

Pengembangan Web Sistem Informasi Manajemen Layanan Apartemen Modul Penanganan Komplain (Studi Kasus: Apartemen Landmark Residence)

1st Achmad Fahresya Rayhannur
Fakultas Rekayasa Industri
Unitversitas Telkom
Bandung, Indonesia
rayhannur@student.telkomuniversity.ac.id

2nd Ilham Perdana
Fakultas Rekayasa Industri
Unitversitas Telkom
Bandung, Indonesia
ilhamp@telkomuniversity.ac.id

3rd Safara Cathasa Riverinda Rijadi
Fakultas Rekayasa Industri
Unitversitas Telkom
Bandung, Indonesia
safaracathasa@telkomuniversity.ac.id

Abstrak — Penelitian ini mengembangkan sistem informasi manajemen layanan apartemen berbasis web dengan fokus pada penanganan komplain di Apartemen Landmark Residence, yang sebelumnya dilakukan secara manual dan rentan terhadap kerusakan serta kehilangan formulir. Sistem dikembangkan menggunakan metode *Iterative Incremental*, memungkinkan penyesuaian bertahap. Pengujian dilakukan melalui *blackbox*, *load*, dan *cross-browser testing* untuk memastikan fungsionalitas, kinerja pada beban tinggi, dan konsistensi antar browser. Hasilnya, laporan komplain kini dapat dilakukan secara daring, mengurangi risiko kerusakan dan kehilangan. Implementasi sistem ini diharapkan meningkatkan kepuasan penghuni dan memperkuat citra apartemen sebagai hunian yang responsif.

Kata kunci— Manajemen komplain, Apartemen, Sistem informasi berbasis web, *Iterative incremental*.

I. PENDAHULUAN

Di era digital yang terus berkembang, teknologi informasi telah membawa transformasi signifikan dalam berbagai aspek kehidupan, termasuk di sektor properti. Di kota-kota dengan pertumbuhan ekonomi yang pesat seperti Kota Bandung, dampak dari transformasi ini terlihat jelas dengan peningkatan permintaan hunian vertikal. Berdasarkan data dari Badan Pusat Statistik Kota Bandung, 2022, Produk Domestik Regional Bruto (PDRB) Kota Bandung pada tahun 2022 tumbuh sebesar 5,41% dibanding tahun sebelumnya. Peningkatan ini mendorong pertumbuhan permintaan apartemen, dengan indeks permintaan apartemen di Bandung meningkat sebesar 45,8% dari Q3 2021 hingga Q3 2022 [2]. Keterbatasan lahan di kota menjadi pendorong utama penggunaan solusi properti vertikal seperti apartemen [3]. Namun, seiring dengan meningkatnya permintaan, tantangan dalam pengelolaan operasional dan kepuasan penghuni juga turut meningkat. Salah satu aspek yang perlu mendapatkan perhatian adalah pengelolaan komplain penghuni, yang saat ini masih dilakukan secara manual di Apartemen Landmark Residence. Pendekatan ini menyebabkan berbagai masalah

seperti pendataan yang tidak akurat dan tindak lanjut yang tidak terorganisir, yang pada akhirnya mengakibatkan ketidakpuasan penghuni [4].

Penelitian ini bertujuan untuk mengembangkan sistem informasi manajemen layanan apartemen berbasis web yang fokus pada modul penanganan komplain. Sistem ini dirancang dengan metode *Iterative Incremental* yang memungkinkan penyesuaian dan peningkatan sistem secara bertahap berdasarkan umpan balik pengguna. Selain itu, sistem ini dirancang untuk mendokumentasikan komplain dengan lebih efisien, meningkatkan koordinasi antar divisi yang terlibat dalam penanganan komplain, dan memberikan informasi terkini kepada penghuni terkait status komplain mereka. Dalam pengembangannya, web ini dibangun dengan *framework* Laravel, CSS Tailwind, dan komponen Flowbite, dengan pengujian yang dilakukan melalui metode *blackbox*, *load testing*, dan *cross-browser testing* untuk memastikan fungsionalitas dan konsistensi tampilan di berbagai kondisi dan browser.

Hasil dari penelitian ini diharapkan dapat mempermudah Badan Pengelola dalam menangani komplain penghuni, meningkatkan kepuasan penghuni, serta memperkuat citra Apartemen Landmark Residence sebagai hunian yang responsif terhadap kebutuhan penghuninya. Penelitian ini juga memberikan kontribusi bagi mahasiswa, kampus, dan peneliti lainnya sebagai referensi dalam penggunaan sistem informasi di sektor properti, serta menawarkan model solusi yang dapat diadaptasi atau dimodifikasi untuk pengelolaan manajemen di apartemen lainnya.

II. KAJIAN TEORI

A. Pengertian Apartemen

Undang-Undang No. 20 Tahun 2011 tentang Rumah Susun di Indonesia mendefinisikan apartemen sebagai konstruksi bertingkat yang berfungsi sebagai hunian, dengan area dan fasilitas bersama yang digunakan secara kolektif oleh penghuninya. Arsitektur apartemen, menurut Harris (2006) adalah unit hunian terpisah dalam gedung bertingkat yang dilengkapi dengan kamar mandi dan merupakan bagian

dari kumpulan unit serupa. Kedua definisi ini menekankan apartemen sebagai hunian yang terorganisir dan multifungsi, dirancang untuk efisiensi ruang dan kenyamanan bersama dalam lingkungan urban.

B. Manajemen

Manajemen adalah proses administrasi dan koordinasi sumber daya secara efisien untuk mencapai tujuan organisasi [7].

Adapun aktivitas utama pada fungsi manajerial terbagi menjadi lima, yaitu perencanaan, pengorganisasian, penyusunan karyawan, mengarahkan, dan mengontrol [8].

