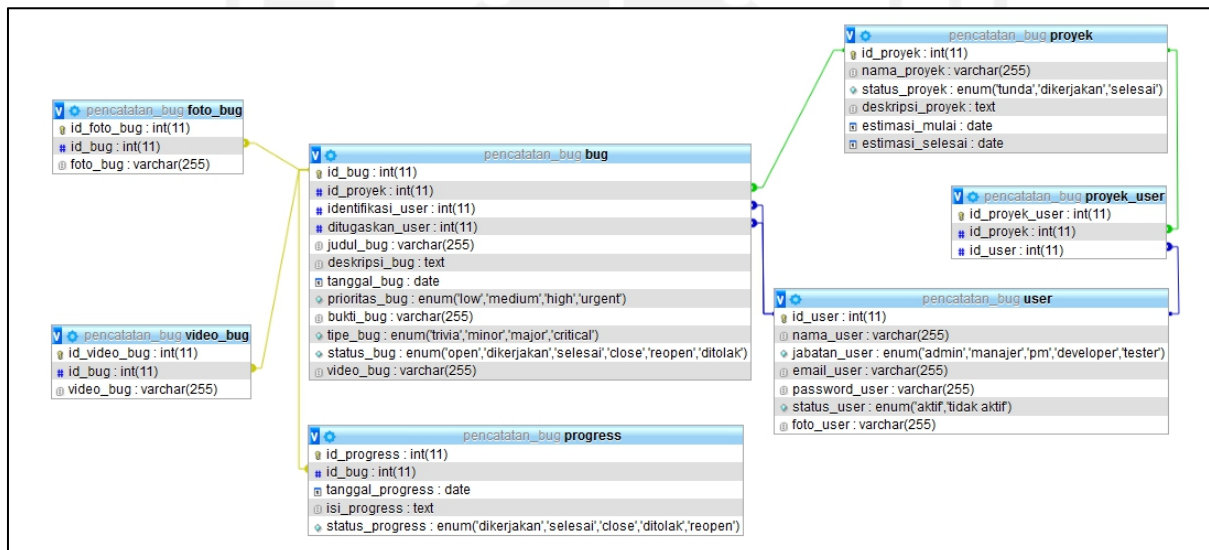
Gambar 3. 21 Activity diagram melihat data progress bug

3.5 Rancangan Basis Data

Rancangan basis data ini merupakan sebuah rancangan dari penyimpanan semua data yang nantinya akan dimasukkan ke dalam Sistem Informasi Pencatatan *Bug*. Berikut ini merupakan hal-hal yang berkaitan dengan basis data yang akan digunakan pada Sistem Informasi Pencatatan *Bug*.

3.5.1 Relasi Tabel

Relasi tabel merupakan relasi atau hubungan antara tabel yang satu dengan yang lain pada *database*. Pada *database* sistem terdapat tujuh buah tabel yang berkaitan satu sama lainnya. Ketujuh tabel tersebut ialah tabel *bug*, *progress*, *foto_bug*, *video_bug*, *proyek*, *proyek_user*, dan *user*. Relasi tiap tabel tersebut divisualkan seperti pada Gambar 3.22 berikut.



Gambar 3. 22 Relasi tabel

3.5.2 Struktur Tabel

a. Struktur Tabel *Bug*

Tabel *bug* digunakan untuk menyimpan data-data *bug* yang dimasukkan ke dalam sistem. Tabel ini terdiri dari 12 kolom yaitu kolom `id_bug`, `id_proyek`, `identifikasi_user`, `ditugaskan_user`, `judul_bug`, `deskripsi_bug`, `tanggal_bug`, `prioritas_bug`, `bukti_bug`, `tipe_bug`, `status_bug`, dan `video_bug` seperti pada Tabel 3.1 berikut ini.

Tabel 3. 1 Struktur tabel *bug*

Nama Kolom	Tipe Data	Keterangan
id_bug	int(11)	<i>Primary Key</i>
id_proyek	int(11)	
identifikasi_user	int(11)	
ditugaskan_user	int(11)	
judul_bug	varchar(255)	
deskripsi_bug	text	
tanggal_bug	date	
prioritas_bug	enum('low', 'medium', 'high', 'urgent')	
bukti_bug	varchar(255)	
tipe_bug	enum('trivia', 'minor', 'major', 'critical')	
status_bug	enum('open', 'dikerjakan', 'selesai', 'close', 'reopen', 'ditolak')	
video_bug	varchar(255)	

b. Struktur Tabel *Progress*

Tabel *progress* yang terlihat seperti pada tabel 3.2 ini digunakan untuk menyimpan data-data *progress* dari pengerjaan suatu *bug*. Tabel ini terdiri dari lima kolom yaitu kolom id_progress, id_bug, tanggal_progress, isi_progress, dan status_progress.

Tabel 3. 2 Struktur tabel *progress*

Nama Kolom	Tipe Data	Keterangan
id_progress	int(11)	<i>Primary Key</i>
id_bug	int(11)	
tanggal_progress	date	
isi_progress	text	
status_progress	enum('dikerjakan', 'selesai', 'close', 'ditolak', 'reopen')	

c. Struktur Tabel Proyek

Tabel 3.3 berikut ini merupakan tabel proyek yang digunakan untuk menyimpan data-data proyek yang *user* masukkan ke sistem. Tabel ini terdiri dari enam kolom yaitu kolom *id_proyek*, *nama_proyek*, *status_proyek*, *deskripsi_proyek*, *estimasi_mulai*, dan *estimasi_selesai*.

Tabel 3. 3 Struktur tabel proyek

Nama Kolom	Tipe Data	Keterangan
<i>id_proyek</i>	int(11)	<i>Primary Key</i>
<i>nama_proyek</i>	varchar(255)	
<i>status_proyek</i>	enum('tunda', 'dikerjakan', 'selesai')	
<i>deskripsi_proyek</i>	text	
<i>estimasi_mulai</i>	date	
<i>estimasi_selesai</i>	date	

d. Struktur Tabel Proyek *User*

Tabel proyek *user* merupakan tabel yang digunakan untuk menyimpan data keterlibatan seorang *user* dengan sebuah proyek. Tabel ini terdiri dari tiga kolom yaitu kolom *id_proyek_user*, *id_proyek*, dan *id_user* seperti pada tabel 3.4 berikut ini.

Tabel 3. 4 Struktur tabel proyek *user*

Nama Kolom	Tipe Data	Keterangan
<i>id_proyek_user</i>	int(11)	<i>Primary Key</i>
<i>id_proyek</i>	int(11)	
<i>id_user</i>	int(11)	

e. Struktur Tabel *User*

Tabel 3.5 berikut ini digunakan untuk menyimpan data-data yang terkait dengan *user*. Tabel *user* ini terdiri dari tujuh kolom yaitu kolom *id_user*, *nama_user*, *jabatan_user*, *email_user*, *password_user*, *status_user*, serta *foto_user*.

Tabel 3. 5 Struktur tabel *user*

Nama Kolom	Tipe Data	Keterangan
id_user	int(11)	<i>Primary Key</i>
nama_user	varchar(255)	
jabatan_user	enum('admin', 'manajer', 'pm', 'developer', 'tester')	
email_user	varchar(255)	
password_user	varchar(255)	
status_user	enum('aktif', 'tidak aktif')	
foto_user	varchar(255)	

f. Struktur Tabel Foto *Bug*

Tabel 3.6 berikut ini digunakan untuk menyimpan data-data yang terkait dengan foto *bug*. Tabel *foto_bug* ini terdiri dari tiga kolom yaitu kolom id_foto_bug, id_bug, dan foto_bug.

Tabel 3. 6 Struktur tabel *foto_bug*

Nama Kolom	Tipe Data	Keterangan
id_foto_bug	int(11)	<i>Primary Key</i>
id_bug	int(11)	
foto_bug	varchar(255)	

g. Struktur Tabel Video *Bug*

Tabel 3.7 berikut ini digunakan untuk menyimpan data-data yang terkait dengan *video bug*. Tabel *video_bug* ini terdiri dari tiga kolom yaitu kolom id_video_bug, id_bug, dan video_bug.

Tabel 3. 7 Struktur tabel *video_bug*

Nama Kolom	Tipe Data	Keterangan
id_video_bug	int(11)	<i>Primary Key</i>
id_bug	int(11)	
video_bug	varchar(255)	

3.6 Rancangan Antar Muka

Rancangan antar muka atau *interface design* merupakan gambaran dari rancangan tampilan sistem yang akan dibuat nanti. Rancangan antar muka ini juga dapat dijadikan sebagai patokan *design* saat proses pembuatan *web* nantinya. Berikut ini merupakan rancangan antar muka dari Sistem Informasi Pencatatan Bug Berbasis *Web*.

3.6.1 Halaman *Login*

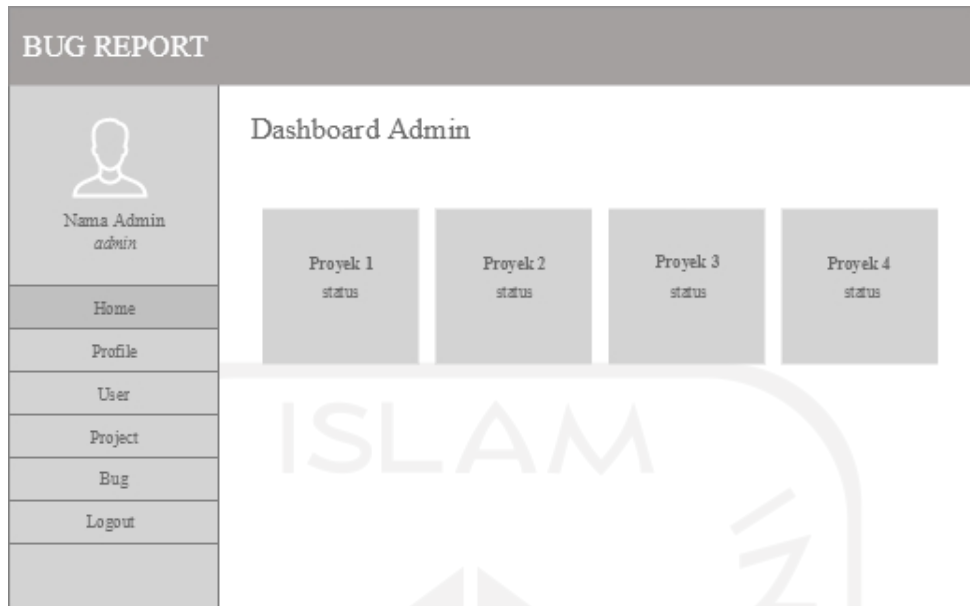
Halaman *login* merupakan halaman pertama yang akan dijumpai tiap *user*. Untuk masuk ke dalam sistem, tiap *user* harus melakukan proses *login* pada halaman ini terlebih dahulu. *User* melakukan *login* dengan memasukkan alamat *email* yang sudah terdaftar sebagai akun *Google* dan memasukkan *password* yang sesuai. Rancangan antar muka halaman *login* terlihat pada Gambar 3. 23 berikut ini.



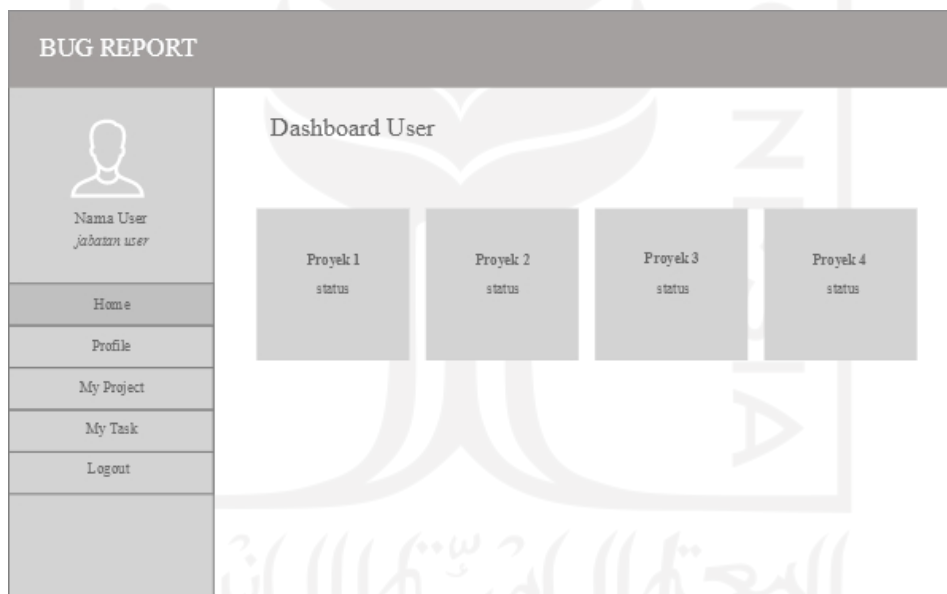
Gambar 3. 23 Rancangan antar muka halaman *login*

3.6.2 Halaman Utama (*Dashboard*)

Halaman *dashboard* merupakan halaman utama atau halaman pertama yang akan muncul setelah *user* berhasil melakukan proses *login* dan masuk ke dalam sistem. Pada halaman utama ini, terdapat daftar proyek, foto *user*, nama *user*, jabatan *user*, serta beberapa menu pilihan di bagian *navbar*. Untuk menu pilihan, terdapat sedikit perbedaan antara admin dengan *user* lainnya. Pada tampilan admin, terdapat enam pilihan menu yaitu *home*, *profile*, *user*, *project*, *bug*, dan *logout*. Sedangkan pada tampilan pengguna lain, menu *user* tidak ada. Gambar 3.24 merupakan tampilan rancangan antar muka *dashboard* untuk admin dan Gambar 3.25 ialah tampilan *dashboard user* lain selain admin.



Gambar 3. 24 Rancangan antar muka halaman *dashboard* Admin



Gambar 3. 25 Rancangan antar muka halaman *dashboard user* selain Admin

3.6.3 Halaman Profil

Halaman profil akan menampilkan data profil *user* seperti nama, *email*, dan foto. Pada halaman ini pula tiap *user* dapat melakukan perubahan pada data profilnya dengan menggunakan tombol *Change Profile*. Gambar 3.26 berikut ini merupakan rancangan antar muka untuk halaman data profil.

BUG REPORT		
Nama User jabatan user Home Profile User Project Bug Logout	Data Profil	
	Nama	
	Email	
	Foto	
	Change Profile	

Gambar 3. 26 Rancangan antar muka halaman data profil

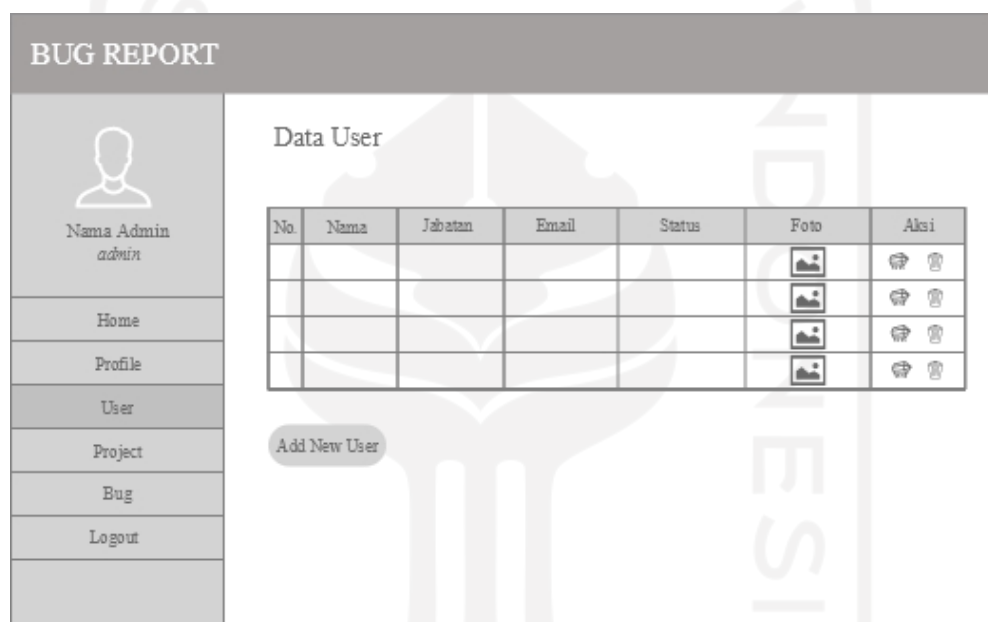
Jika *user* ingin melakukan perubahan pada data profilnya, *user* akan dibawa ke halaman ubah profil setelah menekan tombol *Change Profile*. Pada halaman ubah profil, *user* dapat mengubah nama, *email*, dan fotonya. Rancangan antar muka halaman ubah profil ditunjukkan seperti Gambar 3.27 berikut ini.

BUG REPORT		
Nama User jabatan user Home Profile User Project Bug Logout	Ubah Profil	
	Nama	<input type="text" value="Nama Admin"/>
	Email	<input type="text" value="Email Admin"/>
	Foto	<input type="button" value="browse"/>
	<input type="button" value="Save"/> <input type="button" value="Back"/>	

Gambar 3. 27 Rancangan antar muka halaman ubah profil

3.6.4 Halaman Data *User Admin*

Halaman *user* ini hanya dapat diakses oleh admin. Dengan demikian hanya adminlah yang dapat mengelola data *user*. Terdapat empat kegiatan yang dapat dilakukan admin dalam mengelola data *user*, yaitu menambah, melihat, mengubah, dan menghapus data. Pada halaman data *user*, terdapat sebuah tabel dengan tujuh kolom yaitu kolom nomor, nama, jabatan, *email*, status, foto, dan kolom aksi yang memuat tombol ubah dan hapus data. Tombol tambah data, ubah data, dan hapus data juga berada pada halaman ini. Gambar 3.28 berikut ini merupakan gambar rancangan antar muka dari halaman data *user*.



