

Pengembangan Web Pendataan Pembayaran IPL Apartemen Berbasis Framework Laravel (Studi Kasus: Apartemen Landamark Residence)

1st Sophia Noor Raudhah
Fakultas Rekayasa Industri
Universitas Telkom

Bandung, Indonesia
sophianoorraudhah@student.telkomuni-
versity.ac.id

2nd Dr. Ilham Perdana, S.T., M.T.
Fakultas Rekayasa Industri
Universitas Telkom

Bandung, Indonesia
ilhamp@telkomuniversity.ac.id

3rd Safara Cathasa Riverinda Rijadi,
S.T., M.T.

Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
safaracathasa@telkomuniversity.ac.id

Abstrak — Perekonomian Kota Bandung menunjukkan pertumbuhan yang pesat sejalan dengan perkembangan teknologi informasi yang signifikan di kota-kota besar di Indonesia, seperti Bandung. Salah satu sektor yang terpengaruh adalah sektor properti, terutama hunian apartemen. Perubahan gaya hidup urban yang lebih memprioritaskan kenyamanan dan kepraktisan telah mendorong meningkatnya permintaan terhadap apartemen. Meningkatnya jumlah hunian, memengaruhi proses administrasi di Apartemen Landmark Residence Bandung. Pengelolaan data pemilik, penyewa, dan pembayaran Iuran Pengelolaan Lingkungan (IPL) masih dilakukan secara manual menggunakan aplikasi Microsoft Excel yang mengakibatkan kinerja yang kurang efisien, kurang terorganisir, dan rentan terhadap kesalahan. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem pendataan berbasis web yang efisien, terorganisir, dan terintegrasi untuk menangani administrasi pendataan pemilik, penyewa, dan pembayaran IPL. Metode pengembangan yang digunakan adalah pendekatan iteratif inkremental. Tahap perencanaan meliputi pengumpulan data melalui wawancara dengan pihak apartemen untuk memahami kebutuhan pengguna dan sistem. Tahap desain mencakup pembuatan beragam diagram dan skema basis data yang terstruktur. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan berhasil mengatasi permasalahan administratif dengan baik. Administrasi data pemilik dan penyewa menjadi lebih terorganisir, sementara proses administrasi pembayaran IPL berjalan lebih efisien. Pengujian sistem menunjukkan hasil yang baik dengan fungsionalitas yang sesuai, kemampuan menangani beban tinggi, dan konsistensi tampilan di berbagai browser.

Kata kunci— Web, Pendataan, IPL, Iteratif Inkremental, Laravel

I. PENDAHULUAN

Perekonomian Kota Bandung mengalami kemajuan pertumbuhan yang signifikan. Berdasarkan data BPS Kota Bandung, nilai Produk Domestik Regional Bruto (PDRB) Kota Bandung pada tahun 2023 memiliki nilai yang cukup besar, yaitu Rp351,28 triliun, menjadikannya berada di

peringkat ke-8 di seluruh Indonesia. Berdampingan dengan hal tersebut, perkembangan teknologi informasi di Indonesia, salah satunya Bandung, telah memberi dampak signifikan pada sektor properti, terutama dalam hal hunian apartemen yang ditunjukkan dengan adanya kenaikan indeks permintaan hunian apartemen sebesar 13,4% di kuartal kedua tahun 2023. Tren ini disebabkan oleh peningkatan jumlah penduduk di perkotaan yang cenderung mencari hunian yang praktis dan efisien seperti apartemen. Indeks harga menunjukkan kenaikan sebesar 0,9%, sedangkan penurunan indeks suplai sebesar 0,4%. Artinya, jumlah unit apartemen yang tersedia untuk dijual atau disewa mulai berkurang. Fenomena ini menegaskan pentingnya inovasi dalam sistem manajemen properti.

Contoh nyata dari tren ini adalah Apartemen Landmark Residence Bandung yang dikembangkan oleh Istana Group. Jumlah unit yang besar menambah kompleksitas administrasi bagi Badan Pengelola. Dengan meningkatnya jumlah penghuni, proses administrasi seperti pencatatan data pemilik dan penyewa serta pendataan pembayaran IPL menjadi semakin rumit. Proses administrasi untuk pendataan pemilik dan penyewa serta pembayaran IPL saat ini masih dilakukan secara manual menggunakan aplikasi Microsoft Excel oleh badan pengelola. Hal ini menyebabkan pendataan menjadi kurang efisien. Proses administrasi saat ini yang tidak hanya memakan waktu, tetapi juga rentan terhadap kesalahan dalam hal akurasi, konsistensi, dan keamanan data. Oleh karena itu, ada kebutuhan mendesak akan sistem informasi administrasi pendataan yang lebih efisien dan terorganisir.