C. Iterative Incremental

Iterative and Incremental Development (IID) adalah metode pengembangan perangkat lunak yang dilakukan secara bertahap. Proses ini dibagi menjadi beberapa iterasi, di mana setiap iterasi melibatkan perencanaan, pembuatan, dan pengujian bagian kecil dari perangkat lunak. Setelah setiap iterasi selesai, tim melanjutkan ke bagian berikutnya sambil memperbaiki dan menambahkan fitur dari iterasi sebelumnya, memungkinkan penyesuaian berdasarkan perubahan atau masukan yang muncul [9].

III. METODE

Penelitian ini dilakukan dengan tujuan membangun web sistem informasi untuk mempermudah Badan Pengelola dalam menerima laporan komplain dan penanganannya. Dalam pengembangannya, sistem ini akan melibatkan beberapa aktor seperti Admin, *Tenant Relation*, dan *Engineer*. Sedangkan pengujian yang akan dilakukan adalah *blackbox*, *load testing*, dan *cross-browser testing*.

Dalam penelitian ini, model *iterative incremental* digunakan sebagai model yang diterapkan untuk mengembangkan sistem. Model *iterative incremental* merupakan salah satu metode pengembangan yang bertahap. Dibagi menjadi beberapa iterasi, setiap iterasinya akan melanjutkan bagian selanjutnya dengan memperbaiki iterasi sebelumnya. Proses setiap iterasi dibagi menjadi beberapa bagian, dimulai dari perencanaan, desain, implementasi, pengembangan, dan pengujian [10]. Penjelasan dari setiap tahap yaitu:

A. Perencanaan

Tahap perencanaan ditujukan untuk menetapkan dan mengidentifikasi fitur-fitur yang akan dibangun pada sistem.

B. Desain

Tahap desain adalah tahap di mana berbagai diagram *Unified Modeling Language (UML)* digambarkan. Pada tahap ini, dibuat *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*.

C. Implementasi

Pengembangan sistem dilakukan menggunakan *framework* Laravel dengan CSS Tailwind dan komponen dari Flowbite. Pada tahap ini, hasil perancangan diagram UML dan *class diagram* diimplementasi dan dikembangkan sampai menjadi produk selesai.

D. Pengujian

Pengujian sistem dilakukan untuk memastikan bahwa tidak ada kesalahan atau *bug* pada sistem yang dikembangkan. Pengujian yang dilakukan adalah *blackbox testing*, *load testing*, dan *cross-browser testing*.

IV. HASIL DAN PEMBAHASAN

Pengembangan sistem dimulai dari tahap perencanaan, desain, implementasi, dan pengujian. Hasil tahap perencanaan adalah berikut:

A. Perencanaan

Pada tahap perencanaan, dihasilkan analisis kebutuhan fungsional dan non-fungsional.

i. Kebutuhan fungsional

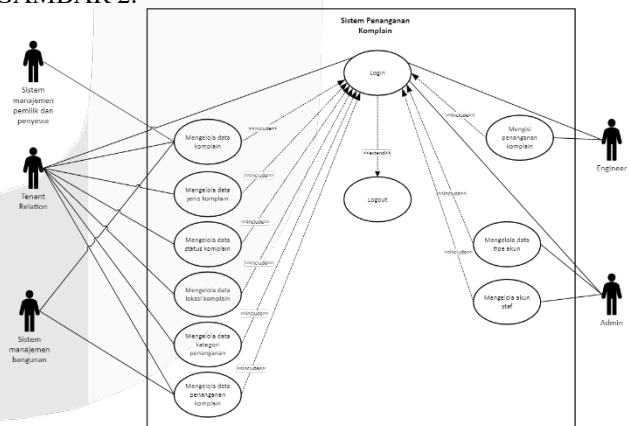
1. Registrasi dan login
2. Mengelola data komplain
3. Mengelola data penanganan
4. Mengelola data akun karyawan
5. Mengelola tabel referensi

ii. Kebutuhan non-fungsional

1. Sistem hanya memberikan akses sesuai hak akses tiap yang sesuai pada tiap pengguna
2. Sistem mampu diakses dengan browser Google Chrome dan Microsoft Edge
3. Sistem mampu berjalan di bawah beban tinggi

B. Desain

Pada tahap desain, dihasilkan diagram-diagram UML seperti *use case diagram* dan *class diagram*. *Use case diagram* sistem penanganan komplain digambarkan pada GAMBAR 1. Sedangkan *Class diagram* digambarkan pada GAMBAR 2.



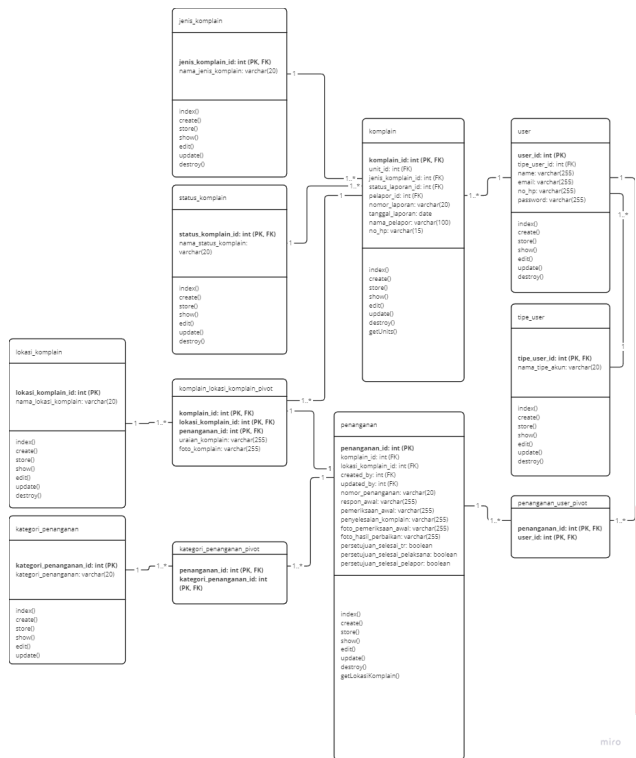
GAMBAR 1

Use Case Diagram Sistem Penanganan Komplain

Ada beberapa tipe pengguna yang dikembangkan pada sistem ini, salah satunya Admin. Setiap tipe pengguna memiliki hak akses yang berbeda. Namun, tipe pengguna Admin dapat mengakses semua halaman. Berikut adalah halaman-halaman yang dapat diakses oleh pengguna berdasarkan halamannya.