Gambar 3. 28 Rancangan antar muka halaman data *user*

Gambar 3.29 berikut ini merupakan gambar dari halaman tambah *user*. Admin akan memasukkan data *user* baru pada halaman ini. Data yang perlu admin masukkan berupa nama, jabatan, *email*, status, dan foto. Pilihan jabatan dibuat dalam bentuk *dropdown* yang terdiri dari lima pilihan, yaitu *project manager*, *project manager assistant*, admin, *tester*, dan *developer*. Sama halnya dengan status, namun pada status hanya terdapat dua pilihan, yaitu aktif dan tidak aktif.

BUG REPORT

Tambah User

Nama

Jabatan

Email

Status

Foto

Nama Admin
admin

Home

Profile

User

Project

Bug

Logout

Gambar 3. 29 Rancangan antar muka halaman tambah *user*

Untuk halaman ubah data *user* tidak jauh berbeda dengan halaman tambah *user*. Bedanya hanya tiap *field* sudah terisi dengan data *user* yang sudah dimasukkan sebelumnya. Untuk menyimpan perubahan, admin harus menekan tombol *Save* yang ada pada bagian bawah halaman ubah data *user*. Gambar 3.30 berikut ini merupakan rancangan antar muka halaman ubah data *user*.

BUG REPORT

Ubah Data User

Nama

Jabatan

Email

Status

Foto

Nama Admin
admin

Home

Profile

User

Project

Bug

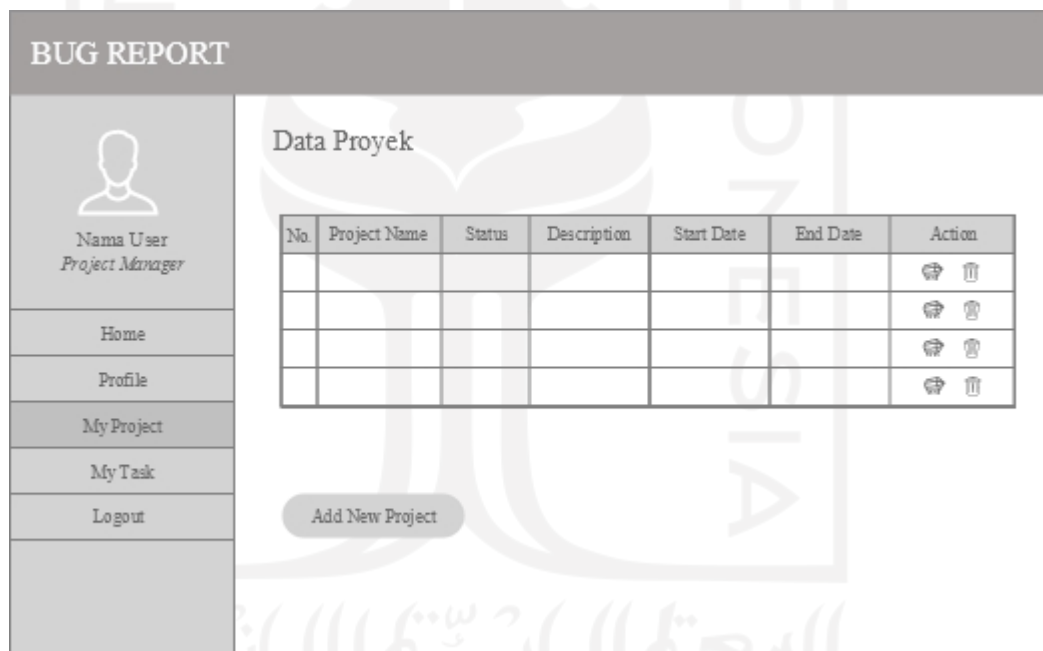
Logout

Gambar 3. 30 Rancangan antar muka halaman ubah data *user*

3.6.5 Halaman Data Proyek

a. Halaman Data Proyek *Project Manager*

Halaman data proyek untuk *Project Manager* berbeda dengan *user* lainnya. Pada halaman ini sistem akan menampilkan data proyek dalam bentuk tabel yang terdiri dari tujuh kolom, yaitu nomor, *project name*, status, *description*, *start date*, *end date*, dan *action*. Seorang *Project Manager* dapat melakukan penambahan data proyek dengan menekan tombol *Add New Project* pada bagian bawah tabel. Selain menambah data, *Project Manager* juga dapat melakukan perubahan serta penghapusan data dengan menekan tombol ubah dan hapus pada kolom *action*. Rancangan antar muka halaman data proyek untuk *Project Manager* ini divisualkan seperti Gambar 3.31 berikut ini.



Gambar 3. 31 Rancangan antar muka halaman data proyek *Project Manager*

Untuk menambah data proyek, sistem akan menampilkan halaman tambah data proyek setelah *Project Manager* menekan tombol *Add New Project* pada halaman data proyek. Pada halaman tambah data proyek ini, *Project Manager* diminta untuk memasukkan data proyek baru berupa *project name*, status, *description*, *start date*, *end date*, dan *user* yang terlibat. Bagian status dibuat dengan menu dropdown dengan tiga pilihan, yaitu *On Progress*, *On Hold*, dan *Completed*. Tombol *Save* yang terletak dibawah tabel berfungsi untuk menyimpan data yang telah dimasukkan.

Rancangan antar muka halaman tambah data proyek ini divisualkan seperti Gambar 3.32 berikut ini.

BUG REPORT

Tambah Data Proyek

Project Name

Status
 Pilih ▼

Description

Start Date

End Date

Select involved users :

Nama User (jabatan)

Nama User (jabatan)

Nama User (jabatan)

Nama User (jabatan)

Save Back

Gambar 3. 32 Rancangan antar muka halaman tambah data proyek

Halaman ubah data proyek tidak berbeda jauh dengan halaman tambah data proyek. Perbedaannya hanya terletak pada *fields* yang sudah terisi ketika masuk ke halaman ini. Gambar 3.33 merupakan rancangan antar muka untuk halaman ubah data proyek.

BUG REPORT

Ubah Data Proyek

Project Name

Status
 Status Proyek ▼

Description

Start Date

End Date

Select involved users :

Nama User (jabatan)

Nama User (jabatan)

Nama User (jabatan)

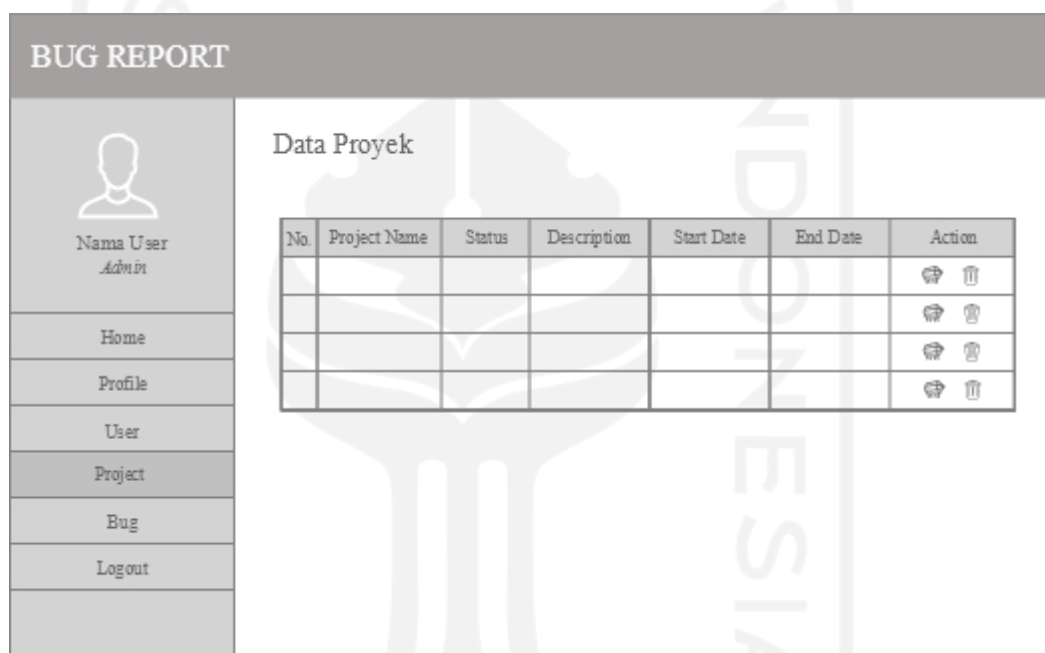
Nama User (jabatan)

Save Back

Gambar 3. 33 Rancangan antar muka halaman ubah data proyek

b. Halaman Data Proyek Admin


Untuk admin menu proyek dapat menampilkan semua data proyek yang sudah tercatat pada sistem. Tampilannya sedikit berbeda dari tampilan milik *Project Manager*, tidak ada tombol *Add New Project* di halaman ini sesuai dengan peran admin yang tidak bisa menambahkan data proyek baru. Sedangkan untuk halaman ubah data proyeknya sama saja seperti tampilan halaman ubah data proyek pada *Project Manager*. Halaman data proyek untuk admin divisualkan dalam Gambar 3.34 berikut ini.



Gambar 3. 34 Rancangan antar muka halaman data proyek Admin

c. Halaman Data Proyek *Project Manager Assistant*, *Tester*, dan *Developer*

Untuk *Project Manager*, *Tester*, dan *Developer*, menu proyek hanya dapat menampilkan data proyek yang melibatkan dirinya saja. Sistem hanya akan menampilkan tabel data proyek pada halaman ini. Tabelnya juga berbeda, tidak ada kolom aksi pada tabel di halaman ini. Pada halaman ini juga tidak terdapat tombol *Add New Project* seperti pada tampilan halaman data proyek *Project Manager*. Halaman data proyek untuk Proyek *Project Manager Assistant*, *Tester*, dan *Developer* divisualkan dalam Gambar 3.35 berikut ini.

BUG REPORT						
 Nama User <i>jabatan user</i>	Data Proyek					
	No.	Project Name	Status	Description	Start Date	End Date
Home						
Profile						
My Project						
My Task						
Logout						

Gambar 3. 35 Rancangan antar muka halaman data proyek *Project Manager Assistant, Tester, dan Developer*

3.6.6 Halaman Data *Bug*


Halaman data *bug* berisikan tabel data *bug* yang terdiri dari 11 kolom, yaitu nomor, *bug title*, *description*, *identified date*, *evidence*, *priority*, *identified by*, *status*, *type*, *project name*, dan *assigned to*. Kolom nomor berisikan nomor urutan masuknya data *bug*, nomor ini akan menjadi *id bug*. Kolom *bug title* berisi judul dari *bug*. Kolom *description* berisi mengenai penjelasan atau kronologi bagaimana hingga *bug* ditemukan dan apa yang terjadi. *Identified date* merupakan kolom yang berisi tanggal muncul atau ditemukannya *bug*. Kolom *evidence* berisi bukti telah munculnya *bug*, dapat berupa foto, *screenshot*, atau video. Kolom *type* berisi tipe dari *bug*. Terdapat empat tipe *bug* yaitu *trivia*, *minor*, *major*, dan *critical*. Kolom *priority* berisi mengenai prioritas pengerjaan *bug*, dapat berupa *low*, *medium*, *high*, atau *urgent*. Kolom *identified by* berisi nama *user* yang menemukan *bug* atau yang menambahkan data *bug*. Kolom *status* berisi mengenai status pengerjaan *bug*, ada tiga pilihan yaitu tunda, dikerjakan, dan selesai. Kolom *project name* berisikan nama proyek dimana *bug* ditemukan. Kolom *assigned to* berisikan nama *user* yang ditugaskan untuk memperbaiki *bug*. Rancangan tabel *bug* tersebut divisualkan seperti pada Gambar 3.36 berikut ini.

No.	Bug Title	Description	Identified Date	Evidence	Priority	Identified By	Status	Type	Project Name	Assigned To

Gambar 3. 36 Rancangan tabel *bug*

a. Halaman Data *Bug* Admin

Berdasarkan perannya, seorang admin hanya dapat melihat data *bug* tanpa dapat menambah, mengubah, ataupun menghapusnya. Pada halaman ini, sistem akan menampilkan tabel dengan 11 kolom, yaitu nomor, *bug title*, *description*, *identified date*, *evidence*, *priority*, *identified by*, *status*, *type*, *project name*, dan *assigned to*. Gambar 3.37 berikut ini merupakan gambar dari rancangan antar muka halaman data *bug* untuk admin.

BUG REPORT											
 Nama User <i>Admin</i>	Data Bug										
	No	Bug Title	Description	Identified Date	Evidence	Priority	Identified By	Status	Type	Project Name	Assigned To
Home											
Profil											
User											
Project											
Bug											
Logout											

Gambar 3. 37 Rancangan antar muka halaman data *bug* Admin

b. Halaman Data *Bug* *Project Manager*, *Project Manager Assistant* dan *Tester*

Halaman data *bug* untuk *Project Manager*, *Project Manager Assistant* dan *Tester* berbeda dengan *user* lainnya. Pada halaman ini sistem akan menampilkan data *bug* dalam bentuk tabel yang terdiri dari 12 kolom, yaitu nomor, *bug title*, *description*, *identified date*, *evidence*, *priority*, *identified by*, *status*, *type*, *project*

name, dan *assigned to*, dan *action*. Seorang *Project Manager*, *Project Manager Assistant* dan *Tester* dapat melakukan penambahan data *bug* dengan menekan tombol *Add Bug* pada bagian bawah tabel. Selain menambah data, ketiga *user* ini juga dapat melakukan perubahan serta penghapusan data dengan menekan tombol ubah dan hapus pada kolom *action*. Untuk melihat data progress bug *user* harus menekan tombol *progress* pada kolom *action*. Rancangan antar muka halaman data *bug* untuk *Project Manager*, *Project Manager Assistant* dan *Tester* ini divisualkan seperti Gambar 3.38 berikut ini.

No.	Bug Title	Description	Identified Date	Evidence	Priority	Identified By	Status	Type	Project Name	Assigned To	Action
											Progress
											Progress
											Progress
											Progress

Gambar 3. 38 Rancangan antar muka halaman data *bug* *Project Manager*, *Project Manager Assistant* dan *Tester*

Untuk menambah data *bug*, sistem akan menampilkan halaman tambah data *bug* setelah *Project Manager*, *Project Manager Assistant* atau *Tester* menekan tombol *Add Bug* pada halaman data *bug*. Pada halaman tambah data *bug* ini, ketiga *user* ini diminta untuk memasukkan data *bug* baru berupa *bug title*, *description*, *identified date*, *evidence*, *priority*, *type*, *project name*, dan *assigned to*. Tombol *Save* yang terletak dibawah tabel berfungsi untuk menyimpan data yang telah dimasukkan. Namun untuk tampilan pada *Tester* terdapat sedikit perbedaan, yaitu *Tester* tidak terdapat *field assigned to*. Rancangan antar muka halaman tambah data *bug* ini divisualkan seperti Gambar 3.39 berikut ini.

BUG REPORT

Tambah Data Bug

Nama User
jabatan user

Home

Profile

My Project

My Task

Logout

Bug Title

Description

Identified Date
nm/dd/yyyy

Type
Pilih

Priority
Pilih

Project Name
Pilih

Assigned To
Pilih

Evidence 1
browse

Evidence 2
browse

Save Back

Gambar 3. 39 Rancangan antar muka halaman tambah data *bug* *Project Manager*, *Project Manager Assistant* dan *Tester*

Halaman ubah data *bug* tidak berbeda jauh dengan halaman tambah data *bug*. Perbedaannya hanya terletak pada *fields* yang sudah terisi ketika masuk ke halaman ini. Gambar 3.40 merupakan rancangan antar muka untuk halaman ubah data *bug*.

BUG REPORT

Ubah Data Bug

Bug Title
Judul Bug

Description
Deskripsi bug

Identified Date
Tanggal ditemukannya bug (nm/dd/yyyy)

Type
tipe bug

Priority
prioritas bug

Project Name
Nama Proyek

Assigned To
ditugaskan kepada

Save Back

Gambar 3. 40 Rancangan antar muka halaman ubah data *bug Project Manager, Project Manager Assistant dan Tester*

Untuk melihat data *progress* sebuah *bug*, *user* harus menekan tombol *progress* yang berada pada kolom *action*. Gambar 3.41 merupakan rancangan antar muka untuk halaman data *progress bug*.

BUG REPORT

Data Progress

No.	Tanggal	Isi	Status

Gambar 3. 41 Rancangan antar muka halaman data *progress bug*

c. Halaman Data *Bug Developer*


Halaman data *bug* untuk *Developer* tidak jauh berbeda dengan *user* lainnya. Pada halaman ini sistem akan menampilkan data *bug* dalam bentuk tabel yang terdiri dari 12 kolom, yaitu nomor, *bug title*, *description*, *identified date*, *evidence*, *priority*, *identified by*, *status*, *type*, *project name*, dan *assigned to*, dan *action*. Seorang *Developer* hanya dapat melakukan perubahan data status *bug* dengan menekan tombol ubah dan melihat data *progress bug* dengan menekan tombol *progress* yang berada pada kolom *action*. Rancangan antar muka halaman data *bug* untuk *Developer* ini divisualkan seperti pada Gambar 3.42 berikut ini.

BUG REPORT											
 Nama User <i>developer</i>		Data Bug									
No	Bug Title	Description	Identified Date	Evidence	Priority	Identified By	Status	Type	Project Name	Assigned To	Aksi

Gambar 3. 42 Rancangan antar muka halaman data *bug Developer*

Halaman ubah data *bug Developer* ini lebih mengarah untuk perubahan status *bug* dan penambahan data *progress bug*. Terdapat dua *fields* yang harus diisi yaitu status yang disajikan dalam bentuk *dropdown* dan *field progress* untuk mengisi deskripsi atau keterangan . Gambar 3.43 berikut ini merupakan rancangan antar muka untuk halaman ubah data *bug Developer*.