Dengan mempertimbangkan kekurangan-kekurangan yang ada, pengembangan aplikasi pendataan berbasis web yang terintegrasi menjadi penting bagi Apartemen Landmark Residence. Saat ini, belum ada sistem yang khusus dirancang untuk mengelola data pemilik, penyewa, dan pembayaran IPL secara efisien. Oleh karena itu, langkah awal yang penting adalah membangun sistem yang mampu mengakomodasi data eksisting dan meningkatkan proses administratif dengan cara yang lebih terorganisir.

Untuk memastikan aplikasi web memenuhi kebutuhan, pengujian yang menyeluruh diperlukan, terutama dengan estimasi pengguna yang banyak. Pengujian *blackbox* mengevaluasi fungsionalitas dari perspektif pengguna, sementara pengujian *load* mengukur kemampuan sistem dalam menangani jumlah pengguna yang tinggi. Selain itu, pengujian *cross browser* memastikan aplikasi berfungsi di beragam *browser* dengan konsisten. Semua pengujian ini bertujuan untuk memastikan keandalan dan fungsionalitas aplikasi dalam berbagai kondisi dan situasi.

II. KAJIAN TEORI

A. Apartemen

Apartemen merupakan tempat tinggal dalam sebuah bangunan yang tinggi yang terdiri atas beberapa lantai atau tingkat. Setiap unit apartemen biasanya memiliki fasilitas seperti ruang tamu, kamar tidur, dapur, dan kamar mandi [1]. Apartemen memiliki fungsi utama sebagai tempat tinggal vertikal dengan titik berat pada kegiatan sehari-hari penghuni, fungsi sekunder sebagai penyedia fasilitas kenyamanan tambahan, seperti fasilitas olahraga, fasilitas kesehatan, area komersial, dan layanan anak, serta fungsi tersier yang terkait dengan pengelolaan bangunan seperti administrasi, pemasaran, dan pemeliharaan [2].

B. Pengelola

Pengelola apartemen adalah entitas yang mengatur dan mengelola operasi sehari-hari apartemen. Terdapat dua bagian pengelola yaitu tim administrasi yang menangani kegiatan administratif apartemen dan tim operasional pengawasan yang bertanggung jawab atas pengawasan dan pengelolaan aspek keamanan, keselamatan (termasuk sistem mekanik dan elektrik), serta penggunaan dan pemeliharaan fasilitas dan perlengkapan bangunan, yang meliputi perawatan bangunan dan *housekeeping* [3].

C. Penghuni Apartemen

Penghuni apartemen adalah individu yang menempati unit atau ruangan tertentu sesuai dengan peraturan perusahaan atau pengelola (kontrak sewa) [4].

D. Iuran Pemeliharaan Lingkungan (IPL)

Iuran Pemeliharaan Lingkungan (IPL) adalah tanggung jawab finansial yang harus dipenuhi oleh penghuni apartemen untuk mendukung pemeliharaan dan operasional fasilitas umum yang disediakan oleh pengelola apartemen [5]. Faktor perhitungan pembentukan IPL didasarkan dua aspek yaitu luas bangunan fasilitas umum apartemen dan pemeliharaan fasilitas umum.

E. Situs Web

Sebuah situs web merupakan sekumpulan halaman yang berada dalam sebuah domain, di mana halaman-halaman tersebut terhubung satu sama lain serta dengan *file-file* terkait untuk menyajikan informasi secara daring [6].

F. Kerangka Kerja Laravel

Laravel memiliki keunggulan pada kemudahan penggunaan serta proses pembelajaran yang mudah diikuti. Laravel memudahkan pengembang dalam menangani

berbagai aspek umum pengembangan web, termasuk pengelolaan *database*, otentikasi pengguna, penjadwalan tugas, pengiriman email, dan penyimpanan *data cache* untuk mempercepat akses web [7].

G. Model Iteratif Inkremental

Metode pengembangan iteratif dan inkremental merupakan sebuah pendekatan yang membagi proyek menjadi serangkaian siklus kecil atau iterasi. Setiap iterasi mencakup beberapa tahapan seperti perencanaan, menganalisis kebutuhan, desain, pengembangan kode, pengujian, evaluasi, dan *deploy*. Metode ini memungkinkan tim proyek untuk memperbaiki dan mengembangkan produk/proyek secara bertahap berdasarkan umpan balik yang diterima di tiap akhir iterasi [8].