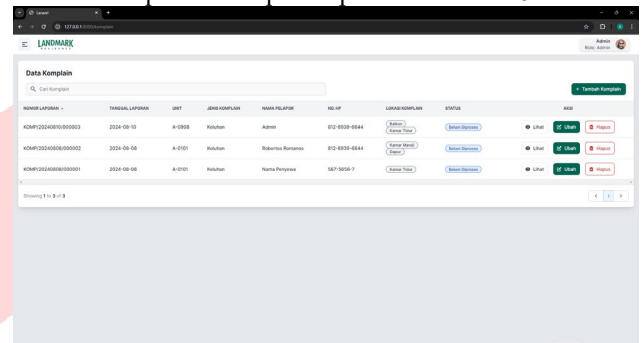
i. Data komplain

Halaman data komplain dapat diakses oleh *Tenant Relation* dan Admin. Pada halaman ini, ditampilkan seluruh laporan komplain yang tersimpan. Aktor dapat menambah, melihat, mengubah, dan menghapus laporan komplain. Halaman data komplain ditampilkan pada GAMBAR 5



GAMBAR 2

Class Diagram Sistem Penanganan Komplain



GAMBAR 5

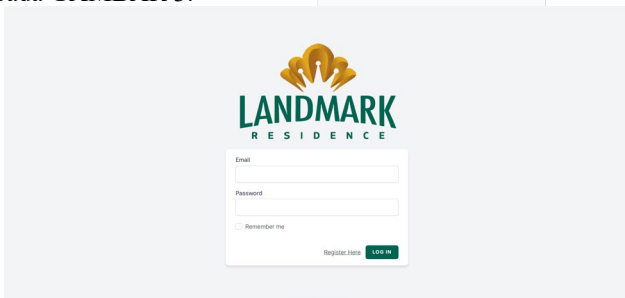
Implementasi Halaman Data Komplain

C. Implementasi

Penelitian ini bertujuan untuk membantu pekerjaan Badan Pengelola dalam mendata dan menangani komplain. Hasil dari tahap desain akan diimplementasikan di tahap ini. Pada sistem yang telah dikembangkan, pengguna akan disambut dengan halaman *login*. Halaman login ditampilkan pada GAMBAR 3.

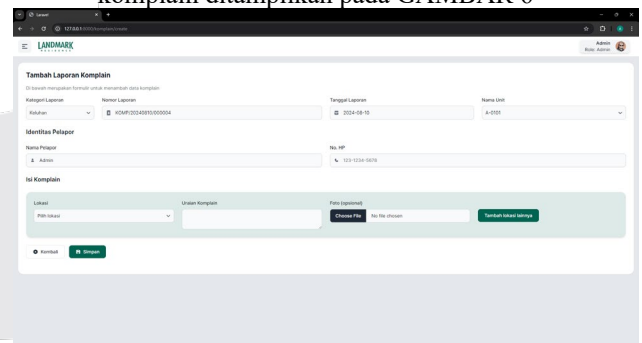
ii. Tambah laporan komplain

Halaman tambah laporan komplain dapat diakses oleh semua tipe pengguna. Pada halaman ini, terdapat beberapa *field* yang harus diisi pelapor serta lokasi di mana komplain ingin dilaporkan. Secara otomatis, penanganan komplain akan terbuat berdasarkan jumlah lokasi komplain. Dengan menyimpan komplain baru, data penanganan komplain akan secara otomatis terbuat berdasarkan jumlah lokasi yang dilaporkan. Halaman tambah komplain ditampilkan pada GAMBAR 6



GAMBAR 3

Implementasi Halaman Login



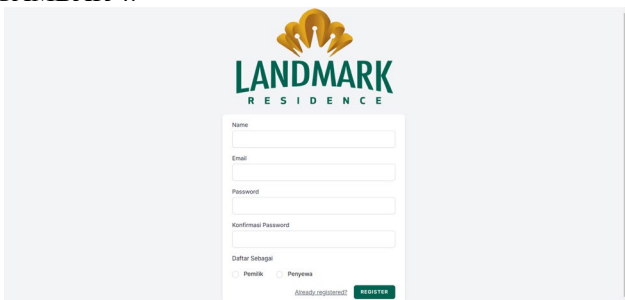
GAMBAR 6

Implementasi Halaman Tambah Komplain

Jika pengguna merupakan seorang pemilik atau penyewa unit dan belum memiliki akun, pengguna dapat masuk ke halaman *register*. Halaman register ditampilkan pada GAMBAR 4.

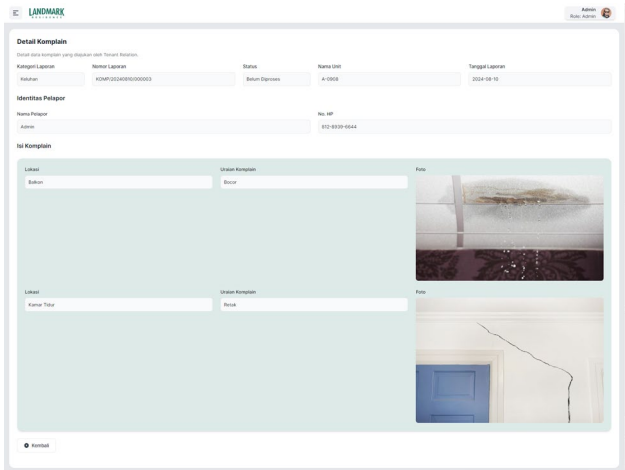
iii. Lihat laporan komplain

Halaman lihat isi komplain dapat diakses oleh *Tenant Relation* dan Admin. Halaman ini hanya menampilkan data yang tersimpan, dan tidak bisa mengubah apapun di dalamnya. Halaman lihat isi komplain ditampilkan pada GAMBAR 7.