Pencatatan Bug


Nama developer
developer

Home

Profil

Proyek

Catatan Bug

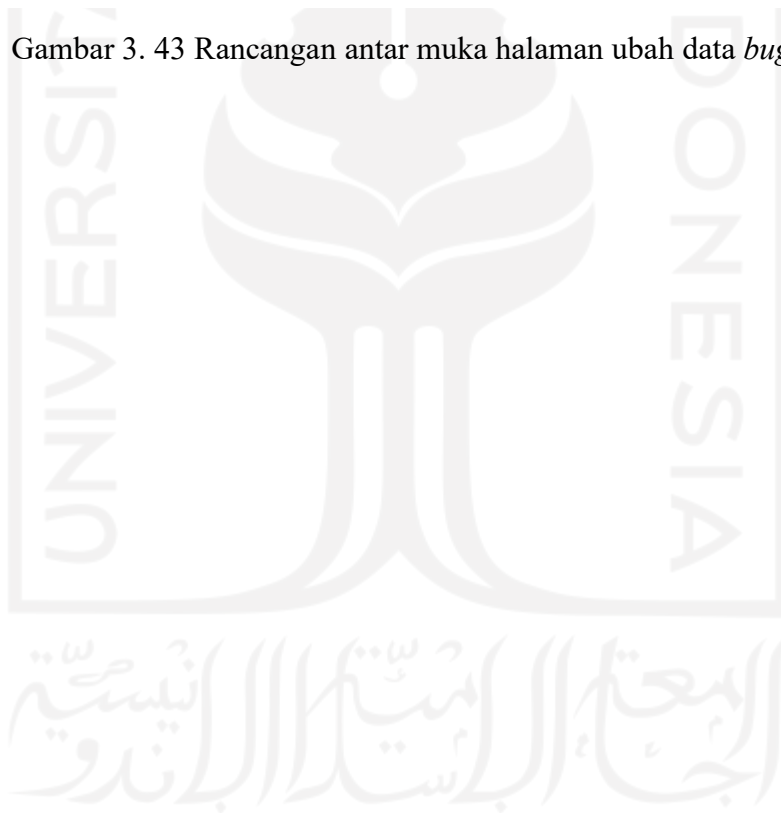
Logout

Ubah Data Bug

Status

Progress

Gambar 3. 43 Rancangan antar muka halaman ubah data *bug Developer*

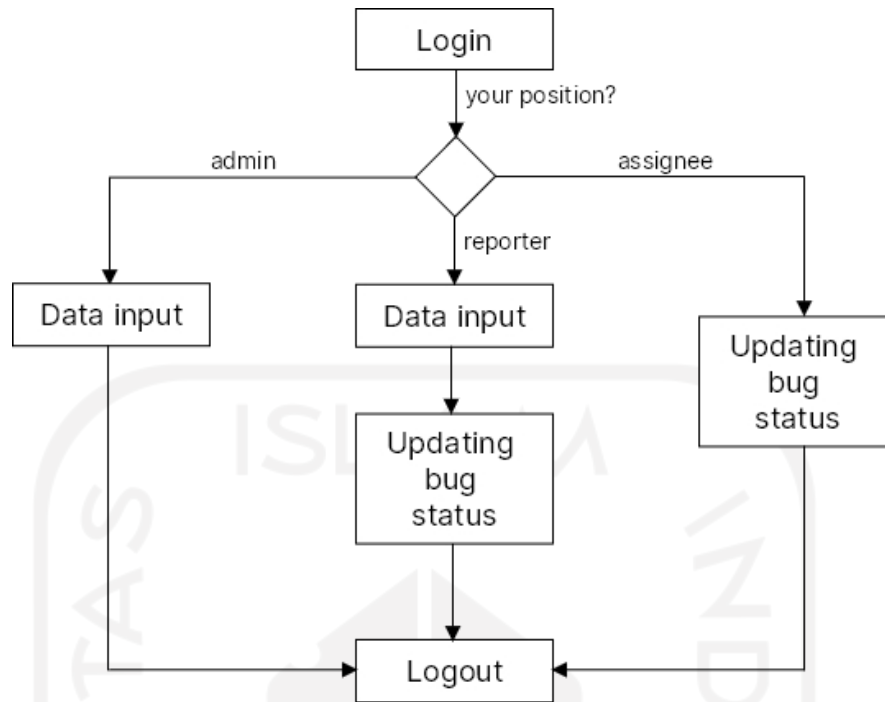


BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

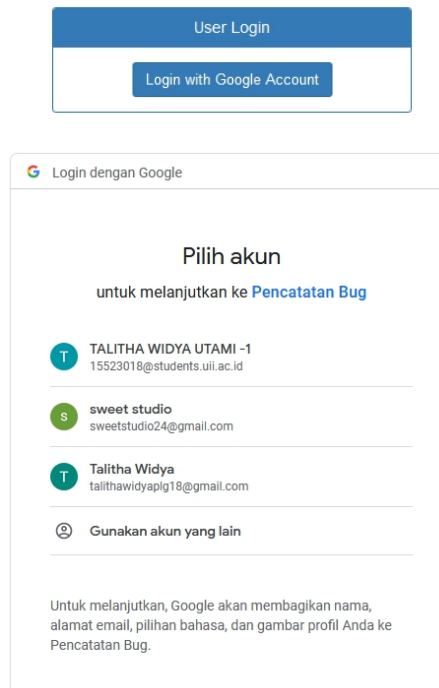
Sistem informasi pencatatan bug berbasis *web* ini rencananya akan digunakan oleh tim *Center of Data Science* Universitas Islam Indonesia (CDS UII) dikemudian hari. Maka dari itu, setelah proses pembangunan sistem selesai, dilakukan *web hosting* dengan alamat *web cdsbugbase.site* agar dapat diakses oleh semua anggota tim CDS UII yang telah terdaftar pada sistem. *Web* ini termasuk dalam kategori web dinamis karena penggunaannya akan diminta untuk mengisi tabel-tabel data pada halaman *web* dengan data-data yang mereka miliki sehingga *web CDS Bug Base* dapat menampilkan informasi yang rapi, lengkap, berguna dan *up-to-date*. Pengguna juga tidak akan merasa kebingungan saat diminta untuk mengisi tabel-tabel data tersebut karena pada *web* ini telah dilengkapi dengan informasi mengenai panduan pengisian pada tiap-tiap *field* pengisian datanya. Hal ini menunjukkan bahwa *web CDS Bug Base* merupakan *website* yang cukup interaktif. Setiap anggota tim CDS UII yang telah terdaftar di sistem harus berhasil melalui proses *login* pada halaman *login web* sistem untuk bisa masuk ke dalamnya. Setelah berhasil *login*, pengguna dapat melakukan berbagai aktivitas pada sistem sesuai dengan *previlage* atau perannya masing-masing. *Reporter* atau pelapor ialah pengguna yang memegang jabatan sebagai *Project Manager*, *Project Manager Assistant*, atau *Tester*. *Assignee* atau orang yang ditugaskan ialah pengguna yang memegang jabatan sebagai *Project Manager*, *Project Manager Assistant*, atau *Developer*. Setelah selesai melakukan aktivitas-aktivitas tersebut, pengguna dapat ke luar dari sistem dengan cara *logout*. Gambar 4.1 berikut merupakan representasi visual bagan alur sederhana sistem informasi pencatatan bug ini.



Gambar 4. 1 Bagan alur sistem

4.1.1 Halaman *Login User*

Halaman login user ialah halaman yang pertama kali akan ditemui oleh semua pengguna saat akan masuk ke sistem. Pengguna bisa masuk ke sistem jika sudah berhasil melalui proses login pada halaman ini. Login dilakukan dengan memasukkan email dan password akun yang sudah terdaftar sebagai akun *Google*. Halaman *login user* ditunjukkan seperti pada Gambar 4.2 berikut ini.

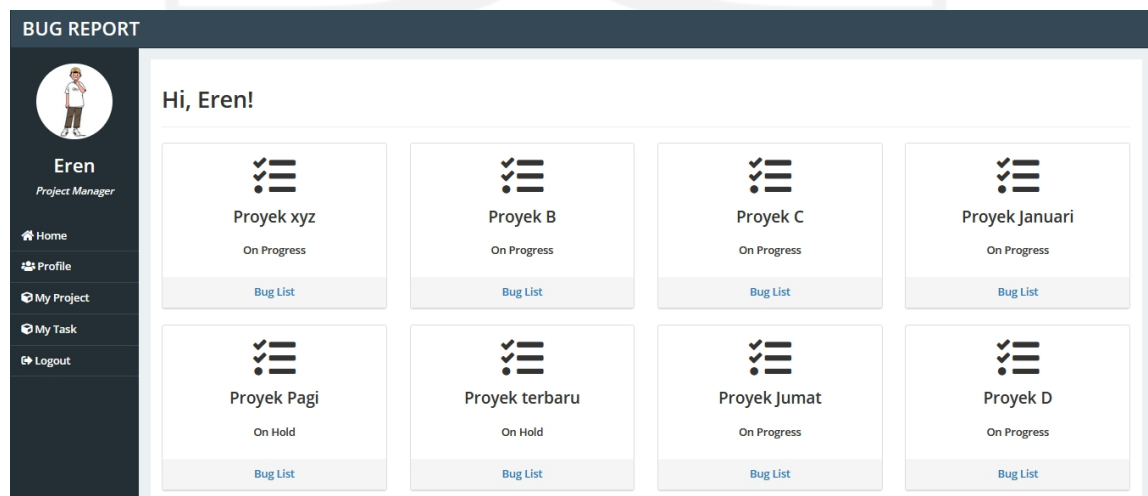


Gambar 4. 2 Halaman *login user*

4.1.2 Halaman Utama (*dashboard*)

- a. *Project Manager, Project Manager Assistant, Developer, Tester*

Setelah *Project Manager, Project Manager Assistant, Developer,* atau *Tester* berhasil melakukan *login*, sistem akan menampilkan halaman utama (*dashboard*). Sebagai salah satu contoh, gambar 4.3 berikut ini merupakan halaman utama milik seorang *Project Manager*.

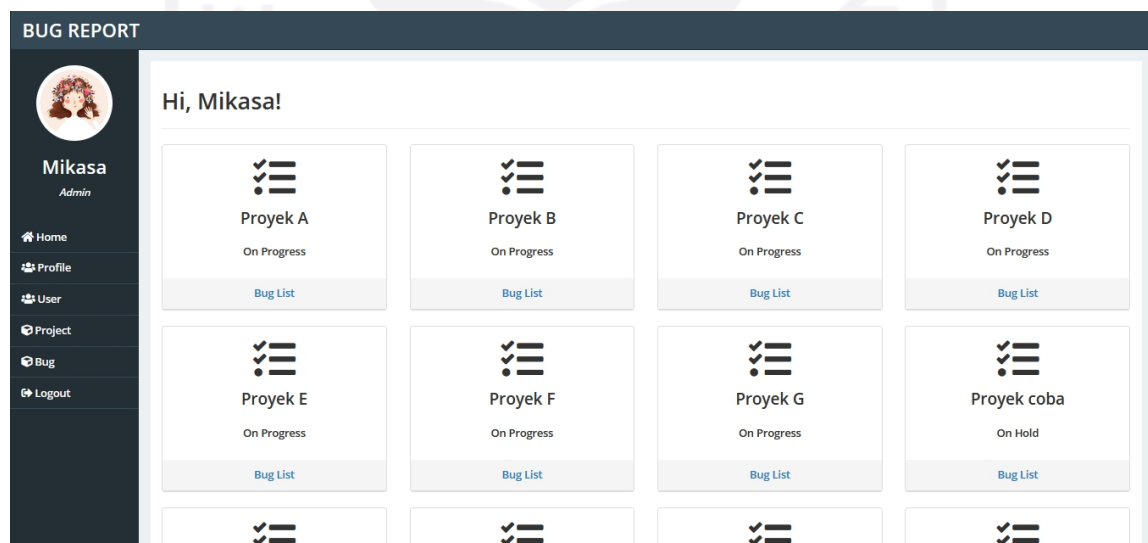


Gambar 4. 3 Halaman utama (*dashboard*) *Project Manager*

Pada sistem, terdapat kesamaan tampilan pada halaman utama (*dashboard*) milik *Project Manager*, *Project Manager Assistant*, *Developer*, dan *Tester*, yang membedakan hanya pada daftar proyek yang tampil, foto profil *user*, nama *user*, serta jabatan *user*. Pada bagian kiri atas terdapat nama sistem, kemudian di bawahnya terdapat foto profil pengguna, nama pengguna, serta jabatan pengguna. Di bagian kiri halaman ini juga terdapat lima menu yang dapat dipilih oleh pengguna, yaitu menu *home*, *profile*, *project*, *my task*, dan *logout*. Pada halaman utama ini, tidak semua daftar proyek akan ditampilkan, sistem hanya akan menampilkan daftar proyek yang mempunyai kaitan dengan pengguna saja. Daftar proyek ditampilkan dalam bentuk kotak atau panel dengan nama proyek, status proyek, dan *link* menuju ke halaman data *bug* proyek di dalamnya.

b. Admin

Halaman utama (*dashboard*) admin akan ditampilkan oleh sistem setelah seorang admin berhasil melakukan proses *login*. *Dashboard* admin ditampilkan seperti Gambar 4.4 berikut ini.



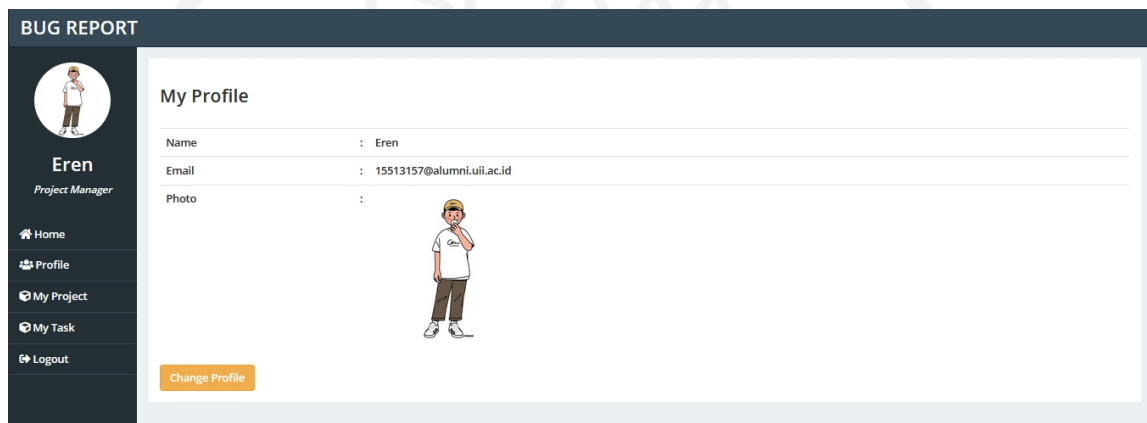
Gambar 4. 4 Halaman utama (*dashboard*) Admin

Pada bagian kiri halaman terdapat foto profil Admin, nama Admin, serta jabatan Admin. Kemudian di bawahnya terdapat enam menu yang dapat dipilih pengguna, yaitu menu *home*, *profile*, *user*, *project*, *bug*, dan *logout*. Pada halaman utama Admin ini, sistem akan menampilkan seluruh daftar proyek yang telah dimasukkan ke dalam sistem.

Daftar proyek ditampilkan dalam bentuk kotak atau panel dengan nama proyek, status proyek, dan *link* menuju ke halaman data *bug* proyek di dalamnya.

4.1.3 Halaman *Profile*

Saat pengguna menekan tombol menu *profile*, maka sistem akan memunculkan halaman data profil masing-masing pengguna. Sebagai contoh, gambar 4.5 berikut ini merupakan halaman data profil milik seorang *Project Manager*.



Gambar 4. 5 Halaman data profil *Project Manager*

Tampilan halaman data profil semua pengguna pada sistem sama saja seperti pada gambar 4.5 di atas, yang membedakan hanya pada isi datanya saja menyesuaikan dengan data tiap-tiap pengguna. Pada halaman data profil ini, pengguna akan disuguhkan dengan data profil pengguna, seperti nama pengguna, *email* pengguna, dan foto pengguna. Pengguna juga dapat melakukan perubahan nama, *email*, serta foto profilnya dengan menekan tombol berwarna *orange* bertuliskan *Change Profile* yang berada pada bagian bawah halaman data profil. Sebagai salah satu contoh, berikut ini merupakan tampilan dari halaman ubah data profil milik *Project Manager*.

The image shows a web interface for a 'Project Manager' user named 'Eren'. The page is titled 'BUG REPORT' at the top. On the left is a dark sidebar with a user profile picture and name 'Eren Project Manager'. Below the name are navigation links: Home, Profile, My Project, My Task, and Logout. The main content area is titled 'Change Profile' and contains three input fields: 'Name' (filled with 'Eren'), 'Email' (filled with '15513157@alumni.uii.ac.id'), and 'Photo' (containing a cartoon character and a 'Browse...' button with the text 'No file selected.'). At the bottom of the form are two buttons: 'Save' (dark blue) and 'Back' (light blue).

Gambar 4. 6 Halaman ubah data profil *Project Manager*


Tampilan dari halaman ubah data profil *Project Manager* seperti pada gambar 4.6 di atas sama saja seperti halaman ubah data profil pengguna lainnya. Jika pengguna telah selesai melakukan perubahan data, pengguna harus menekan tombol *Save* berwarna biru tua yang berada pada bagian bawah halaman ini untuk menyimpan perubahan data. Jika pengguna menekan tombol *Back* berwarna biru muda yang berada di samping tombol *Save*, maka sistem akan menuju ke halaman sebelumnya yaitu halaman data profil.

4.1.4 Halaman Data *User*

a. Halaman daftar semua *user*

Halaman daftar data *user* ini hanya akan dijumpai oleh pengguna dengan jabatan sebagai seorang Admin. Halaman *user* ditunjukkan seperti pada Gambar 4.7 berikut ini.