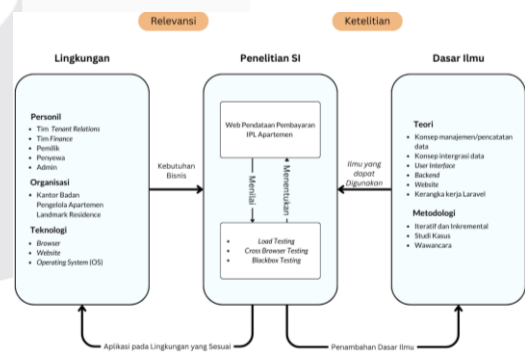
H. Unified Modelling Language (UML)

UML merupakan pemodelan visual untuk merancang sebuah perangkat lunak berbasis objek, termasuk elemen-elemen dan diagram yang mengikuti dasar pemrograman berorientasi objek serta berfungsi sebagai standar mendokumentasikan *blueprint* sistem yang mencakup proses bisnis, struktur kelas pemrograman, skema basis data, dan komponen lain yang dibutuhkan dalam pengembangan perangkat lunak [9]. Pada penelitian ini digunakan beberapa UML yaitu use case diagram, activity diagram, entity relationship diagram, class diagram, dan sequence diagram.

III. METODE

A. Model Konseptual

Model konseptual yang digunakan untuk memahami, mengeksekusi, dan mengevaluasi sebuah penelitian yang terhubung dengan sistem informasi dengan menghubungkan dua pola, yaitu *behaviorial science* dan *design science*. Pola *behaviorial science* berfokus pada pengembangan dan verifikasi teori yang menjelaskan atau memprediksi perilaku manusia atau organisasi. Sementara itu, pola *design science* bertujuan untuk meningkatkan kemampuan manusia dan organisasi melalui penciptaan artefak baru dan inovatif [10].



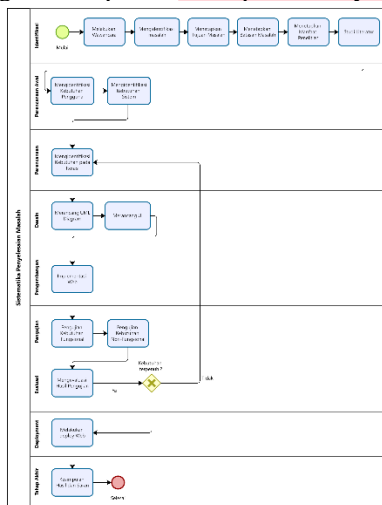
Gambar III. 1 Model Konseptual

Gambar III.1 menjelaskan bahwa model konseptual terdiri atas tiga bagian, yakni lingkungan, penelitian, dan dasar ilmu. Pada bagian lingkungan, aspek personil yang terlibat mencakup Tim *Tenant Relations*, Tim *Finance* yang dapat mengelola pendataan penghuni dan pembayaran IPL. Selain itu, ada pemilik dan penyewa yang dapat mendaftarkan diri sebagai pemilik atau penyewa dan melihat tagihan IPL dari unit yang dimiliki atau disewa. Selain itu, Admin yang dapat mengelola semua. Aspek organisasi terdiri dari kantor Badan Pengelola Apartemen Landmark

Residence, dan yang terakhir aspek teknologi terdiri dari infrastruktur yang digunakan seperti *browser*, web, dan *operating system* (OS). Pada bagian penelitian, web akan dinilai dengan melakukan rangkaian pengujian *load testing*, *cross browser testing*, dan *blackbox testing*. Bagian dasar ilmu, aspek teori terdiri dari konsep manajemen atau pencatatan data, konsep integrasi data, *user interface*, *backend*, web, dan kerangka kerja Laravel. Aspek metodologi meliputi metode yang digunakan, antara lain iteratif dan inkremental, studi kasus, dan wawancara.

B. Sistematika Penyelesaian Masalah

Sistematika penyelesaian masalah adalah sebuah rangkaian prosedur dan langkah-langkah penelitian yang ditujukan untuk mencapai langkah-langkah terstruktur secara tepat, memungkinkan, dan pelaksanaan penelitian yang terukur.



Gambar III. 2 Sistematika Penyelesaian Masalah

Penelitian ini menerapkan metode iteratif dan inkremental. Gambar III.2 merupakan sistematika penyelesaian masalah yang pada bagian pengembangan dibagi menjadi lima tahap sebagai berikut:

1. Tahap Perencanaan

Pada tahap ini, wawancara dilakukan kepada pihak apartemen untuk mengumpulkan data yang kemudian diidentifikasi dan dianalisis untuk memahami kebutuhan web secara menyeluruh. Informasi ini diharapkan mencakup spesifikasi fitur, kebutuhan fungsional, serta preferensi desain dan interaksi pengguna.