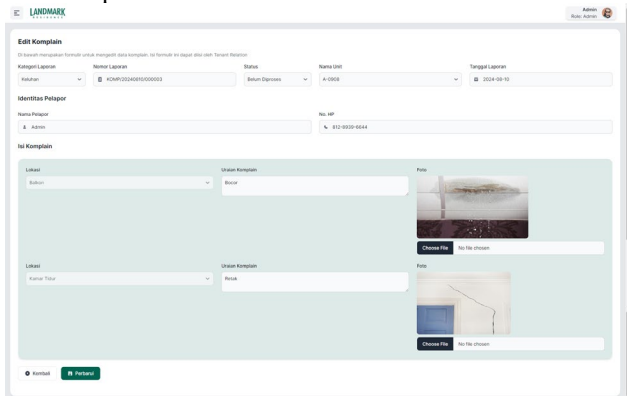
GAMBAR 4

Implementasi Halaman Register



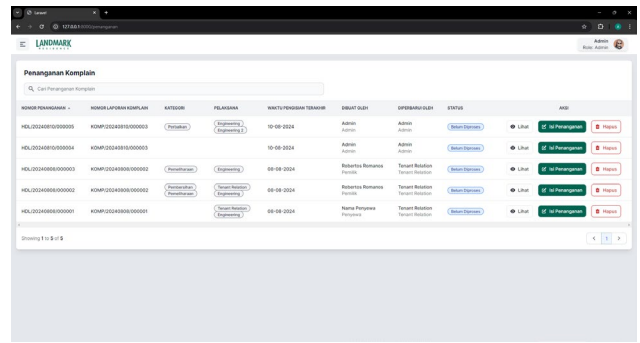
GAMBAR 7
Implementasi Halaman Lihat Isi Komplain

- iv. Ubah isi laporan komplain
Halaman ubah isi komplain dapat diakses oleh *Tenant Relation* dan Admin. Halaman ini berfungsi untuk mengubah isi dari laporan komplain jika diperlukan. Halaman ubah isi komplain ditampilkan pada GAMBAR 8.



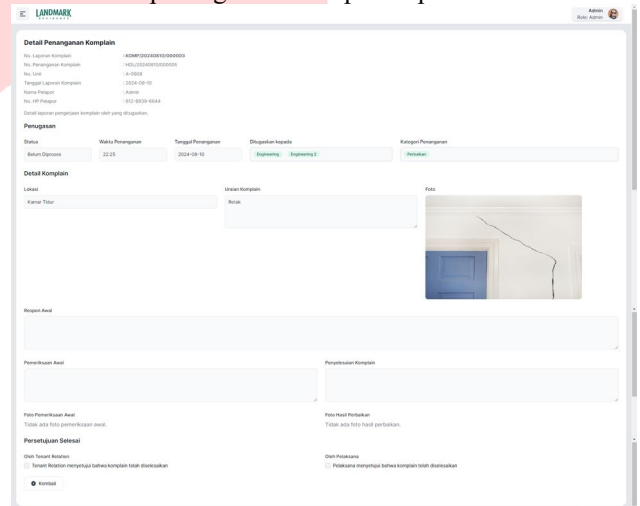
GAMBAR 8
Implementasi Halaman Ubah Isi Komplain

- v. Data penanganan komplain
Halaman data penanganan komplain dapat diakses oleh *Tenant Relation*, Admin, *Engineer*, dan Pelapor. Pada halaman ini, ditampilkan seluruh laporan komplain yang tersimpan. *Tenant Relation* dan Admin dapat menambah, melihat, mengubah, dan menghapus laporan komplain. *Engineer* hanya dapat melihat dan mengubah isi data penanganan komplain yang ia ditugaskan. Sedangkan pelapor hanya dapat melihat penanganan komplain dari komplain yang mereka laporkan. Halaman data penanganan ditampilkan pada GAMBAR 9.



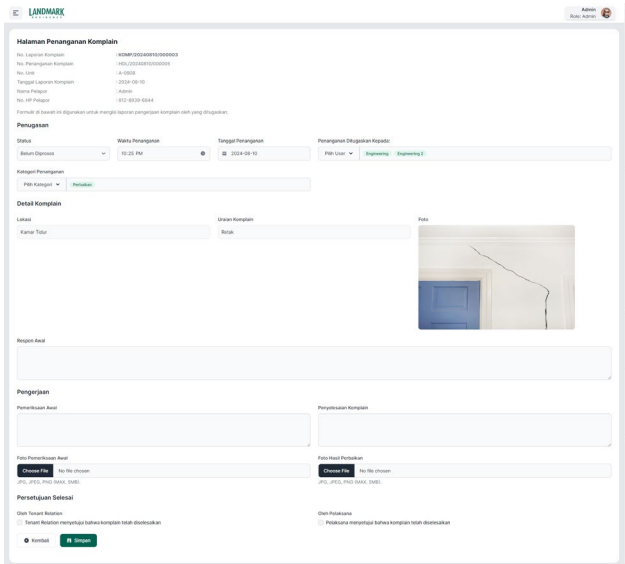
GAMBAR 9
Implementasi Halaman Data Penanganan

- vi. Lihat penanganan komplain
Halaman lihat penanganan komplain dapat diakses sepenuhnya oleh *Tenant Relation*, Admin, *Engineer*, dan Pelapor. Pada halaman ini, hanya ditampilkan data yang sudah tersimpan saja dan tidak bisa mengubah apapun di dalamnya. Halaman lihat penanganan ditampilkan pada GAMBAR 10.