BUG REPORT
























Mikasa
Admin

- Home
- Profile
- User
- Project
- Bug
- Logout

All Users

Show entries Search:

No	Name	Position	Email	Status	Photo	Action
1	Mikasa	Admin	15523018@students.uui.ac.id	Aktif		 
2	Eren	Project Manager	15513157@alumni.uui.ac.id	Aktif		 
3	Sandy Cheeks	Project Manager Assistant	15320361@alumni.uui.ac.id	Aktif		 
4	Yura Chloe Ramadhan	Project Manager Assistant	15511129@alumni.uui.ac.id	Aktif		 
5	Belle	Tester	sweetstudio24@gmail.com	Aktif		 
6	Russel	Developer	talithawidyaplg18@gmail.com	Aktif		 
7	Squidward Tentacles	Developer	dianapratiwi24@gmail.com	Aktif		 

Showing 1 to 7 of 7 entries Previous **1** Next

[Add New User](#)

Gambar 4. 7 Halaman data semua *user*

Pada halaman ini, sistem akan menampilkan tabel data *user* yang terdiri dari tujuh kolom yaitu *no*, *name*, *position*, *email*, *status*, *photo*, dan *action*. Seorang Admin dapat melakukan perubahan dan penghapusan data *user* dengan menekan tombol yang berada pada kolom *action*. Untuk memudahkan pencarian data *user*, Admin dapat menggunakan *search bar* yang terletak pada bagian kanan atas tabel data *user*. Bagian *show entries* pada kiri atas tabel data *user* berguna agar Admin dapat menentukan berapa banyaknya data yang ingin ditampilkan dalam satu halaman. Jika daftar data melebihi jumlah data yang ditampilkan per satu halaman, maka Admin dapat melihatnya pada halaman selanjutnya dengan menekan tombol *next* pada bagian kanan bawah tabel data *user*. Untuk menambahkan data *user* baru, Admin dapat menekan tombol *Add New User* pada bagian kiri bawah tabel data *user*. Halaman tambah data *user* ditunjukkan seperti pada Gambar 4.8 berikut ini.

The screenshot shows a web interface for adding new user data. On the left is a dark sidebar with a user profile for 'Mikasa Admin' and a menu with items: Home, Profile, User, Project, Bug, and Logout. The main content area is titled 'Add New User Data' and contains the following form fields:

- Name:** A text input field.
- Position:** A dropdown menu with 'Pilih' (Select) as the current selection.
- Email:** A text input field.
- Status:** A dropdown menu with 'Pilih' (Select) as the current selection.
- Photo:** A file upload field with a 'Browse...' button and the text 'No file selected.'

At the bottom of the form are two buttons: a blue 'Save' button and a light blue 'Back' button.

Gambar 4. 8 Halaman tambah data *user*

Jika pengguna telah selesai melakukan pengisian data, pengguna harus menekan tombol *Save* berwarna biru tua yang berada pada bagian bawah halaman ini untuk menyimpan data baru. Jika pengguna menekan tombol *Back* berwarna biru muda yang berada di samping tombol *Save*, maka sistem akan menuju ke halaman sebelumnya yaitu halaman data semua *user*.

4.1.5 Halaman *Project*

a. *Project Manager*

Halaman data proyek akan muncul jika pengguna menekan menu *My Project* yang ada pada bagian *navbar*. Gambar 4.9 berikut ini merupakan tampilan dari halaman data proyek milik seorang *Project Manager*.

BUG REPORT

My Project

Show 10 entries Search:

No	Project Name	Status	Description	Start Date	End Date	Action
1	Proyek Xyz	On Progress	Cobaa aja	2021-01-16	2021-01-19	
2	Proyek B	On Progress	Yang ini juga cuma buat contoh aja	2021-01-17	2021-01-21	
3	Proyek C	On Progress	Ini waktu nyoba tombol ubah.	2021-01-17	2021-01-23	
4	Proyek Januari	On Progress	Asdf	2021-01-18	2021-01-29	
5	Proyek Pagi	On Hold	Coba proyek baru tapi statusnya tunda	2021-03-06	2021-03-31	
6	Proyek Terbaru	On Hold	Uji coba dari awal	2021-03-09	2021-03-31	
7	Proyek Jumat	On Progress	Deskripsi proyek	2021-03-12	2021-03-30	
8	Proyek D	On Progress	Ini waktu nyoba ubah statusnya dari tunda jadi dikerjakan	2021-03-16	2021-03-27	
9	Proyek E	On Progress	Ini nambahin lagi karna tadi nyoba tombol hapus.	2021-03-21	2021-03-26	

Showing 1 to 9 of 9 entries

[Previous](#) **1** [Next](#)

[Add New Project](#)

Gambar 4. 9 Halaman data proyek *Project Manager*

Sesuai dengan perannya, seorang *Project Manager* dapat melihat daftar data proyek, menambahkan, mengubah, serta menghapus data proyek pada halaman ini. Namun tidak semua data proyek yang terdaftar pada sistem yang akan ditampilkan, hanya proyek-proyek yang dimana ia terlibat saja. Sistem akan menampilkan tabel data proyek yang terdiri dari tujuh kolom, yaitu no, *project name*, status, *description*, *start date*, *end date* dan *action*. Kolom *action* berisi dua tombol yang dapat digunakan untuk mengubah dan menghapus data bug. Untuk memudahkan pencarian data proyek, pengguna dapat menggunakan *search bar* yang terletak pada bagian kanan atas tabel data proyek. Bagian *show entries* pada kiri atas tabel data proyek berguna agar pengguna dapat menentukan berapa banyaknya data yang ingin ditampilkan dalam satu halaman. Jika daftar data melebihi jumlah data yang ditampilkan per satu halaman, maka pengguna dapat melihatnya pada halaman selanjutnya dengan menekan tombol *next* pada bagian kanan bawah tabel data proyek. Jika pengguna ingin menambahkan data, maka pengguna harus menekan tombol *Add New Project* yang terletak pada bagian kiri bawah tabel data proyek. Tampilan halaman tambah data proyek ditunjukkan seperti pada Gambar 4.10 berikut ini.

BUG REPORT

Add Project Data

Project Name

Status
 Pilih ▼

Description

Start Date (estimation)

End Date (estimation)

Select involved users :

Eren (Project Manager)
 Sandy Cheeks (Project Manager Assistant)
 Yura Chloë Ramadhan (Project Manager Assistant)
 Belle (Tester)
 Russel (Developer)
 Squidward Tentacles (Developer)

Gambar 4. 10 Halaman tambah data proyek *Project Manager*

Jika pengguna telah selesai melakukan pengisian data, pengguna harus menekan tombol *Save* berwarna biru tua yang berada pada bagian bawah halaman ini untuk menyimpan data baru. Jika pengguna menekan tombol *Back* berwarna biru muda yang berada di samping tombol *Save*, maka sistem akan menuju ke halaman sebelumnya yaitu halaman data proyek.

b. *Project Manager Assistant, Tester, dan Developer*

Sesuai dengan perannya, seorang *Project Manager Assistant, Tester, dan Developer* hanya dapat melihat daftar data proyek saja tanpa dapat melakukan aksi lainnya, sehingga tampilan halaman data proyek mereka pun sama. Pada halaman data proyek ini tidak semua data proyek yang terdaftar pada sistem yang akan ditampilkan, hanya proyek-proyek yang dimana ia (pengguna) terlibat saja. Sistem akan menampilkan tabel data proyek yang terdiri dari enam kolom, yaitu no, *project name*, status, *description*, *start date*, dan *end date*. Untuk memudahkan pencarian data proyek, pengguna dapat menggunakan *search bar* yang terletak pada bagian kanan atas tabel data proyek. Bagian *show entries* pada kiri atas tabel data proyek berguna agar pengguna dapat menentukan berapa banyaknya data yang ingin ditampilkan dalam satu halaman. Jika daftar data melebihi jumlah data yang ditampilkan per satu halaman, maka pengguna dapat melihatnya pada halaman selanjutnya dengan menekan tombol *next* pada bagian kanan

bawah tabel data proyek. Sebagai salah satu contoh, Gambar 4.11 berikut ini merupakan tampilan dari halaman data proyek milik seorang *Project Manager Assistant*.

BUG REPORT

My Project

Show entries Search:

No	Project Name	Status	Description	Start Date	End Date
1	Proyek Xyz	On Progress	Cobaa aja	2021-01-16	2021-01-19
2	Proyek C	On Progress	Ini waktu nyoba tombol ubah.	2021-01-17	2021-01-23
3	Proyek B	On Progress	Yang ini juga cuma buat contoh aja	2021-01-17	2021-01-21
4	Proyek Pagi	On Hold	Coba proyek baru tapi statusnya tunda	2021-03-06	2021-03-31
5	Proyek Terbaru	On Hold	Uji coba dari awal	2021-03-09	2021-03-31
6	Proyek Jumat	On Progress	Deskripsi proyek	2021-03-12	2021-03-30
7	Proyek D	On Progress	Ini waktu nyoba ubah statusnya dari tunda jadi dikerjakan	2021-03-16	2021-03-27
8	Proyek E	On Progress	Ini nambahin lagi karna tadi nyoba tombol hapus.	2021-03-21	2021-03-26

Showing 1 to 8 of 8 entries Previous **1** Next

Gambar 4. 11 Halaman data proyek *Project Manager Assistant*





















c. Admin

Seorang *Admin* dapat melihat daftar data proyek, mengubah, serta menghapus semua data proyek yang sudah terdaftar di sistem pada halaman ini. Sistem akan menampilkan tabel data proyek yang terdiri dari tujuh kolom, yaitu no, *project name*, status, *description*, *start date*, *end date* dan *action*. Kolom *action* berisi dua tombol yang dapat digunakan untuk mengubah dan menghapus data bug. Untuk melakukan perubahan data proyek, pengguna harus menekan tombol yang berwarna orange pada kolom *action*, selanjutnya sistem akan menampilkan halaman ubah data proyek. Sedangkan untuk menghapus data proyek, pengguna dapat menekan tombol berwarna merah pada kolom *action* yang barisnya sesuai dengan proyek yang ingin dihapus. Untuk memudahkan pencarian data proyek, pengguna dapat menggunakan *search bar* yang terletak pada bagian kanan atas tabel data proyek. Bagian *show entries* pada kiri atas tabel data proyek berguna agar pengguna dapat menentukan berapa banyaknya data yang ingin ditampilkan dalam satu halaman. Jika daftar data melebihi jumlah data yang ditampilkan per satu halaman, maka pengguna dapat melihatnya pada halaman selanjutnya dengan menekan tombol *next* pada bagian kanan bawah tabel data proyek. Gambar 4.12 berikut ini merupakan tampilan dari halaman data proyek milik seorang Admin

BUG REPORT

All Projects

Show entries Search:

No	Project Name	Status	Description	Start Date	End Date	Action
1	Proyek A	On Progress	Ini cuma contoh aja	2021-01-18	2021-01-23	 
2	Proyek B	On Progress	Yang ini juga cuma buat contoh aja	2021-01-17	2021-01-21	 
3	Proyek C	On Progress	Ini waktu nyoba tombol ubah.	2021-01-17	2021-01-23	 
4	Proyek D	On Progress	Ini waktu nyoba ubah statusnya dari tunda jadi dikerjakan	2021-03-16	2021-03-27	 
5	Proyek E	On Progress	Ini nambahin lagi karna tadi nyoba tombol hapus.	2021-03-21	2021-03-26	 
6	Proyek F	On Progress	Asdfh	2021-03-06	2021-03-31	 
7	Proyek G	On Progress	Coba	0000-00-00	0000-00-00	 
8	Proyek Coba	On Hold	coba aja	2021-01-14	2021-01-30	 
9	Proyek Xyz	On Progress	Cobaa aja	2021-01-16	2021-01-19	 
10	Proyek Januari	On Progress	Asdf	2021-01-18	2021-01-29	 

Showing 1 to 10 of 14 entries Previous **1** 2 Next

Gambar 4. 12 Halaman data proyek Admin

4.1.6 Halaman My task atau Bug

a. Project Manager dan Project Manager Assistant

Halaman data *bug* akan ditampilkan jika pengguna menekan menu *My Task* pada sebelah kiri laman atau pada bagian *navbar*. Untuk tampilan halaman data *bug* milik *Project Manager* sama dengan tampilan halaman data *bug* milik *Project Manager Assistant*. Sebagai salah satu contoh, Gambar 4.13 berikut ini merupakan tampilan dari halaman data *bug* milik *Project Manager*.

BUG REPORT

My Bug List

Show entries Search:

No	Bug Title	Description	Identified Date	Evidence	Priority	Identified by	Status	Type	Project Name	Assigned To	Action
1	Error Cobaa	Ukmiwsmmed	2021-01-09	Bukti Foto Bukti video	Low	Sandy Cheeks	Fixed	Minor	Proyek B	Eren	
2	Fatal	Error	2021-01-09	Bukti Foto Bukti video	Low	Sandy Cheeks	Not Found	Minor	Proyek C	Eren	
3	Bug A	Ini cuma coba aja	2020-12-31	Bukti Foto Bukti video	Low	Eren	Fixed	Minor	Proyek D	Russel	
4	Multiple	coba banyak foto	2021-03-26	Bukti Foto Bukti video	Low	Eren	Open	Trivia	Proyek E	Russel	
5	Multiple Lagi	contohnya	2021-03-26	Bukti Foto Bukti video	Low	Eren	Open	Trivia	Proyek E	Sandy Cheeks	
6	Percobaan Multiple	Nyoba add bug di tambahbugproyek	2021-03-31	Bukti Foto Bukti video	Low	Eren	On Progress	Trivia	Proyek Xyz	Eren	
7	Bug Januari	Contoh	2021-03-16	Bukti Foto Bukti video	Medium	Eren	Open	Trivia	Proyek Januari	Russel	
8	Coba Multiple	asdf	2021-03-29	Bukti Foto Bukti video	Medium	Eren	Open	Trivia	Proyek Jumat	Yura Chloe Ramadhan	
9	Baruu	Coba a i sama	2021-03-29	Bukti Foto Bukti video	Low	Eren	Open	Trivia	Proyek Jumat	Eren	
10	Multiple Evidence	Coba multiple	2021-03-30	Bukti Foto Bukti video	Low	Eren	On Progress	Trivia	Proyek Jumat	Eren	

Showing 1 to 10 of 10 entries Previous **1** Next

[Add Bug](#)

Gambar 4. 13 Halaman data *bug* Project Manager

Berdasarkan perannya, seorang *Project Manager* dan *Project Manager Assistant* dapat melihat data *bug*, menambahkan data *bug*, mengubah data *bug*, menghapus data *bug*, mencari data *bug*, serta melihat data *progress bug*, menambahkan data *progress bug*, dan mencari data *progress bug* pada sistem. Pada halaman data *bug* *Project Manager* dan *Project Manager Assistant* ini akan ditampilkan tabel data *bug* yang terdiri dari dua belas kolom yaitu no, *bug title*, *description*, *identified date*, *evidence*, *priority*, *identified by*, *status*, *type*, *project name*, *assigned to* dan *action*. Pengguna dapat melakukan pencarian data *bug* dengan menggunakan *search bar* yang berada pada bagian kanan atas tabel data *bug*. Bagian *show entries* pada kiri atas tabel data *bug* berguna agar pengguna dapat menentukan berapa banyaknya data yang ingin ditampilkan pada satu halaman. Jika daftar data melebihi jumlah data yang ditampilkan per satu halaman, maka pengguna dapat melihatnya pada halaman selanjutnya dengan menekan tombol *next* pada bagian kanan bawah tabel data *bug*. Untuk menambahkan data *bug*, pengguna harus menekan tombol *Add Bug* pada bagian kiri bawah tabel data *bug*. Gambar 4.14 berikut ini merupakan tampilan dari halaman tambah data *bug* milik *Project Manager*.

BUG REPORT

Add Bug Data

Bug Title

Description
(ceritakan bagaimana situasi hingga bug muncul / cara memproduksi bug)

Identified Date

Type

Priority

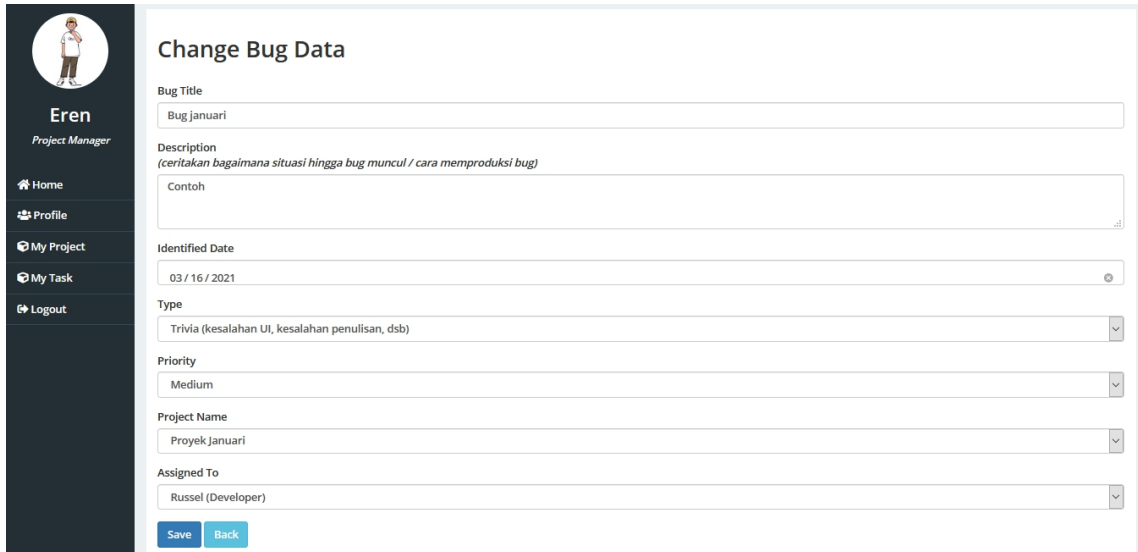
Project Name

Evidence 1
(berupa foto atau screenshot)

Evidence 2
(berupa video)

Gambar 4. 14 Halaman tambah data *bug* *Project Manager*

Pada kolom aksi, terdapat tiga tombol yang berbeda dengan fungsi yang berbeda pula. Ada tombol ubah yang berwarna hijau, tombol progress bug yang berwarna biru muda, dan tombol hapus yang berwarna merah. Untuk tombol ubah, pengguna akan dibawa ke tiga halaman yang berbeda bergantung pada status *bug*, *identified by* atau pengguna yang melaporkan bug, dan *assigned to* atau pengguna yang ditugaskan untuk menindaklanjuti *bug* tersebut. Untuk melakukan perubahan data *bug* seperti *bug title*, *description*, *identified date*, *evidence*, *priority*, *type*, *project name*, dan atau *assigned to*, pengguna harus menekan tombol berwarna hijau pada kolom *action* yang sebaris dengan data *bug* yang ingin diubah dan dengan dua persyaratan. Persyaratannya adalah pengguna merupakan orang yang melaporkan *bug* tersebut (tercantum sebagai *identified by*) dan *bug* masih berstatus *open*. Selain dari itu maka ia tidak dapat mengubah data *bug*nya lagi. Gambar 4.15 berikut ini merupakan tampilan dari halaman ubah data *bug* milik *Project Manager* jika syarat kedua syaratnya terpenuhi.



Change Bug Data

Bug Title
Bug Januari

Description
(ceritakan bagaimana situasi hingga bug muncul / cara memproduksi bug)
Contoh

Identified Date
03 / 16 / 2021

Type
Trivia (kesalahan UI, kesalahan penulisan, dsb)

Priority
Medium

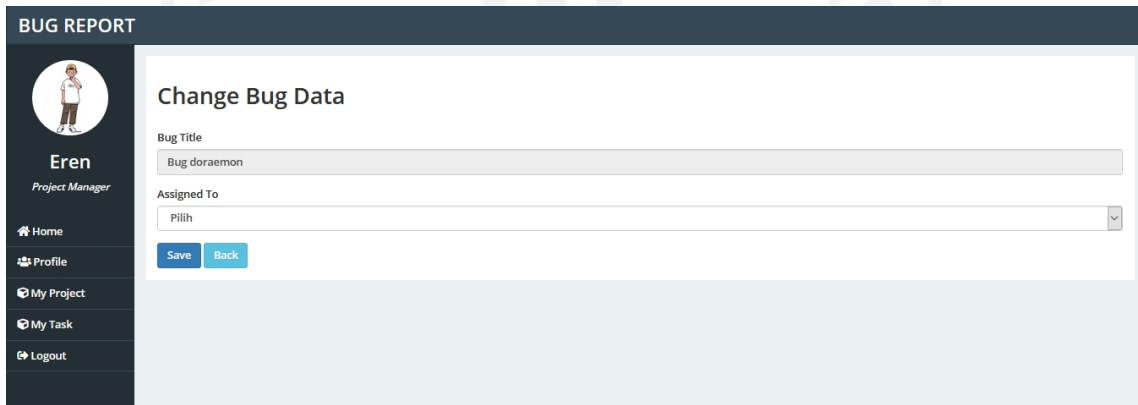
Project Name
Proyek Januari

Assigned To
Russel (Developer)

Save Back

Gambar 4. 15 Halaman ubah data *bug* *Project Manager* (a)

Jika pada kolom *assigned to* masih berisi “-”, maka pengguna dapat melakukan perubahan terhadap data *bug* tersebut namun hanya untuk mengisi *assigned to* atau orang yang akan ditugaskannya saja. Klik *Save* untuk menyimpan perubahan. Gambar 4.16 berikut ini merupakan tampilan dari halaman ubah data *bug* milik *Project Manager* dengan kondisi tersebut.



BUG REPORT

Change Bug Data

Bug Title
Bug doraemon

Assigned To
Pilih

Save Back

Gambar 4. 16 Halaman ubah data *bug* *Project Manager* (b)

Namun jika pengguna merupakan orang yang melaporkan *bug* (tercantum sebagai *identified by*) dan *bug* berstatus “*Fixed*” atau “*Not Found*”, atau pengguna merupakan orang yang ditugaskan untuk menindaklanjuti *bug* (tercantum sebagai *assigned to*) dan *bug* berstatus “*open*” atau “*reopen*” atau “*on process*”, maka saat pengguna menekan

tombol ubah maka sistem akan menampilkan halaman ubah status *bug* seperti pada Gambar 4.17 berikut ini.

Gambar 4. 17 Halaman ubah status *bug* Project Manager

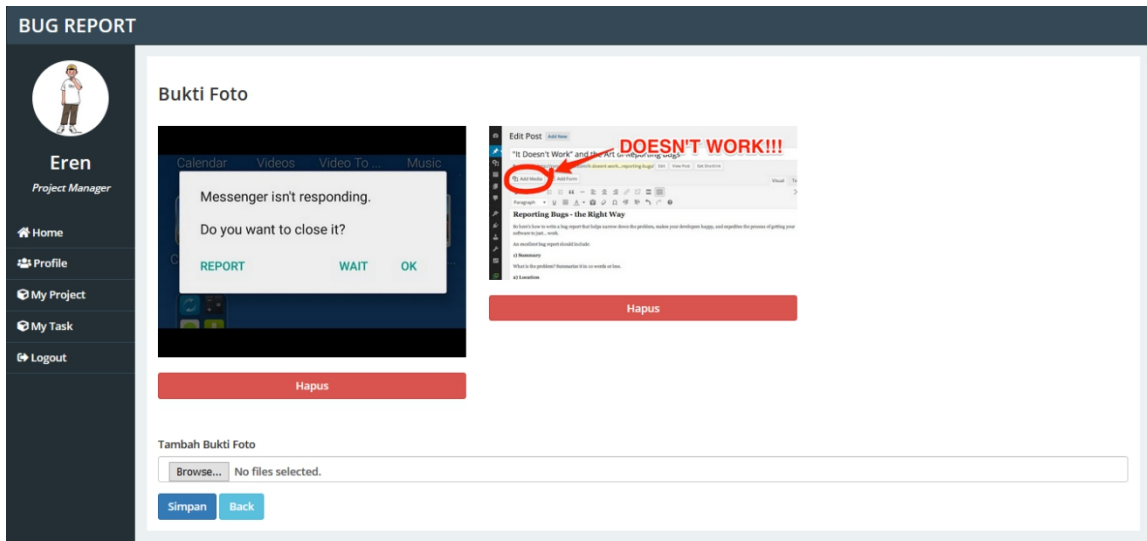
Tombol berwarna biru muda pada kolom *action* berfungsi untuk menampilkan halaman data *progress bug*. Tombol ini akan muncul jika *bug* tidak berstatus “Open” atau “Closed”. Halaman data *progress bug* divisualkan seperti pada Gambar 4.18 berikut ini.

No	Date	Description	Status
1	2021-03-05	tes 1	On Progress
2	2021-03-05	cobaaa	On Progress
3	2021-03-05	done	Fixed
4	2021-03-05		Reopen
5	2021-03-05		Not Found

Gambar 4. 18 Halaman data progress *bug* Project Manager

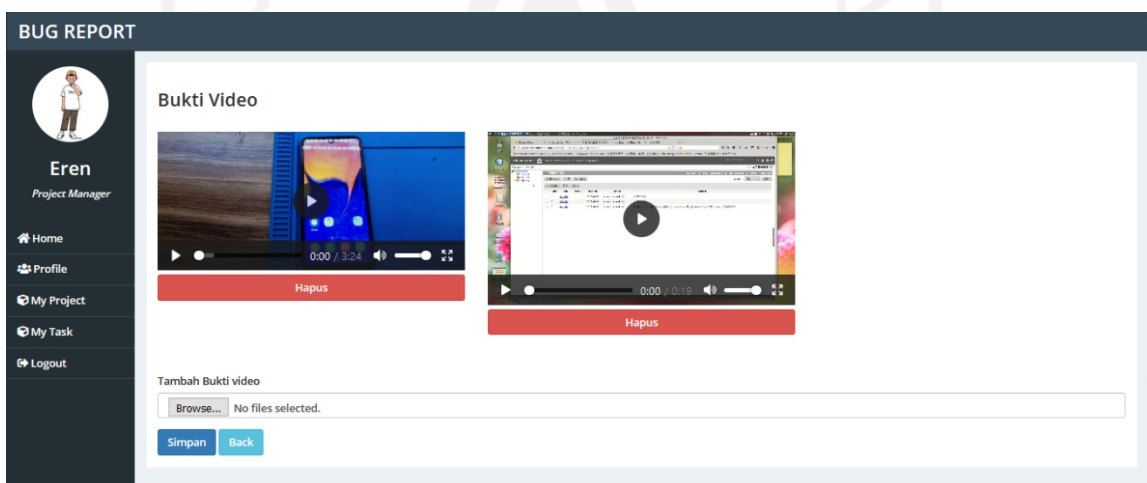
Untuk menghapus *bug*, pengguna harus menekan tombol berwarna merah pada kolom *action* dengan syarat ia merupakan pelapor *bug* (tercantum sebagai *identified by*) dan *bug* tidak berstatus “Closed”. Pada kolom *evidence* terdapat dua *link* yaitu Bukti Foto dan Bukti *Video*. Jika pengguna menekan *link* Bukti Foto maka sistem akan menampilkan halaman bukti foto *bug*. Jika pengguna merupakan pelapor *bug* (tercantum sebagai *identified by*) dan status *bug* masih “Open”, maka pengguna dapat melakukan

aksi penghapusan foto bukti dan juga dapat menambah foto bukti *bug*. Tampilan dari halaman bukti foto ditampilkan seperti pada Gambar 4.19 berikut ini.



Gambar 4. 19 Halaman bukti foto *bug*









Jika pengguna menekan *link* Bukti *Video*, maka sistem akan menampilkan halaman bukti *video bug*. Jika pengguna merupakan pelapor bug (tercantum sebagai *identified by*) dan status *bug* masih “*Open*”, maka pengguna dapat melakukan aksi penghapusan *video* bukti dan juga dapat menambah *video* bukti *bug*. Tampilan dari halaman bukti *video* ditampilkan seperti pada Gambar 4.20 berikut ini.



Gambar 4. 20 Halaman bukti *video bug*

b. *Tester*

Sama seperti *Project Manager* dan *Project Manager Assistant*, seorang *Tester* dapat melihat data *bug*, menambahkan data *bug*, mengubah data *bug*, menghapus data *bug*, mencari data *bug*, serta melihat data *progress bug*, menambahkan data *progress bug*, dan mencari data *progress bug* pada sistem. Pada halaman data *bug Tester* ini akan ditampilkan tabel data *bug* yang terdiri dari dua belas kolom yaitu *no*, *bug title*, *description*, *identified date*, *evidence*, *priority*, *identified by*, *status*, *type*, *project name*, *assigned to* dan *action*. Gambar 4.21 berikut ini merupakan tampilan dari halaman data *bug* milik *Tester*.

No	Bug Title	Description	Identified Date	Evidence	Priority	Identified by	Status	Type	Project Name	Action
1	Fatal Error2	aa	2021-01-07	Bukti Foto Bukti video	Low	Belle	On Progress	Minor	Proyek A	 
2	Bug B	Asdfg	2021-01-04	Bukti Foto Bukti video	Medium	Belle	Close	Trivia	Proyek B	
3	April 3	Cobainn add bug tester	2021-04-01	Bukti Foto Bukti video	Low	Belle	Open	Trivia	Proyek G	 
4	Jumat	Deskripsi	2021-03-12	Bukti Foto Bukti video	Low	Belle	Close	Trivia	Proyek Jumat	
5	Bug doraemon	contoh aja1	2021-03-15	Bukti Foto Bukti video	Medium	Belle	Open	Minor	Proyek Jumat	 

Gambar 4. 21 Halaman data *bug Tester*

Pengguna dapat melakukan pencarian data *bug* dengan menggunakan *search bar* yang berada pada bagian kanan atas tabel data *bug*. Bagian *show entries* pada kiri atas tabel data *bug* berguna agar pengguna dapat menentukan berapa banyaknya data yang ingin ditampilkan pada satu halaman. Jika daftar data melebihi jumlah data yang ditampilkan per satu halaman, maka pengguna dapat melihatnya pada halaman selanjutnya dengan menekan tombol *next* pada bagian kanan bawah tabel data *bug*. Untuk menambahkan data *bug*, pengguna harus menekan tombol *Add Bug* pada bagian kiri bawah tabel data *bug*. Untuk mengubah data *bug* pengguna dapat menekan tombol yang berwarna hijau pada kolom *action*. Untuk tampilan dari halaman tambah data *bug* dan ubah data *bug* milik *Tester* ini sama saja seperti milik *Project Manager* dan *Project Manager Assistant*, hanya ada sedikit perbedaan yaitu *Tester* tidak memilih atau mengisi

nama orang yang akan ditugaskan untuk memperbaiki *bug* (*assigned to*). Untuk melihat data *progress bug* pengguna dapat menekan tombol yang berwarna biru muda pada kolom *action* dengan syarat *bug* tidak berstatus “*Open*” atau “*Closed*”. Dan untuk menghapus *bug*, pengguna harus menekan tombol berwarna merah pada kolom *action* dengan syarat ia merupakan pelapor *bug* (tercantum sebagai *identified by*) dan *bug* tidak berstatus “*Closed*”.

c. Developer

Halaman data *bug* akan ditampilkan jika pengguna menekan menu *My List* pada sebelah kiri laman. Halaman data *bug* ditampilkan seperti pada Gambar 4.22 berikut ini.

No	Bug Title	Description	Identified Date	Evidence	Priority	Identified by	Status	Type	Project Name	Assigned To	Action
1	Bug Doraemon	contoh aja1	2021-03-15	Bukti Foto Bukti video	Medium	Belle	Open	Minor	Proyek Jumat	-	
2	April 3	Cobainn add bug tester	2021-04-01	Bukti Foto Bukti video	Low	Belle	Open	Trivia	Proyek G	-	
3	Crash Ubah, Belle Coba Ubahh	ubah	2020-11-17	Bukti Foto Bukti video	Low	Sandy Cheeks	Close	Major	Proyek B	Russel	
4	Data A	Qwert	2020-12-22	Bukti Foto Bukti video	Low	Sandy Cheeks	Fixed	Major	Proyek C	Russel	
5	Bug A	Ini cuma coba aja	2020-12-31	Bukti Foto Bukti video	Low	Eren	Fixed	Minor	Proyek D	Russel	
6	Bug B	Asdfg	2021-01-04	Bukti Foto Bukti video	Medium	Belle	Close	Trivia	Proyek B	Russel	
7	Eroorr	Alsdkc	2021-01-09	Bukti Foto Bukti video	Low	Sandy Cheeks	On Progress	Minor	Proyek C	Russel	
8	Coba 1	Nyoba nambah error di tambahbugproyek	2021-01-15	Bukti Foto Bukti video	Low	Sandy Cheeks	On Progress	Minor	Proyek B	Russel	
9	Jumat	Deskripsi	2021-03-12	Bukti Foto Bukti video	Low	Belle	Close	Trivia	Proyek Jumat	Russel	
10	Bug Januari	Contoh	2021-03-16	Bukti Foto Bukti video	Medium	Eren	Open	Trivia	Proyek Januari	Russel	

Gambar 4. 22 Halaman data *bug Developer*

Berdasarkan perannya, seorang *Developer* dapat melihat data *bug*, mencari data *bug*, menambah data *progress bug*, melihat data *progress bug*, serta mencari data *progress* pada sistem. Pada halaman data *bug Developer* ini akan ditampilkan tabel data *bug* yang terdiri dari dua belas kolom yaitu no, *bug title*, *description*, *identified date*, *evidence*, *priority*, *identified by*, *status*, *type*, *project name*, *assigned to* dan *action*. Untuk melakukan pengubahan status (menambah data *progress bug*) dan melihat data *progress*

bug, pengguna dapat menggunakan tombol yang berwarna hijau dan biru muda pada kolom *action*.

d. Admin

Halaman data *bug* akan ditampilkan jika pengguna menekan menu *bug* pada sebelah kiri laman atau pada bagian *navbar*. Halaman data *bug* ditampilkan seperti pada Gambar 4.23 berikut ini.

BUG REPORT

All Bug List

Show entries

Search:

No	Bug Title	Description	Identified Date	Evidence	Priority	Identified by	Status	Type	Project Name	Assigned To
1	Tombol Simpan Tidak Berfungsi	Data Tidak Bisa Tersimpan Karna Tombol Tidak Berfungsi	2020-11-23	Bukti Foto Bukti video	Low	Yura Chloe Ramadhan	Close	Minor	Proyek A	Squidward Tentacles
2	Fatal Error2	Aa	2021-01-07	Bukti Foto Bukti video	Low	Belle	On Progress	Minor	Proyek A	Squidward Tentacles
3	Crash Ubah, Belle Coba Ubahh	Ubah	2020-11-17	Bukti Foto Bukti video	Low	Sandy Cheeks	Close	Major	Proyek B	Russel
4	Bug B	Asdfg	2021-01-04	Bukti Foto Bukti video	Medium	Belle	Close	Trivia	Proyek B	Russel
5	Error Cobe	Addas	2021-01-09	Bukti Foto Bukti video	Low	Sandy Cheeks	On Progress	Minor	Proyek B	Squidward Tentacles
6	Error Cobaa	Ukmiwsmmed	2021-01-09	Bukti Foto Bukti video	Low	Sandy Cheeks	Fixed	Minor	Proyek B	Eren
7	Coba 1	Nyoba Nambah Error Di Tambahbugproyek	2021-01-15	Bukti Foto Bukti video	Low	Sandy Cheeks	On Progress	Minor	Proyek B	Russel
8	Coba2	Cobaa	2021-01-16	Bukti Foto Bukti video	Low	Sandy Cheeks	On Progress	Minor	Proyek B	Sandy Cheeks
9	Coba B	Test	2021-03-22	Bukti Foto Bukti video	High	Sandy Cheeks	Open	Major	Proyek B	Squidward Tentacles
10	Data A	Qwert	2020-12-22	Bukti Foto Bukti video	Low	Sandy Cheeks	Fixed	Major	Proyek C	Russel

Showing 1 to 10 of 28 entries

Previous **1** 2 3 Next

Gambar 4. 23 Halaman data *bug* Admin

Berdasarkan perannya, seorang Admin hanya dapat melihat dan mencari data *bug* pada sistem. Pada halaman data bug admin ini akan ditampilkan tabel data bug yang terdiri dari sebelas kolom yaitu no, *bug title*, *description*, *identified date*, *evidence*, *priority*, *identified by*, *status*, *type*, *project name*, dan *assigned to*. Pengguna juga dapat melakukan pencarian data *bug* dengan menggunakan *search bar* yang berada pada bagian kanan atas tabel. Bagian *show entries* pada kiri atas tabel berguna agar pengguna dapat menentukan berapa banyaknya data yang ingin ditampilkan pada satu halaman. Jika daftar data melebihi jumlah data yang ditampilkan per satu halaman, maka pengguna dapat melihatnya pada halaman selanjutnya dengan menekan tombol *next* pada bagian kanan bawah tabel.