GAMBAR 10
Implementasi Halaman Lihat Penanganan Komplain

- vii. Ubah penanganan komplain
Halaman ubah penanganan komplain dapat diakses oleh *Tenant Relation*, Admin, dan *Engineer*. Halaman ini berfungsi sebagai isi pengerjaan penanganan komplain di mana *Engineer* dapat mencatat pekerjaan yang dilakukan. Halaman data penanganan ditampilkan pada GAMBAR 11.



GAMBAR 11
Implementasi Halaman Ubah Penanganan Komplain

D. Pengujian

Pengujian dilakukan untuk memastikan bahwa sistem dapat berjalan sesuai dengan semestinya. Tujuan dari dilakukannya pengujian ini adalah untuk mencari kesalahan atau *bug* yang terlewat saat pengembangan sistem.

Pengujian sistem yang dilakukan ada tiga, yaitu *blackbox testing*, *load testing*, dan *cross-browser testing*. Pengujian *blackbox* dilakukan untuk memastikan bahwa sistem berjalan seperti seharusnya. *Load testing* dilakukan untuk memastikan bahwa sistem dapat berjalan di bawah *load* puncak. Sedangkan *cross-browser testing* dilakukan untuk memastikan bahwa tampilan web konsisten di browser Google Chrome dan Microsoft Edge.

TABEL 1
Hasil *Blackbox Testing Login dan Logout*

No	<i>Blackbox Testing Login dan Logout</i>		
	Fungsi	Hasil yang diharapkan	Hasil Pengujian
1	Login dengan mengosongkan email dan password	Sistem menolak dan memberi pesan "Harap isi bidang ini" pada input email	Sesuai
2	Login mengisi email dan mengosongkan password	Sistem menolak dan memberi pesan "Harap isi bidang ini" pada input password	Sesuai
3	Login mengisi email tanpa "@" dan mengosongkan password	Sistem menolak dan memberi pesan "Sertakan '@' pada alamat email. 'xxx' pada password"	Sesuai
4	Login mengosongkan email dan mengisi password	Sistem menolak dan memberi pesan "Harap isi bidang ini" pada password	Sesuai
5	Login mengisi email dan password yang salah	Sistem menolak dan memberi pesan "These credentials do not match our records." pada password	Sesuai
6	Login mengosongkan email dan password yang benar	Sistem menerima dan mengarahkan pengguna ke halaman dashboard	Sesuai

No	<i>Blackbox Testing Login dan Logout</i>		
	Fungsi	Hasil yang diharapkan	Hasil Pengujian
7	Logout dari sistem	Sistem melakukan proses keluar (sign out) dan mengarahkan Pengguna ke halaman login.	Sesuai

TABEL 2
Hasil *Blackbox Testing Mengelola Data Komplain*

No	<i>Blackbox Testing Mengelola Data Komplain</i>		
	Fungsi	Hasil yang diharapkan	Hasil Pengujian
1	Membuat komplain baru	Sistem akan menerima pembuatan komplain dan akan menampilkan modal/kotak dialog konfirmasi penyimpanan. Setelah pengguna mengonfirmasi, pengguna diarahkan ke halaman data komplain dengan pesan "Komplain berhasil ditambahkan"	Sesuai
2	Membuat komplain baru dengan lokasi lebih dari satu	Sistem akan menerima pembuatan komplain dan akan menampilkan modal/kotak dialog konfirmasi penyimpanan. Setelah pengguna mengonfirmasi, pengguna diarahkan ke halaman data komplain dengan pesan "Komplain berhasil ditambahkan"	Sesuai
3	Melihat data komplain	Sistem membuka halaman detail komplain pada data komplain yang dipilih.	Sesuai
4	Mengubah detail data komplain	Sistem akan menerima perubahan data komplain dan akan menampilkan modal/kotak dialog konfirmasi penyimpanan. Setelah pengguna mengonfirmasi, pengguna diarahkan ke halaman data komplain dengan pesan "Komplain berhasil ditambahkan"	Sesuai
5	Menghapus data komplain	Sistem akan menerima pembuatan komplain dan akan menampilkan modal/kotak dialog konfirmasi penyimpanan.	Sesuai

No	<i>Blackbox Testing</i> Mengelola Data Komplain		
	<i>Fungsi</i>	<i>Hasil yang diharapkan</i>	<i>Hasil Pengujian</i>
		Setelah pengguna mengonfirmasi, pengguna diarahkan ke halaman data komplain dengan pesan “Komplain berhasil ditambahkan”	

No	<i>Blackbox Testing</i> Mengelola Data Penanganan Komplain		
	<i>Fungsi</i>	<i>Hasil yang diharapkan</i>	<i>Hasil Pengujian</i>
		komplain dengan pesan “Penanganan komplain berhasil dihapus”	

TABEL 3
Hasil *Blackbox Testing* Mengelola Data Penanganan Komplain

No	<i>Blackbox Testing</i> Mengelola Data Penanganan Komplain		
	<i>Fungsi</i>	<i>Hasil yang diharapkan</i>	<i>Hasil Pengujian</i>
1	Membuat penanganan komplain	Sistem akan menerima pembuatan penanganan komplain dan akan menampilkan modal/kotak dialog konfirmasi penyimpanan. Setelah pengguna mengonfirmasi, pengguna diarahkan ke halaman data penanganan komplain dengan pesan “Penanganan komplain — berhasil ditambahkan”	Sesuai
2	Membuat penanganan komplain dengan menugaskan lebih dari satu karyawan	Membuat penanganan komplain dengan menugaskan lebih dari satu karyawan	Sesuai
3	Melihat penanganan komplain	Sistem membuka halaman detail komplain pada data komplain yang dipilih.	Sesuai
4	Mengubah detail penanganan komplain	Sistem akan menerima perubahan isi penanganan komplain dan akan menampilkan modal/kotak dialog konfirmasi penyimpanan. Setelah pengguna mengonfirmasi, pengguna diarahkan ke halaman data penanganan komplain dengan pesan “Penanganan komplain berhasil disimpan”	Sesuai
5	Menghapus data penanganan komplain	Sistem akan menerima pembuatan komplain dan akan menampilkan modal/kotak dialog konfirmasi penyimpanan. Setelah pengguna mengonfirmasi, pengguna diarahkan ke halaman data penanganan	Sesuai