4.2 Pengujian *Black Box*

Pengujian dilakukan oleh penulis setelah sistem selesai dibuat dengan menggunakan metode *black box testing*. *Black box testing* merupakan salah satu tipe dari *User Acceptance Test* (UAT). Pengujian ini dilakukan dengan tujuan untuk mengetahui atau mengidentifikasi kesalahan yang mungkin terjadi pada sistem. Pengujian dengan metode ini dilakukan dengan cara menguji fungsionalitas sistem tanpa melihat *source code* atau struktur kode internalnya. Detail pengujian ditunjukkan seperti pada tabel 4.1 dengan keterangan sebagai berikut.

A : Admin

PM : *Project Manager*

PMA : *Project Manager Assistant*

D : *Developer*

T : *Tester*

Tabel 4. 1 *User Acceptance Test*

No.	Test Case	Steps	Expected Result	Actual Result
A-01	Menguji tombol <i>login</i> dengan Akun <i>Google</i> pada halaman <i>login user</i> dan <i>login</i> dengan akun admin.	<ol style="list-style-type: none"> 1. Pilih tombol <i>login</i> dengan akun <i>Google</i> pada halaman <i>login user</i>. 2. Pilih akun <i>google</i> yang sudah ada. Jika belum ada maka pilih gunakan akun yang lain. 3. Masukkan <i>email</i> atau nomor ponsel yang terdaftar sebagai admin. 4. Pilih berikutnya. 	Sistem menampilkan halaman <i>dashboard</i> admin menandakan <i>login</i> berhasil.	Admin berhasil <i>login</i> dan halaman <i>dashboard</i> admin berhasil tampil.

		<ol style="list-style-type: none"> 5. Masukkan kata sandi. 6. Pilih berikutnya. 		
PM-01	Menguji tombol <i>login</i> dengan Akun <i>Google</i> pada halaman <i>login user</i> dan login dengan akun <i>project manager</i> .	<ol style="list-style-type: none"> 1. Pilih tombol <i>login</i> dengan akun <i>Google</i> pada halaman <i>login user</i>. 2. Pilih akun <i>google</i> yang sudah ada. Jika belum ada maka pilih gunakan akun yang lain. 3. Masukkan <i>email</i> atau nomor ponsel yang terdaftar sebagai <i>project manager</i>. 4. Pilih berikutnya. 5. Masukkan kata sandi. 6. Pilih berikutnya. 	Sistem menampilkan halaman <i>dashboard project manager</i> menandakan <i>login</i> berhasil.	<i>Project manager</i> berhasil <i>login</i> dan halaman <i>dashboard project manager</i> berhasil tampil.
PMA-01	Menguji tombol <i>login</i> dengan Akun <i>Google</i> pada halaman <i>login user</i> dan login dengan akun <i>project manager</i>	<ol style="list-style-type: none"> 1. Pilih tombol <i>login</i> dengan akun <i>Google</i> pada halaman <i>login user</i>. 2. Pilih akun <i>google</i> yang sudah ada. Jika belum ada maka pilih gunakan akun 	Sistem menampilkan halaman <i>dashboard project manager assistant</i> menandakan <i>login</i> berhasil.	<i>Project manager assistant</i> berhasil <i>login</i> dan halaman <i>dashboard project manager</i>

	<i>assistant.</i>	<p>yang lain.</p> <ol style="list-style-type: none"> 3. Masukkan <i>email</i> atau nomor ponsel yang terdaftar sebagai <i>project manager assistant.</i> 4. Pilih berikutnya. 5. Masukkan kata sandi. 6. Pilih berikutnya. 		<i>assistant</i> berhasil tampil.
D-01	Menguji tombol <i>login</i> dengan Akun <i>Google</i> pada halaman <i>login user</i> dan <i>login</i> dengan akun <i>developer.</i>	<ol style="list-style-type: none"> 1. Pilih tombol <i>login</i> dengan akun <i>Google</i> pada halaman <i>login user.</i> 2. Pilih akun <i>google</i> yang sudah ada. Jika belum ada maka pilih gunakan akun yang lain. 3. Masukkan <i>email</i> atau nomor ponsel yang terdaftar sebagai <i>developer.</i> 4. Pilih berikutnya. 5. Masukkan kata sandi. 6. Pilih berikutnya. 	Sistem menampilkan halaman <i>dashboard developer</i> menandakan <i>login</i> berhasil.	<i>Tester</i> berhasil <i>login</i> dan halaman <i>dashboard developer</i> berhasil tampil.
T-01	Menguji tombol	<ol style="list-style-type: none"> 1. Pilih tombol <i>login</i> 	Sistem	<i>Tester</i> berhasil

	<p><i>login</i> dengan Akun <i>Google</i> pada halaman <i>login user</i> dan login dengan akun <i>tester</i>.</p>	<p>dengan akun <i>Google</i> pada halaman <i>login user</i>.</p> <ol style="list-style-type: none"> 2. Pilih akun <i>google</i> yang sudah ada. Jika belum ada maka pilih gunakan akun yang lain. 3. Masukkan <i>email</i> atau nomor ponsel yang terdaftar sebagai <i>tester</i>. 4. Pilih berikutnya. 5. Masukkan kata sandi. 6. Pilih berikutnya. 	<p>menampilkan halaman <i>dashboard tester</i> menandakan <i>login</i> berhasil.</p>	<p><i>login</i> dan halaman <i>dashboard tester</i> berhasil tampil.</p>
--	---	---	--	--

Tabel 4.2 berikut merupakan keseluruhan test case untuk kegiatan pengujian pada *user Admin*.

Tabel 4. 2 *Test case summary* Admin

No.	Test Case
A-02	Menguji fungsi <i>link Bug List</i> pada halaman <i>dashboard</i> .
A-03	Menguji fungsi tombol <i>Profile</i> pada bagian <i>navbar</i> .
A-04	Menguji fungsi tombol <i>Change Profile</i> pada halaman profil.
A-05	Menguji fungsi tombol <i>Save</i> pada halaman ubah profil.
A-06	Menguji fungsi tombol <i>Back</i> pada halaman ubah profil.
A-07	Menguji fungsi tombol <i>User</i> pada bagian <i>navbar</i> .

A-08	Menguji fungsi tombol <i>Add New User</i> pada halaman data <i>user</i> .
A-09	Menguji fungsi tombol <i>Save</i> pada halaman tambah data <i>user</i> .
A-10	Menguji fungsi tombol <i>Back</i> pada halaman tambah data <i>user</i> .
A-11	Menguji fungsi tombol ubah <i>user</i> pada halaman data <i>user</i> .
A-12	Menguji fungsi tombol <i>Save</i> pada halaman ubah data <i>user</i> .
A-13	Menguji fungsi tombol <i>Back</i> pada halaman ubah data <i>user</i> .
A-14	Menguji fungsi tombol hapus pada halaman data <i>user</i> .
A-15	Menguji fungsi tombol <i>Project</i> pada bagian <i>navbar</i> .
A-16	Menguji fungsi tombol ubah pada halaman data proyek.
A-17	Menguji fungsi tombol <i>Save</i> pada halaman ubah data proyek.
A-18	Menguji fungsi tombol <i>Back</i> pada halaman ubah data proyek.
A-19	Menguji fungsi tombol hapus pada halaman data proyek.
A-20	Menguji fungsi tombol <i>Bug</i> pada bagian <i>navbar</i> .
A-21	Menguji fungsi <i>link</i> bukti foto pada halaman data <i>bug</i> .
A-22	Menguji fungsi tombol <i>Back</i> pada halaman bukti foto.
A-23	Menguji fungsi <i>link</i> bukti <i>video</i> pada halaman data <i>bug</i> .
A-24	Menguji fungsi tombol <i>Back</i> pada halaman bukti <i>video</i> .
A-25	Menguji fungsi tombol <i>Logout</i> pada <i>navbar</i> .

Tabel 4.3 berikut merupakan keseluruhan test case untuk kegiatan pengujian pada *user Project Manager*.

Tabel 4. 3 *Test case summary Project Manager*

No.	Test Case
PM-02	Menguji fungsi <i>link Bug List</i> pada halaman <i>dashboard</i> .

PM-03	Menguji fungsi tombol <i>Profile</i> pada bagian <i>navbar</i> .
PM-04	Menguji fungsi tombol <i>Change Profile</i> pada halaman profil.
PM-05	Menguji fungsi tombol <i>Save</i> pada halaman ubah profil.
PM-06	Menguji fungsi tombol <i>Back</i> pada halaman ubah profil.
PM-07	Menguji fungsi tombol <i>My Project</i> pada bagian <i>navbar</i> .
PM-08	Menguji fungsi tombol ubah pada halaman data proyek.
PM-09	Menguji fungsi tombol <i>Save</i> pada halaman ubah data proyek.
PM-10	Menguji fungsi tombol <i>Back</i> pada halaman ubah data proyek.
PM-11	Menguji fungsi tombol hapus pada halaman data proyek.
PM-12	Menguji fungsi tombol <i>Add New Project</i> pada halaman data proyek.
PM-13	Menguji fungsi tombol <i>Save</i> pada halaman tambah data proyek.
PM-14	Menguji fungsi tombol <i>Back</i> pada halaman tambah data proyek.
PM-15	Menguji fungsi tombol <i>My Task</i> pada bagian <i>navbar</i> .
PM-16	Menguji fungsi tombol <i>Add Bug</i> pada halaman <i>My Task</i> .
PM-17	Menguji fungsi tombol <i>Save</i> pada halaman tambah data <i>bug</i> .
PM-18	Menguji fungsi tombol <i>Back</i> pada halaman tambah data <i>bug</i> .
PM-19	Menguji fungsi tombol ubah pada halaman data <i>bug</i> .
PM-20	Menguji fungsi tombol hapus pada halaman data <i>bug</i> .
PM-21	Menguji fungsi tombol <i>progress</i> pada halaman data <i>bug</i> .
PM-22	Menguji fungsi <i>link</i> bukti foto pada halaman data <i>bug</i> .
PM-23	Menguji fungsi tombol <i>delete</i> pada halaman bukti foto.
PM-24	Menguji fungsi tombol <i>Save</i> pada halaman bukti foto.
PM-25	Menguji fungsi tombol <i>Back</i> pada halaman bukti foto.

PM-26	Menguji fungsi <i>link</i> bukti <i>video</i> pada halaman data <i>bug</i> .
PM-27	Menguji fungsi tombol <i>delete</i> pada halaman bukti <i>video</i> .
PM-28	Menguji fungsi tombol <i>Save</i> pada halaman bukti <i>video</i> .
PM-29	Menguji fungsi tombol <i>Back</i> pada halaman bukti <i>video</i> .
PM-30	Menguji fungsi tombol <i>Logout</i> pada <i>navbar</i> .

Tabel 4.4 berikut merupakan keseluruhan test case untuk kegiatan pengujian pada *user Project Manager Assistant*.

Tabel 4. 4 *Test case summary Project Manager Assistant*

No.	Test Case
PMA-02	Menguji fungsi <i>link Bug List</i> pada halaman <i>dashboard</i> .
PMA-03	Menguji fungsi tombol <i>Profile</i> pada bagian <i>navbar</i> .
PMA-04	Menguji fungsi tombol <i>Change Profile</i> pada halaman profil.
PMA-05	Menguji fungsi tombol <i>Save</i> pada halaman ubah profil.
PMA-06	Menguji fungsi tombol <i>Back</i> pada halaman ubah profil.
PMA-07	Menguji fungsi tombol <i>My Project</i> pada bagian <i>navbar</i> .
PMA-08	Menguji fungsi tombol <i>My Task</i> pada bagian <i>navbar</i> .
PMA-09	Menguji fungsi tombol <i>Add Bug</i> pada halaman <i>My Task</i> .
PMA-10	Menguji fungsi tombol <i>Save</i> pada halaman tambah data <i>bug</i> .
PMA-11	Menguji fungsi tombol <i>Back</i> pada halaman tambah data <i>bug</i> .
PMA-12	Menguji fungsi tombol ubah pada halaman data <i>bug</i> .
PMA-13	Menguji fungsi tombol hapus pada halaman data <i>bug</i> .
PMA-14	Menguji fungsi tombol <i>progress</i> pada halaman data <i>bug</i> .
PMA-15	Menguji fungsi <i>link</i> bukti foto pada halaman data <i>bug</i> .

PMA-16	Menguji fungsi tombol <i>delete</i> pada halaman bukti foto.
PMA-17	Menguji fungsi tombol <i>Save</i> pada halaman bukti foto.
PMA-18	Menguji fungsi tombol <i>Back</i> pada halaman bukti foto.
PMA-19	Menguji fungsi <i>link</i> bukti <i>video</i> pada halaman data <i>bug</i> .
PMA-20	Menguji fungsi tombol <i>delete</i> pada halaman bukti <i>video</i> .
PMA-21	Menguji fungsi tombol <i>Save</i> pada halaman bukti <i>video</i> .
PMA-22	Menguji fungsi tombol <i>Back</i> pada halaman bukti <i>video</i> .
PMA-23	Menguji fungsi tombol <i>Logout</i> pada <i>navbar</i> .

Tabel 4.5 berikut merupakan keseluruhan test case untuk kegiatan pengujian pada *user Developer*.

Tabel 4. 2 *Test case summary Developer*

No.	Test Case
D-02	Menguji fungsi <i>link Bug List</i> pada halaman <i>dashboard</i> .
D-03	Menguji fungsi tombol <i>Profile</i> pada bagian <i>navbar</i> .
D-04	Menguji fungsi tombol <i>Change Profile</i> pada halaman profil.
D-05	Menguji fungsi tombol <i>Save</i> pada halaman ubah profil.
D-06	Menguji fungsi tombol <i>Back</i> pada halaman ubah profil.
D-07	Menguji fungsi tombol <i>My Project</i> pada bagian <i>navbar</i> .
D-08	Menguji fungsi tombol <i>My Task</i> pada bagian <i>navbar</i> .
D-09	Menguji fungsi tombol ubah pada halaman data <i>bug</i> .
D-10	Menguji fungsi tombol <i>progress</i> pada halaman data <i>bug</i> .
D-11	Menguji fungsi <i>link</i> bukti foto pada halaman data <i>bug</i> .
D-12	Menguji fungsi tombol <i>Back</i> pada halaman bukti foto.

D-13	Menguji fungsi <i>link</i> bukti <i>video</i> pada halaman data <i>bug</i> .
D-14	Menguji fungsi tombol <i>Back</i> pada halaman bukti <i>video</i> .
D-15	Menguji fungsi tombol <i>Logout</i> pada <i>navbar</i> .

Tabel 4.6 berikut merupakan keseluruhan test case untuk kegiatan pengujian pada *user Tester*.

Tabel 4. 3 *Test case summary Tester*

No.	Test Case
T-02	Menguji fungsi <i>link Bug List</i> pada halaman <i>dashboard</i> .
T-03	Menguji fungsi tombol <i>Profile</i> pada bagian <i>navbar</i> .
T-04	Menguji fungsi tombol <i>Change Profile</i> pada halaman profil.
T-05	Menguji fungsi tombol <i>Save</i> pada halaman ubah profil.
T-06	Menguji fungsi tombol <i>Back</i> pada halaman ubah profil.
T-07	Menguji fungsi tombol <i>My Project</i> pada bagian <i>navbar</i> .
T-08	Menguji fungsi tombol <i>My Task</i> pada bagian <i>navbar</i> .
T-09	Menguji fungsi tombol <i>Add Bug</i> pada halaman data <i>bug</i> .
T-10	Menguji fungsi tombol <i>Save</i> pada halaman tambah data <i>bug</i> .
T-11	Menguji fungsi tombol <i>Back</i> pada halaman tambah data <i>bug</i> .
T-12	Menguji fungsi tombol ubah pada halaman data <i>bug</i> .
T-13	Menguji fungsi tombol hapus pada halaman data <i>bug</i> .
T-14	Menguji fungsi tombol <i>progress</i> pada halaman data <i>bug</i> .
T-15	Menguji fungsi <i>link</i> bukti foto pada halaman data <i>bug</i> .
T-16	Menguji fungsi tombol <i>delete</i> pada halaman bukti foto.
T-17	Menguji fungsi tombol <i>Save</i> pada halaman bukti foto.

T-18	Menguji fungsi tombol <i>Back</i> pada halaman bukti foto.
T-19	Menguji fungsi <i>link</i> bukti <i>video</i> pada halaman data <i>bug</i> .
T-20	Menguji fungsi tombol <i>delete</i> pada halaman bukti <i>video</i> .
T-21	Menguji fungsi tombol <i>Save</i> pada halaman bukti <i>video</i> .
T-22	Menguji fungsi tombol <i>Back</i> pada halaman bukti <i>video</i> .
T-23	Menguji fungsi tombol <i>Logout</i> pada <i>navbar</i> .

4.3 User Experience Testing (UX Testing)

User experience testing merupakan salah satu metode pengujian yang dilakukan untuk menilai tingkat kepuasan serta kenyamanan seseorang atau pengguna terhadap suatu produk, sistem, dan atau jasa. Hasil dari pengujian atau evaluasi ini dapat berupa rasa dan pengalaman yang dirasakan atau dialami oleh pengguna seperti memuaskan, nyaman, menghibur, membantu, indah, kreatif, stimulasi kognitif, menyenangkan, lucu, mengejutkan, menantang, membosankan, membuat frustrasi, atau bahkan mengganggu. Berikut merupakan tahap-tahap *UX testing* yang dilakukan pada *web* Sistem Informasi Pencatatan *Bug* ini.

1. Membuat skenario dan *form* penilaian.

Skenario dibuat berdasarkan masing-masing jabatan pada sistem yaitu Admin (Gambar 4.24), *Project Manager* (Gambar 4.25), *Project Manager Assistant*, *Tester* (Gambar 4.26), dan *Developer* (Gambar 4.27). Skenario kasus untuk Admin ialah melakukan penambahan data *user* baru. Skenario kasus untuk *Project Manager* ialah menambahkan data proyek baru dan menambahkan data bug baru. Untuk skenario kasus *Project Manager Assistant* dianggap sama seperti skenario kasus *Project Manager* karena kemiripan peran mereka. Selanjutnya untuk *Tester* diberi skenario kasus menambahkan *bug* baru. Untuk *Developer* skenario kasusnya ialah *updating* atau perubahan status *bug*. Terakhir, ada *form* penilaian yang berisikan pertanyaan-pertanyaan mengenai pengalaman pengguna saat mengakses *web* Sistem Informasi Pencatatan *Bug* CDS *Bug Base*.

Skenario Tes Admin

Bukalah situs web cdsbugbase.site lalu login dengan akun google Anda untuk menyelesaikan skenario tes berikut ini.

Skenario : Kamu adalah seorang Admin dan hendak menambahkan tiga user baru pada sistem. Ketiga user baru tersebut ialah seorang Project Manager, seorang Tester, dan seorang Developer. Note : isi nama user sesuai keinginan Anda dengan jabatan(position) sesuai ketentuan pada skenario. Isi email user dengan alamat email yang terdaftar sebagai Google Account.

Case : Tambah user baru

Setelah berhasil mengerjakan skenario di atas, berikan link form skenario berikut ini kepada Project Manager yang telah anda daftarkan pada sistem.

Form skenario Project Manager : <https://forms.gle/APnwY8cJQY8pFugWA>

Selanjutnya, berikan link form skenario berikut ini kepada Tester yang telah anda daftarkan pada sistem.

Form skenario Tester : <https://forms.gle/zVNdFns2qyft6GyD8>

Kemudian, setelah memastikan Project Manager dan Tester sudah selesai mengerjakan skenario tesnya, berikan link form skenario berikut ini kepada Developer yang telah anda daftarkan pada sistem.

Form skenario Developer : <https://forms.gle/CP2iq5GXeSFHBsx3A>

Gambar 4. 24 Skenario UX *testing* Admin

Skenario Tes Project Manager

Bukalah situs web cdsbugbase.site lalu login dengan akun google Anda untuk menyelesaikan skenario tes berikut ini.

Skenario 1 : Anda adalah seorang Project Manager. Saat ada proyek baru Anda harus mencatatkannya pada web CDSBugBase. Proyek tersebut bernama Test Case Project dengan status On Progress. Isi deskripsinya dengan kalimat ini: Test Case Project merupakan proyek yang dijalankan untuk pengujian cobaan. Isi estimasi mulainya proyek ini dengan tanggal saat Anda menjalankan skenario ini, dan estimasi selesainya adalah satu bulan setelahnya. Anda perlu menambahkan nama Anda, seorang Project Manager Assistant, seorang Tester, dan seorang Developer yang akan dilibatkan dalam proyek ini.
Case : Tambah Proyek Baru

Skenario 2 : Hari ini Anda menemukan sebuah bug pada Test Case Project yang telah Anda tambahkan sebelumnya dan Anda hendak melaporkan atau mencatatkannya pada sistem dengan Foto (boleh 1 atau lebih) dan Video (boleh 1 atau lebih) sebagai bukti (evidence). Bug tersebut akan Anda tugaskan kepada seorang Developer yang sudah Anda libatkan dalam proyek tersebut. Note : Anda dapat menambahkan bug melalui pilihan menu My Task atau melalui panel proyek pada halaman utama. Nama, deskripsi, tipe, dan prioritas bug diisi dengan data sesuai keinginan Anda.
Case : Tambah Bug Baru

Gambar 4. 25 Skenario UX testing Project Manager

Skenario Tes Tester

Bukalah situs web cdsbugbase.site lalu login dengan akun google Anda untuk menyelesaikan skenario tes berikut ini.

Skenario 1 : Anda adalah seorang Tester dan hari ini Anda menemukan sebuah bug pada Test Case Project dimana Anda terlibat dalam proyek tersebut. Anda hendak melaporkan atau mencatatkannya pada sistem dengan Foto (boleh 1 atau lebih) dan Video (boleh 1 atau lebih) sebagai bukti (evidence). Note : Anda dapat menambahkan bug melalui pilihan menu My Task atau melalui panel proyek pada halaman utama. Judul, deskripsi, tipe, dan prioritas bug diisi dengan data sesuai keinginan Anda.
Case : Tambah Bug Baru

Gambar 4. 26 Skenario UX testing Tester

Skenario Tes Developer

Bukalah situs web cdsbugbase.site lalu login dengan akun google Anda untuk menyelesaikan skenario tes berikut ini.

Skenario : Kamu adalah seorang Developer yang telah menerima tugas untuk memperbaiki bug pada proyek "Test Case Project". Kamu hendak mengerjakan semua tugas tersebut dan melakukan update status bug pada sistem. Untuk menyatakan bahwa kamu hendak memulai mengerjakan tugas tersebut, kamu harus melakukan update status bug dari "Open" menjadi "On Progress" dengan menyertakan deskripsi/keterangan, contohnya "perbaikan bug dimulai". **Note :** bisa melakukan update status bug melalui menu My Task pada navbar atau bisa juga melalui panel proyek pada halaman home(dashboard).
Case : Update status bug

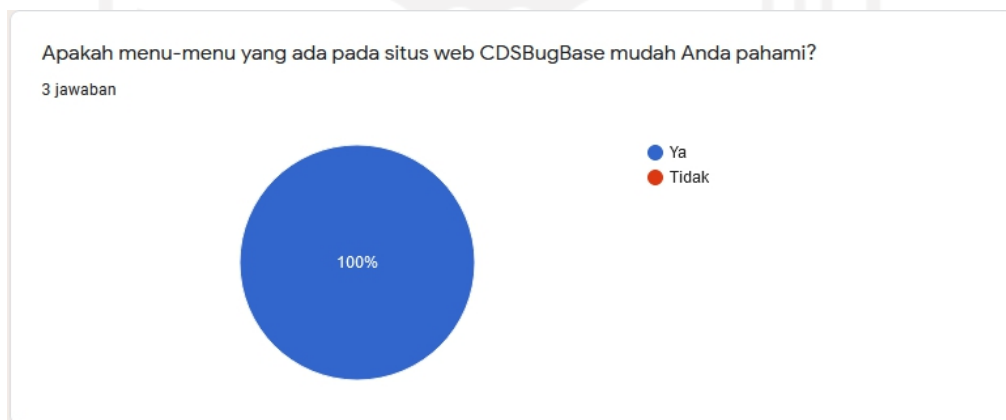
Gambar 4. 27 Skenario UX *testing Developer*

2. Menentukan *tester* atau *user* penguji yang akan mengerjakan skenario dan memberikan penilaian. Pengujian sistem dilakukan oleh tim Center of Data Science Universitas Islam Indonesia (CDS UII) yang terdiri dari tiga orang penguji dan salah satu diantaranya merangkap dua jabatan sekaligus. Ketiganya ialah Septia Rani, S.T., M.Cs. sebagai Admin dan *Project Manager*, Muhammad Pandu Widodo sebagai *Developer*, dan Ridho Imantiyar sebagai *Tester*. Tim CDS UII inilah yang nantinya akan menggunakan sistem informasi pencatatan bug berbasis web ini.
3. Penguji mengerjakan skenario yang telah diberikan sesuai dengan jabatannya masing-masing. Pengujian *user experience* ini dilakukan secara daring menggunakan aplikasi *Zoom Meeting* dan dilakukan bersamaan pada hari Jumat, 7 Mei 2021.
4. Penguji mengisi *form* penilaian berdasarkan pengalamannya menggunakan web Sistem Informasi Pencatatan *Bug*. Berikut ini merupakan penilaian yang diberikan *tester* setelah mengerjakan skenario yang telah diberikan sebelumnya.



Gambar 4. 28 Hasil penilaian pemilihan *font* yang digunakan

Berdasarkan Gambar 4.28, pertanyaan mengenai pemilihan *font* yang digunakan mendapatkan persentase 100% untuk jawaban ya dan 0% untuk jawaban tidak yang berarti semua penguji setuju bahwa *font* yang digunakan pada situs *web* CDS Bug Base sudah terlihat jelas dan terasa nyaman untuk dibaca oleh pengguna.



Gambar 4. 29 Hasil penilaian pilihan menu

Berdasarkan Gambar 4.29, pertanyaan mengenai pilihan menu yang mudah dipahami mendapatkan persentase 100% untuk jawaban ya dan 0% untuk jawaban tidak. Hal ini menunjukkan bahwa seluruh penguji telah dapat dengan mudah memahami menu-menu pilihan yang ada pada sistem *web* CDS Bug Base.



Gambar 4. 30 Hasil penilaian penggunaan *web*

Berdasarkan Gambar 4.30, pertanyaan mengenai kesulitan pada saat menggunakan *web CDS Bug Base*, jawaban tidak mendapatkan nilai presentase terbesar yaitu 66,7%, sedangkan jawaban ya mendapatkan nilai presentase sebesar 33,3% . Persentase ini menunjukkan bahwa dua dari tiga penguji tidak mengalami atau menemukan adanya kesulitan pada saat menggunakan *web* ini, sedangkan salah seorang penguji lainnya masih menemukan adanya kesulitan.

Berdasarkan pertanyaan sebelumnya, jika Anda menjawab "Ya", jelaskan kesulitan apa yang Anda temui?

1 jawaban

Ketika akan mengubah status sebuah bug yang akan dikerjakan, form yang ada hanya menampilkan kolom status bug dan deskripsi, mungkin bisa diperjelas lagi dengan mencantumkan data dari bug yang dipilih agar di halaman form tersebut ada tertulis keterangan mengenai bug yang sedang dikerjakan.

Semisal kasus yang saya temui tadi, ketika saya sudah mengklik edit salah satu bug, saya tidak tau apakah ini benar bug yang saya maksud atau mungkin saya salah klik bug yang di atas atau di bawahnya.

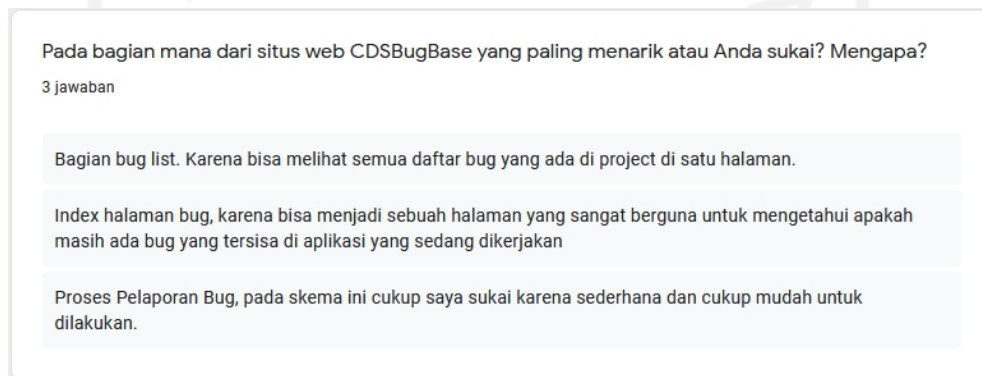
Kemudian pada halaman bug progress, mungkin bisa lebih baik jika ditampilkan juga data orang yang menambahkan status (mungkin dari sisi admin / pm perlu tau mana status dari tester mana status yang ditambahkan developer)

Bisa dipertimbangkan untuk kebutuhan menambahkan lebih dari 1 foto / video evidence bug, mungkin juga attachment evidence dalam bentuk lain seperti pdf atau link google drive

Gambar 4. 31 Kesulitan yang ditemui pengguna

Berdasarkan Gambar 4.31, salah seorang penguji dengan jabatan sebagai *Developer* mengalami kesulitan untuk mengidentifikasi atau mengetahui *bug* mana yang sedang ia ubah

statusnya pada halaman ubah status *bug*. Hal ini dikarenakan pada halaman tersebut tidak tercantum judul *bug* yang sedang diubah statusnya sehingga menyebabkan adanya keraguan atau ketakutan pada pengguna akan terjadi kesalahan pengubahan status suatu *bug*. Selain itu, ia juga menyarankan agar untuk *evidence* tidak terbatas hanya satu foto atau *video* saja dan juga penambahan *evidence* dalam format lainnya. Sebenarnya pada saat penguji menjalankan skenario kasus ini sistem sudah bisa menerima lebih dari satu foto dan *video*, namun karena peneliti belum mencantumkan cara agar dapat mengunggah lebih dari satu foto dan *video* pada *field evidence* dalam satu kali unggahan, mengakibatkan penguji menganggap bahwa sistem hanya dapat menerima satu foto dan *video* saja.



Gambar 4. 32 Bagian *web* yang disukai penguji

Berdasarkan Gambar 4.32, penguji yang merangkap jabatan sebagai Admin dan *Project Manager* menyukai bagian *bug list* karna pada bagian tersebut semua daftar informasi *bug* yang sudah dimasukkan ke sistem tersedia pada satu halaman yang sama. Selanjutnya penguji dengan jabatan sebagai *Tester* juga sama seperti penguji sebelumnya menyukai halaman *bug* namun dengan alasan yang sedikit berbeda. Menurutnya halaman *bug* tersebut sangat berguna karena dapat melakukan pemantauan atau pelacakan apakah masih ada *bug* yang tersisa pada suatu proyek yang sedang dijalankan. Hal ini sudah cukup menunjukkan bahwa tujuan dari pembangunan sistem ini sebagai wadah pelaporan *bug*, penyimpanan data *bug*, dan pemantauan status *bug* telah berhasil dicapai.

Apa pendapat Anda tentang antarmuka dari situs web CDSBugBase?

3 jawaban

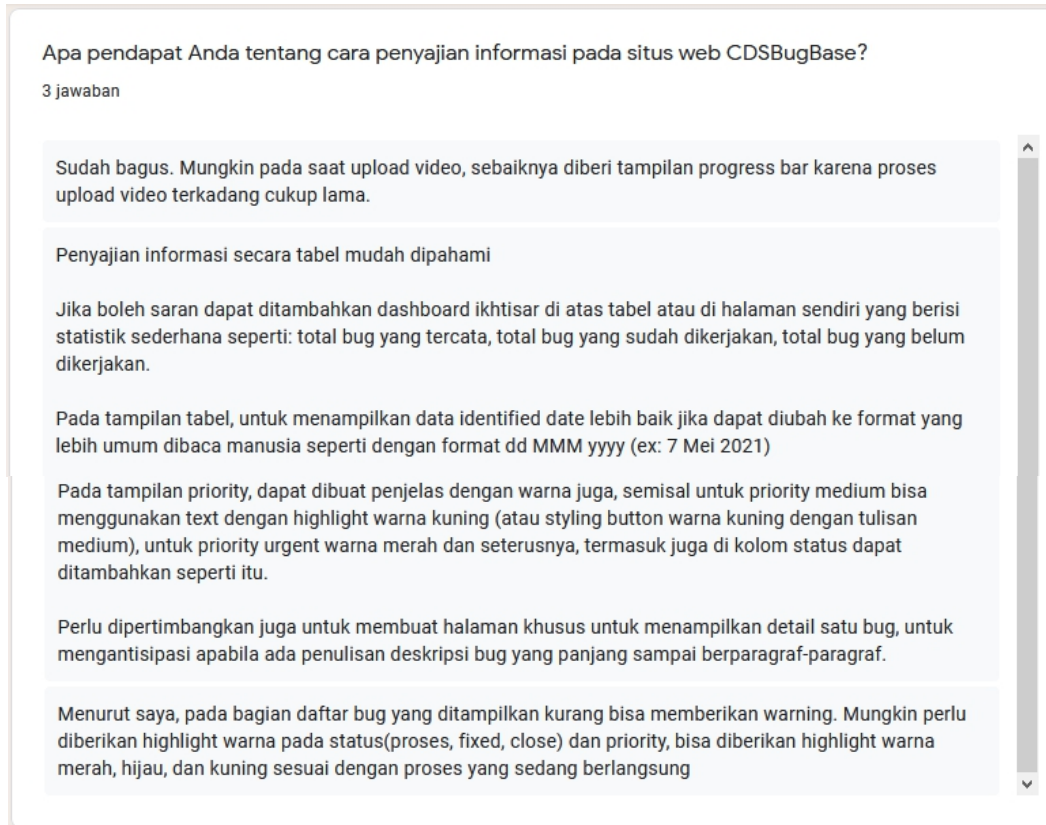
Antarmuka sederhana dan mudah dimengerti.

Antarmuka telah disusun dengan baik dengan kaidah umum aplikasi dashboard, sehingga tidak menyulitkan untuk onboarding pada aplikasi. Pemilihan ukuran & warna pada font dan tombol sudah cukup sesuai untuk membedakan setiap action yang ada

Antar muka pada situs ini cukup sederhana. Kebutuhan yang diberikan oleh situs ini sudah terpenuhi. Untuk kedepannya, mungkin bisa ditambahkan berupa notifikasi email ke setiap pengguna terkait update proses bug. Pada bagian pengisian form untuk foto dan video, bisa ditambahkan informasi mengenai format foto atau video yang harus diunggah dan bisa menambahkan informasi terkait batasan maksimal durasi dan ukuran video. Kemudian, saat proses mengunggah bug bisa diberikan semacam informasi proses upload, semacam pop up mini window yang menampilkan persentase upload agar user bisa mendapat informasi terkait proses upload sedang berlangsung.