TABEL 4

Hasil Implementasi *Blackbox Testing* Mengelola Data Akun Karyawan

No	<i>Blackbox Testing</i> Mengelola Data Akun Karyawan		
	<i>Fungsi</i>	<i>Hasil yang diharapkan</i>	<i>Hasil Pengujian</i>
1	Membuat akun karyawan dengan mengisi semua input	Sistem akan menolak dan menampilkan pesan “Harap isi bidang ini.”	Sesuai
2	Membuat akun karyawan dengan mengisi semua input, tetapi email tidak menggunakan “@”	Sistem akan menolak dan menampilkan pesan “Sertakan ‘@’ pada alamat email. ‘xxx’ tidak memiliki ‘@’.”	Sesuai
3	Membuat akun karyawan dengan mengisi semua input, tetapi konfirmasi password salah	Sistem akan menolak dan menampilkan pesan “Sesuaikan dengan format yang diminta”.	Sesuai
4	Membuat akun karyawan dengan mengisi semua input dengan benar	Sistem akan menerima pembuatan akun dan akan menampilkan modal/kotak dialog konfirmasi penyimpanan.	Sesuai
5	Mengubah detail akun karyawan dengan mengisi semua input, tetapi email tidak menggunakan “@”	Sistem akan menolak dan menampilkan pesan “Sertakan ‘@’ pada alamat email. ‘xxx’ tidak memiliki ‘@’.”	Sesuai
6	Mengubah detail akun karyawan dengan mengisi semua input, tetapi konfirmasi password salah	Sistem akan menolak dan menampilkan pesan “Sesuaikan dengan format yang diminta”.	Sesuai
7	Mengubah akun karyawan dengan mengisi semua input dengan benar	Sistem akan menerima pembuatan akun dan akan menampilkan modal/kotak dialog konfirmasi penyimpanan.	Sesuai

Berdasarkan hasil *blackbox testing* pada TABEL 1, TABEL 2, TABEL 3, dan TABEL 4, dapat disimpulkan bahwa seluruh fitur dapat bekerja dengan baik, sesuai apa yang diharapkan.

```

C:\Users\asus>ab -n 900 -c 180 http://127.0.0.1:8000/komplain/create
This is ApacheBench, Version 2.3 <Revision: 1913912 >
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Finished 900 requests

Server Software:
Server Hostname: 127.0.0.1
Server Port: 8000

Document Path: /komplain/create
Document Length: 354 bytes

Concurrency Level: 180
Time taken for tests: 131.918 seconds
Complete requests: 900
Failed requests: 0
Non-2xx responses: 900
Total transferred: 1377000 bytes
HTML transferred: 318600 bytes
Requests per second: 6.82 [#/#sec] (mean)
Time per request: 188.843 [ms] (mean)
Time per request: 146.576 [ms] (mean, across all concurrent requests)
Transfer rate: 10.19 [Kbytes/sec] received

Connection Times (ms)
min mean[+/-sd] median max
Connect: 0 0 0.4 0 2
Processing: 171 23772 6166.0 26131 27421
Waiting: 102 23770 6166.2 26130 27420

Percentage of the requests served within a certain time (ms)
50% 26132
60% 26196
75% 26245
80% 26337
90% 26752
95% 27033
98% 27265
99% 27385
100% 27421 (longest request)

```

GAMBAR 12
Load Testing Halaman Register

```

C:\Users\asus>ab -n 900 -c 180 http://127.0.0.1:8000/komplain/create
This is ApacheBench, Version 2.3 <Revision: 1913912 >
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Finished 900 requests

Server Software:
Server Hostname: 127.0.0.1
Server Port: 8000

Document Path: /komplain/create
Document Length: 354 bytes

Concurrency Level: 180
Time taken for tests: 131.918 seconds
Complete requests: 900
Failed requests: 0
Non-2xx responses: 900
Total transferred: 1377000 bytes
HTML transferred: 318600 bytes
Requests per second: 6.82 [#/#sec] (mean)
Time per request: 188.843 [ms] (mean)
Time per request: 146.576 [ms] (mean, across all concurrent requests)
Transfer rate: 10.19 [Kbytes/sec] received

Connection Times (ms)
min mean[+/-sd] median max
Connect: 0 0 0.4 0 2
Processing: 171 23772 6166.0 26131 27421
Waiting: 102 23770 6166.2 26130 27420

Percentage of the requests served within a certain time (ms)
50% 26132
60% 26196
75% 26245
80% 26337
90% 26752
95% 27033
98% 27265
99% 27385
100% 27421 (longest request)

```

GAMBAR 14
Load Testing Menambah Komplain

```

C:\Users\asus>ab -n 900 -c 180 http://127.0.0.1:8000/login
This is ApacheBench, Version 2.3 <Revision: 1913912 >
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Finished 900 requests

Server Software:
Server Hostname: 127.0.0.1
Server Port: 8000

Document Path: /login
Document Length: 353 bytes

Concurrency Level: 180
Time taken for tests: 142.588 seconds
Complete requests: 900
Failed requests: 0
Total transferred: 4213300 bytes
HTML transferred: 319700 bytes
Requests per second: 6.32 [#/#sec] (mean)
Time per request: 2891.630 [ms] (mean)
Time per request: 158.843 [ms] (mean, across all concurrent requests)
Transfer rate: 28.91 [Kbytes/sec] received

Connection Times (ms)
min mean[+/-sd] median max
Connect: 0 0 0.4 0 2
Processing: 158 25660 6840.1 28385 29269
Waiting: 136 25659 6840.2 28384 29269
Total: 159 25660 6840.2 28385 29269

Percentage of the requests served within a certain time (ms)
50% 28385
60% 28569
75% 28666
80% 28762
90% 29042
95% 29151
98% 29210
99% 29239
100% 29269 (longest request)

```

GAMBAR 13
Load Testing Halaman Login

```

C:\Users\asus>ab -n 900 -c 180 http://127.0.0.1:8000/penanganan/create
This is ApacheBench, Version 2.3 <Revision: 1913912 >
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Finished 900 requests

Server Software:
Server Hostname: 127.0.0.1
Server Port: 8000

Document Path: /penanganan/create
Document Length: 354 bytes

Concurrency Level: 180
Time taken for tests: 128.822 seconds
Complete requests: 900
Failed requests: 0
Non-2xx responses: 900
Total transferred: 1377000 bytes
HTML transferred: 318600 bytes
Requests per second: 6.99 [#/#sec] (mean)
Time per request: 25764.337 [ms] (mean)
Time per request: 143.135 [ms] (mean, across all concurrent requests)
Transfer rate: 10.44 [Kbytes/sec] received

Connection Times (ms)
min mean[+/-sd] median max
Connect: 0 0 0.4 0 2
Processing: 143 23174 6193.3 25721 26098
Waiting: 141 23172 6193.4 25720 26098
Total: 143 23174 6193.3 25722 26098

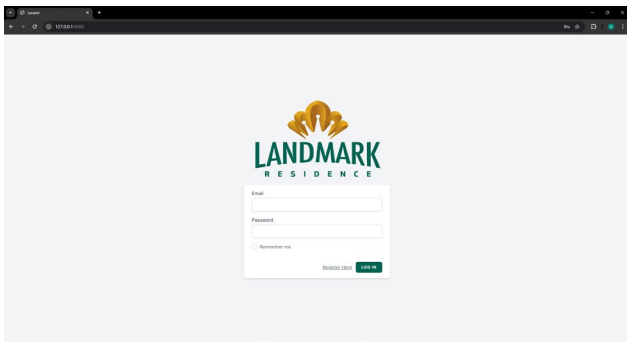
Percentage of the requests served within a certain time (ms)
50% 25722
60% 25799
75% 25836
80% 25871
90% 25986
95% 26022
98% 26065
99% 26085
100% 26098 (longest request)

```

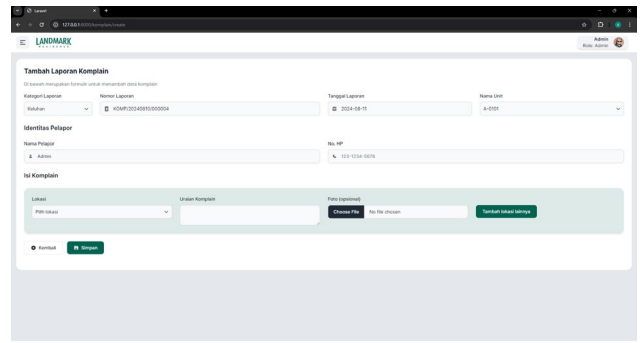
GAMBAR 15
Load Testing Menambah Penanganan

Pada *load testing*, jumlah permintaan yang dikirim (-n) adalah 900 dan jumlah yang dikirim dalam waktu bersamaan (-c) adalah 180.

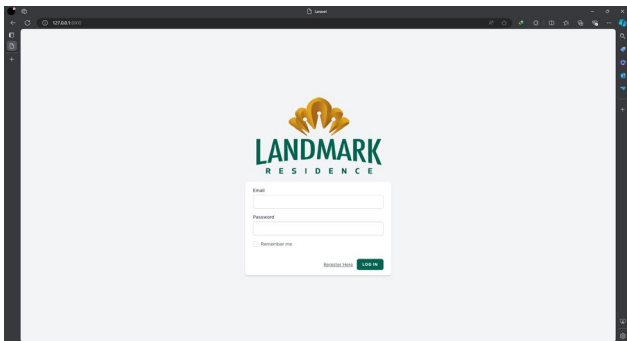
Hasil *load testing* halaman register pada GAMBAR 11 menunjukkan waktu rata-rata per permintaan adalah 26383 ms. Sedangkan, untuk halaman login pada GAMBAR 12 adalah 28501 ms. Untuk hasil *load testing* halaman menambah komplain pada GAMBAR 13, hasil waktu rata-rata per permintaan adalah 26384 ms. Sedangkan, untuk halaman penanganan pada GAMBAR 14 adalah 25764 ms.



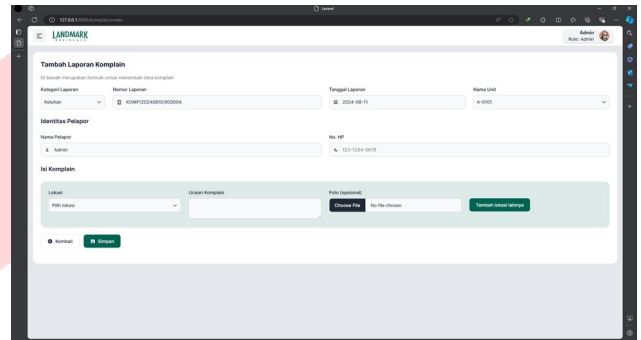
Gambar 16
Pengujian *Cross Browser* Halaman Login pada Google Chrome



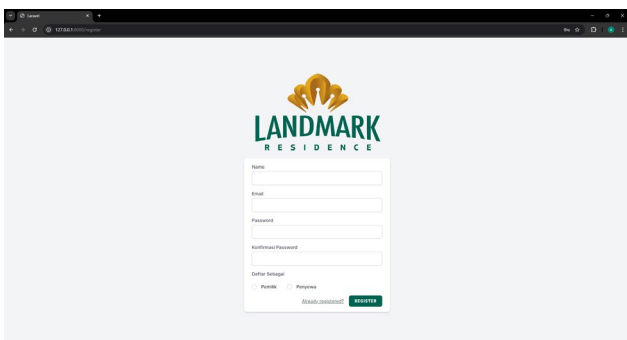
Gambar 20
Pengujian *Cross Browser* Halaman Tambah Komplain pada Google Chrome