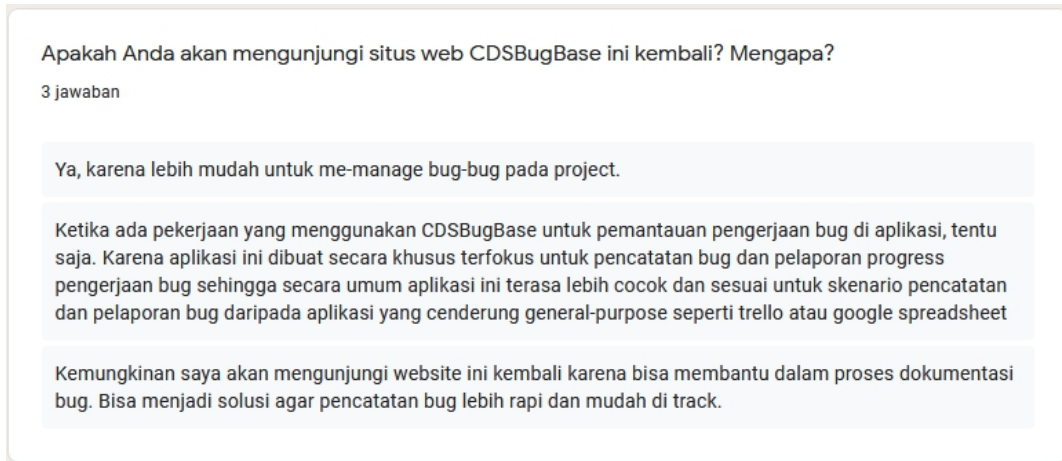
Gambar 4. 33 Pendapat penguji mengenai UI

Berdasarkan Gambar 4.33, penguji dengan jabatan sebagai Admin dan *Project Manager* berpendapat bahwa tampilan antarmuka dari situs *web CDS Bug Base* ini sederhana dan dapat dengan mudah untuk dipahami. Selanjutnya, menurut penguji yang menjalankan skenario kasus *Developer* berpendapat bahwa tampilan antarmuka situs *web CDS Bug Base* ini telah tersusun dengan dengan baik sehingga tidak menyulitkannya. Selain itu, penguji juga berpendapat bahwa penggunaan ukuran serta warna pada *font* dan tombol sudah sesuai. Penguji ketiga dengan jabatan sebagai *Tester* berpendapat senada dengan penguji pertama. Menurutnya tampilan antarmuka situs *web CDS Bug Base* sudah cukup sederhana dan sudah sesuai dengan kebutuhan. Ia menambahkan, untuk kedepannya disarankan untuk menambah fitur seperti notifikasi *email* terkait dengan *update* proses *bug* yang sedang berjalan. Ia juga menyarankan untuk menambahkan *pop up mini window* yang menampilkan proses *uploading* saat pengunggahan suatu *bug* sedang berlangsung.



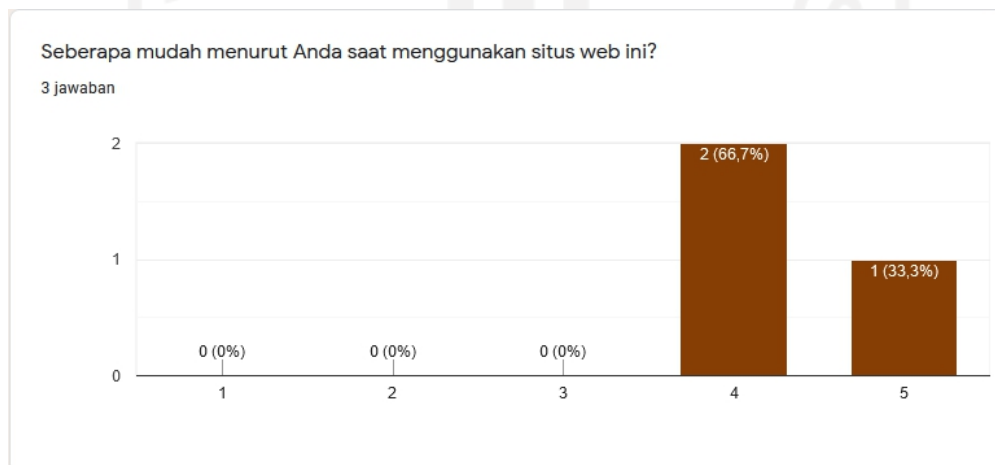
Gambar 4. 34 Pendapat penguji mengenai cara penyajian informasi

Berdasarkan Gambar 4.34, penguji berpendapat bahwa penyajian informasi pada situs *web CDS Bug Base* sudah bagus dan mudah dipahami karena informasi ditampilkan dalam bentuk tabel. Namun penguji juga memberikan kritik seperti pada bagian *bug list* dimana tampilan status dan prioritas *bug* kurang memberikan *warning*, misalnya pemberian *highlight* warna yang berbeda pada tiap macam status dan prioritas *bug* tersebut. Selain memberikan kritik, penguji juga memberikan saran yaitu perubahan format tanggal yang digunakan pada sistem dari format awalnya *yyyy/mm/dd* menjadi *dd/mm/yyyy* seperti format tanggal yang umum digunakan.



Gambar 4. 35 Kemungkinan penguji mengunjungi kembali *web CDS Bug Base*

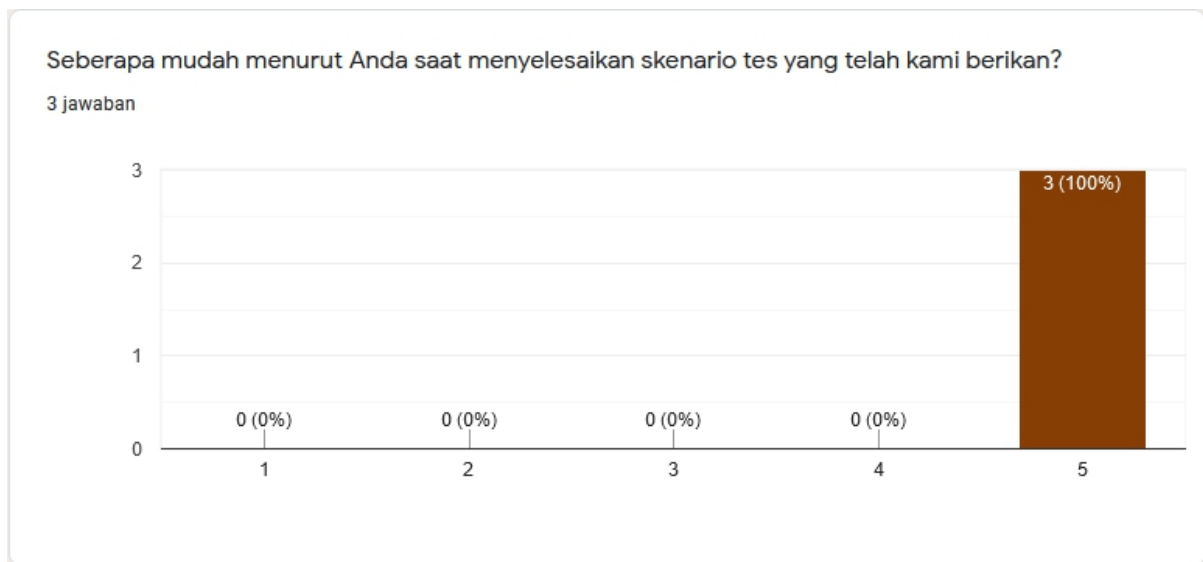
Berdasarkan Gambar 4.35, semua penguji mengatakan bahwa mereka akan kembali mengunjungi situs *web CDS Bug Base* dikemudian hari. Mereka juga memberikan alasan atas jawaban mereka tersebut. Alasannya karena adanya *web CDS Bug Base* ini memudahkan untuk *me-manage bug* yang dijumpai pada proyek, merasa cocok dan sesuai dengan tujuan utama yaitu pemantauan pengerjaan *bug (tracking)*, pencatatan serta pelaporan *bug*, dan terakhir karena *web* ini bisa menjadi solusi agar pencatatan *bug* menjadi lebih rapi.



Gambar 4. 36 Tingkat kemudahan penggunaan *web CDS Bug Base*

Pertanyaan ini merupakan pertanyaan dengan jawaban berupa angka tingkat kemudahan mulai dari satu sampai lima dimana angka satu bernilai sangat tidak mudah, dua bernilai tidak mudah, tiga bernilai netral atau biasa saja, empat bernilai mudah, dan lima bernilai sangat

mudah. Berdasarkan Gambar 4.36, dua dari tiga penguji memberi nilai empat dari lima sebagai nilai tertinggi penilaian, sedangkan seorang penguji lainnya memberi nilai lima. Hal ini dikarenakan kedua penguji masih menemukan sedikit kesulitan saat menggunakan situs *web CDS Bug Base* selaras dengan pertanyaan sebelumnya mengenai adakah temuan kesulitan saat menggunakan *web* ini.



Gambar 4. 37 Tingkat kemudahan menyelesaikan skenario kasus

Sama seperti pertanyaan sebelumnya, pertanyaan ini juga merupakan pertanyaan dengan jawaban berupa angka tingkat kemudahan mulai dari satu sampai lima dimana angka satu bernilai sangat tidak mudah, dua bernilai tidak mudah, tiga bernilai netral atau biasa saja, empat bernilai mudah, dan lima bernilai sangat mudah. Berdasarkan Gambar 4.37, semua penguji sependapat bahwa sangat mudah untuk mengerjakan semua skenario kasus yang telah diberikan sesuai dengan jabatan mereka masing-masing.

4.3.1 Hasil *User Experience Testing* (UX Testing)

Setelah seluruh rangkaian pengujian dilaksanakan, penulis melakukan sedikit perubahan pada sistem mengikuti saran dan anjuran para *tester*. Beberapa perubahan yang dilakukan oleh penulis ialah sebagai berikut.

- a. Penambahan data *field* judul *bug* pada setiap halaman ubah status *bug* untuk memperjelas data *bug* mana yang sedang diubah statusnya.

- b. Penambahan keterangan petunjuk di *form* tambah *bug* pada *field evidence* foto dan *video*.
- c. Pengubahan format tanggal pada sistem sesuai dengan saran yang telah diberikan oleh *tester*.

Berdasarkan penilaian yang telah diberikan oleh UX *tester*, berikut ini merupakan poin-poin kesimpulannya.

- a. Situs *web* CDS *Bug Base* memberi kenyamanan visual bagi para penggunanya. Hal ini selaras dengan hasil pengujian dimana semua penguji setuju bahwa *font* yang digunakan pada situs *web* CDS *Bug Base* sudah jelas dan nyaman dibaca.
- b. Situs *web* CDS *Bug Base* ialah situs *web* dengan tampilan antarmuka yang sederhana dan disukai oleh semua pengguna. Penyajian informasi ditampilkan dalam bentuk tabel yang memuat data lengkap sehingga terkesan rapi, detail, dan mudah untuk dimengerti.
- c. Pengguna merasa situs *web* CDS *Bug Base* ini sudah cocok dan sesuai dengan kebutuhan mereka dan tujuan dari pembangunan sistem ini sendiri yaitu pemantauan pengerjaan *bug* (*tracking*), pencatatan atau penyimpanan data *bug*, serta pelaporan *bug*. Hal ini selaras dengan pernyataan mereka yang akan kembali menggunakan situs *web* ini dikemudian hari saat diperlukan.
- d. Dua dari tiga penguji masih menemukan adanya sedikit kesulitan saat menggunakan *web* CDS *Bug Base*. Penguji juga memberikan beberapa kritik dan saran seperti memperjelas data *bug* mana yang sedang diubah statusnya pada halaman ubah status *bug*, penambahan keterangan ketentuan *upload file* pada *field* tambah foto atau *video*, penambahan notifikasi email terkait *update bug*, pengubahan format tanggal, penambahan *pop up mini window* yang menampilkan presentase *progress upload file*, dan pemberian *warning* pada prioritas *bug*. Sebagian dari kritik dan saran ini telah ditindaklanjuti oleh penulis.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah telah berhasil dibangun Sistem Informasi Pencatatan *Bug* sebagai media pencatatan atau pengumpulan temuan *bug* pada suatu sistem atau aplikasi yang terstruktur dan terperinci. Sistem ini berbasis *website* dengan alamat URL *cdsbugbase.site*, di mana semua informasi yang dibutuhkan dapat disimpan dan diakses secara terpusat. Tampilan *website* ini dibuat sesederhana mungkin agar dapat dengan mudah dipahami oleh semua pengguna sehingga mempercepat proses perbaikan suatu *bug*. Semua informasi terkait temuan *bug* pada *web* ditampilkan dalam bentuk tabel agar lebih rapi dan jelas.

5.2 Saran

Saran ini ditujukan untuk pengembangan Sistem Informasi Pencatatan *Bug* agar kedepannya dapat menjadi sistem yang lebih baik lagi. Berikut ini saran yang dapat diberikan:

- a. Penambahan fitur komunikasi antar pengguna sistem sebagai media diskusi langsung
- b. Penambahan fitur notifikasi agar pengguna sistem dapat mengetahui *update data* terbaru pada sistem.
- c. Penambahan panduan penggunaan pada bagian *sidebar*.
- d. Penambahan *pop up mini window* yang menampilkan presentase *progress upload file*.
- e. Pemberian *highlight* warna yang berbeda pada tiap tipe dan prioritas *bug*.

DAFTAR PUSTAKA

- A. S., Rosa dan Shalahuddin, M. (2013). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Allan. (2005). *Pengertian Internet dan Asal Usul dari Kata Internet*. Surabaya: Penerbit Indah.
- Ardana, Cenik dan Hendro Lukman. (2016). *Sistem Informasi Akutansi*. Jakarta: Mitra Wacana Media.
- Black, Rex. (2009). *Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing*. 03. Indianapolis: Wiley.
- Daulay, Melwin Syafrizal. (2007). *Mengenal Hardware Software dan Pengelolaan Instalasi Komputer*. Yogyakarta: Andi.
- Fanjiang, Y. Y., dan Wu, C. P. (2017). *Automatic Data Logging and Quality Analysis System for Mobile Devices*. Taipei: Hindawi.
- Galin, D. (2004). *Software Quality Assurance: from Theory to Implementation*. In Pearson Education Limited. Pearson Education Limited.
- Hesananda, R., Warnars, H. L. H. S., dan Sianipar, N. F. (2017). *Supervised Classification Karakter Morfologi Tanaman Keladi Tikus (Typhonium Flagelliforme) Menggunakan Database Management System*. *Jurnal Sistem Komputer*, 7(2), 50–58.
- Hirin, A. M. & Virgi. (2011). *Cepat Mahir Pemrograman Web dengan PHP dan MySQL (Level dasar sampai mahir)*. Jakarta: PT Prestasi Pustakaraya.
- Husni. (2007). *Pemrograman Database Berbasis Web*. Yogyakarta: Graha Ilmu.
- Kadir, Abdul. (2008). *Tuntunan Praktis Belajar Database Menggunakan MySQL*. Yogyakarta: Andi.
- Kadir, Abdul. (2014). *Pengenalan Sistem Informasi Edisi Revisi*. Yogyakarta: Andi.
- Kolluri, A.B., Tameezuddin, K., & Gudikandula, K. (2012). *Effective Bug Tracking Systems: Theories and Implementation*. *IOSR Journal of Computer Engineering*, 4, 31-36.
- Krismiaji. (2005). *Sistem Informasi Akuntansi, Edisi Kedua*. Yogyakarta: Akademi Manajemen Perusahaan YKPN.
- Lasminiasih, P, S., Akbar, A., Andriansyah, M., & Utomo, R. B. (2016). *Perancangan Sistem Informasi Kredit Mikro Mahasiswa Berbasis Web*. *Jurnal Sistem Informasi (JSI)*, 8(1), 883–893.

- Laudon, K.C. and Laudon, J.P. (2007) *Management Information Systems: Managing the Digital Firm*. 9th Edition, Pearson/Prentice Hall, Upper Saddle River, 145-278.
- Lukins, S. K., Kraft, N. A., dan Etzkorn, L. H. (2008). *Source Code Retrieval for Bug Localization Using Latent Dirichlet Allocation*. *Proceeding Working Conference on Reverse Engineering*, pp. 155-164.
- Madcoms. (2008). *Aplikasi PHP dan MYSQL Untuk Membuat Website Interaktif*. Yogyakarta: Andi.
- Mulyanto, Agus. (2009). *Sistem Informasi Konsep dan Aplikasi*. Yogyakarta: Pustaka Pelajar.
- Nugroho, A. A., & Setiyawati, N. (2019). *Perancangan Dan Implementasi Aplikasi It Investment Log Berbasis Web (Studi Kasus : PT. XYZ)*. *Jurnal of Business and Audit Information Systems*, 2(1), 38–47.
- Oetomo, D., Budi, S., dkk. (2007). *Pengantar Teknologi Informasi Internet, Konsep dan Aplikasi*. Yogyakarta: Andi.
- Oetomo, Dharma. (2002). *Perencanaan dan Pengembangan Sistem Informasi*. Edisi I. Yogyakarta: Andi.
- Pohan, dan Betha. (2012). *Pemrograman WEB dengan PHP*. Bandung: Informatika Bandung.
- Rizkita, N., Rosely, E., & Nugroho, H. (2018). *Aplikasi Pendaftaran dan Transaksi Pasien Klinik Hewan di Bandung Berbasis Web*. *EProceedings of Applied Science*, 4(3), 1512–1520.
- Roger S, Pressman. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*. Yogyakarta: Andi.
- Romney, Marshall B., dan Paul John Steinbart. (2015). *Accounting Information Systems*, 13th ed. England: Pearson Educational Limited.
- Sergeev, Alexander. (2016). *What is a Bug Tracking System?*, <https://hygger.io/blog/what-is-a-bug-tracking-system>, diakses pada 13 November 2020 pukul 22.45.
- Supriyanto, Aji . (2007). *Pengantar Teknologi Informasi*. Jakarta: Penerbit Salemba Infotek.
- Sutabri, Tata. (2002). *Analisis Sistem Informasi*. Yogyakarta: Andi
- Sutarman. (2009). *Pengantar Teknologi Informasi*. Jakarta: PT Bumi Aksara.
- Telles, M., dan Hseih, Y. (2001). *The Science of Debugging*. Scottsdale: Coriolis.
- Weyuker, E. J., Bell, R. M., dan Ostrand, T. J. (2010). *We're Finding Most of the Bugs, but What are We Missing?*. *Proceeding International Conference on Software Testing*, pp. 313-322.
- Zaki, Ali. (2008). *PHP dan MySQL*. Jakarta: PT Alex Media Komputindo.

LAMPIRAN