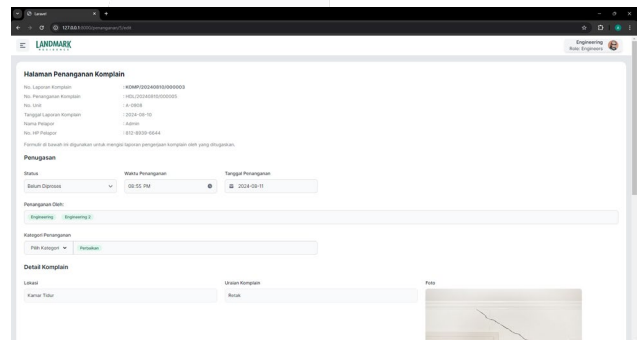
Gambar 17
Pengujian *Cross Browser* Halaman Login pada Microsoft Edge



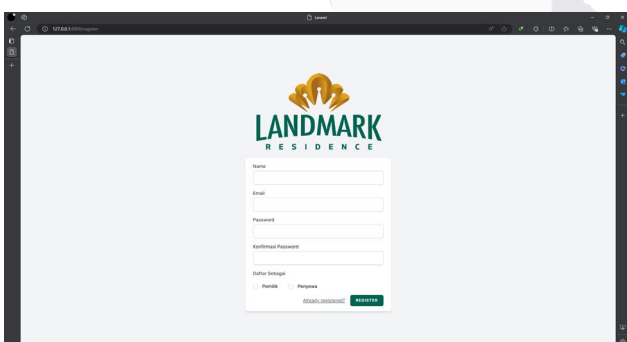
Gambar 21
Pengujian *Cross Browser* Halaman Tambah Komplain pada Microsoft Edge



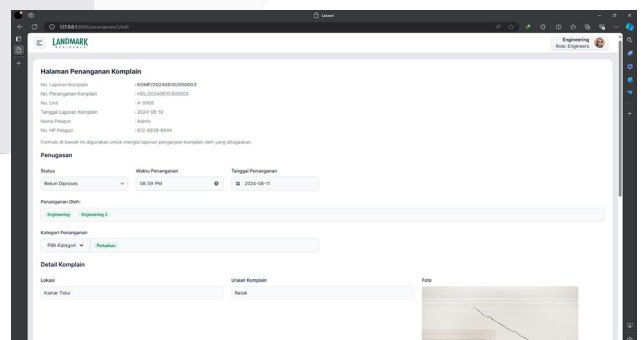
Gambar 18
Pengujian *Cross Browser* Halaman Register pada Google Chrome



Gambar 22
Pengujian *Cross Browser* Halaman Ubah Penanganan Komplain pada Google Chrome



Gambar 19
Pengujian *Cross Browser* Halaman Register pada Microsoft Edge



Gambar 23
Pengujian *Cross Browser* Halaman Ubah Penanganan Komplain pada Microsoft Edge

Dari hasil pengujian *cross browser*, dapat disimpulkan bahwa web yang dikembangkan mampu berjalan dengan browser Google Chrome dan Microsoft Edge dengan tampilan yang konsisten.

V. KESIMPULAN

Berdasarkan hasil pengembangan sistem, penelitian ini berhasil menghasilkan web manajemen layanan apartemen dengan modul penanganan komplain. Sistem ini dapat digunakan oleh *Tenant Relation*, Admin, dan *Engineer* untuk mengelola dan menangani komplain yang dilaporkan. Dengan *Tenant Relation* berperan mengelola data komplain dan penanganan komplain, Admin berperan khususnya mengelola data karyawan dan data dari tabel referensi, dan *Engineer* berperan menangani komplain dengan melakukan perbaikan yang diperlukan dan mencatat perkembangan dari penanganan komplain tersebut agar dapat dipantau oleh Pelapor.

REFERENSI

- [1] Badan Pusat Statistik Kota Bandung, "Pertumbuhan Ekonomi Kota Bandung Tahun 2022," Bandung, 2022.
- [2] PropertyGuru, "Tren Indeks Harga Properti, Apartemen, dan Rumah di Bandung Q2 2018 – Q2 2023," PropertyGuru. Accessed: Dec. 29, 2023. [Online]. Available: https://propertygurumarketplaces.infogram.com/pgid_data-led-content_2022-12_tren-indeks-permintaan-properti-apartemen-dan-rumah-di-bandung-q3-2021-q3-2022-1hxj48pd7nqv52v
- [3] J. Susilo, "Pendekatan Bioklimatik pada Rancangan Apartemen di Bandung," *Jurnal Arsitektur Iternas*, vol. 4, no. 3, pp. 1–10, 2019.
- [4] M. I. Dermawan, I. Aknuranda, and W. H. N. Putra, "Pengembangan Sistem Informasi Penanganan Keluhan Warga Desa Berbasis Web (Studi Kasus Pemerintah Desa Suwayuwo)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 2, pp. 1557–1566, Feb. 2019, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [5] Republik Indonesia, *Undang-Undang No. 20 Tahun 2011 tentang Rumah Susun*. Jakarta: Dewan Perwakilan Rakyat Republik Indonesia, 2011.
- [6] C. M. Harris, *Dictionary of Architecture and Construction*, 4th ed. McGraw-Hill Education, 2006.
- [7] E. A. Benowitz, *CliffsQuickReview Principles of management*. Hungry Minds, Inc, 2001.
- [8] H. Koontz and C. O'Donnell, *Principles of Management: An Analysis of Managerial Functions*. New York: McGraw-Hill, Incorporated, 1955.
- [9] C. Larman, *Agile and Iterative Development: A Manager's Guide*. 2004.
- [10] I. Samuel C, *Software Development Life Cycle (SDLC)*. tutorialspoint, 2018.