2. Tahap Desain

Setelah tahap perencanaan dilakukan, langkah berikutnya adalah memasuki tahap desain, di mana berbagai diagram, skema *database*, dan *user interface* dimodelkan. Detail diagram dan skema *database* yang diperoleh mencakup *use case diagram*, *use case scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*, yang kemudian digunakan untuk membangun basis data yang terstruktur dan rapi menggunakan MySQL

3. Tahap Pengembangan

Setelah tahap perencanaan dan desain diselesaikan, tahap berikutnya melibatkan implementasi pengembangan sistem web yang sesuai dengan kebutuhan pengguna. Kerangka kerja yang digunakan untuk tahap ini adalah Laravel.

4. Tahap Pengujian

Pada tahap ini, pengujian dilakukan terhadap sistem web yang sudah dikembangkan pada tahap sebelumnya. Pengujian dilakukan menggunakan *blackbox testing* untuk menguji kesesuaian antara kebutuhan dan fungsionalitas yang telah dikembangkan. Jika hasil belum sesuai, maka akan dilakukan penyesuaian. Jika hasil sudah sesuai, maka dapat melanjutkan ke tahap berikutnya.

5. Tahap Evaluasi

Pada tahap ini dilakukan evaluasi dari hasil pengerjaan dalam iterasi. Bila terdapat kekurangan maupun perbaikan, maka akan dilakukan pada iterasi berikutnya. Namun, bila tidak ada perbaikan dan sistem dikatakan sudah baik maka iterasi dapat dicukupkan dan siap untuk di-*deploy*. Metode yang digunakan adalah pengujian *blackbox*, pengujian *load*, dan pengujian *cross browser*.

C. Pengumpulan Data

Pemilihan metode pengumpulan data yang tepat merupakan kunci untuk memastikan kualitas data penelitian yang akan diolah agar terjamin keakuratan, relevansi, dan keabsahannya. Pada proses ini digunakan metode pengumpulan data dimulai dari wawancara terhadap Tim *Tenant Relations* untuk mengetahui proses bisnis pendataan pembayaran IPL, mengumpulkan informasi melalui platform daring untuk mendapat atribut pendataan pembayaran IPL oleh tim *finance*, melakukan pengujian *blackbox* dengan bantuan alat browser untuk mengetahui fungsionalita fitur, pengujian *load* untuk menguji kemampuan server dalam menerima beban, serta melakukan pengujian *cross browser* dengan membandingkan respons tampilan dan fungsi fitur di *browser* yang berbeda.

IV. ANALISIS DAN PERANCANGAN

A. Analisis Kebutuhan Pengguna

Analisis kesenjangan dilakukan untuk menentukan solusi dari kebutuhan kondisi yang saat ini terjadi dan dapat berubah sesuai kondisi yang diharapkan. Analisis kesenjangan dilakukan terhadap permasalahan proses pendataan pemilik dan penyewa dan pembayaran IPL dengan kebutuhan sistem web yang akan dikembangkan. Berikut disajikan IV.1 yang meliputi penjelasan analisis kesenjangan.

Tabel IV. 1 Analisis Kesenjangan

| Kondisi Saat Ini | Kebutuhan | Akar Masalah |
|---|---|---|
| Data pemilik, penyewa, dan pembayaran IPL berbentuk Excel dan berada di lembaran yang sama sehingga pengolahan data menjadi kurang efisien dan potensi kesalahan manusia yang tinggi. | Platform pendataan dengan tampilan data dan formulir penambahan data yang sederhana. Selain itu, tempat penyimpanan data pemilik dan penyewa yang terpisah sehingga pengelolaan data dapat dilakukan dengan mudah, terstruktur, dan dapat dipahami. | a. Tidak ada alat yang memenuhi kebutuhan pengelolaan data pemilik dan penyewa dengan cara yang lebih efisien dan mengurangi potensi kesalahan manusia. b. Tidak ada alat yang memenuhi kebutuhan pengelolaan data pembayaran IPL dengan cara yang lebih efisien dan mengurangi potensi kesalahan manusia. Adanya hambatan keterbatasan Microsoft Excel dalam menampilkan data terstruktur dan formulir input yang ramah pengguna. |
| Pemilik yang membayar IPL secara langsung memperlambat proses pendataan karena sering bertanya mengenai jumlah tagihan kepada resepsionis, walaupun tagihan telah dikirim melalui WA. | Platform data pembayaran IPL yang dapat membantu pemilik mendapatkan informasi tagihan IPL secara cepat dan efisien. | Tidak ada alat yang membantu pemilik mengetahui informasi tagihan dengan cepat. |

B. Perancangan Sistem

Perancangan desain sistem dilakukan berdasarkan kebutuhan yang telah diidentifikasi. Rancangan dibuat dan didokumentasikan menggunakan UML (*Unified Modeling Language*).

1. Use Case Diagram

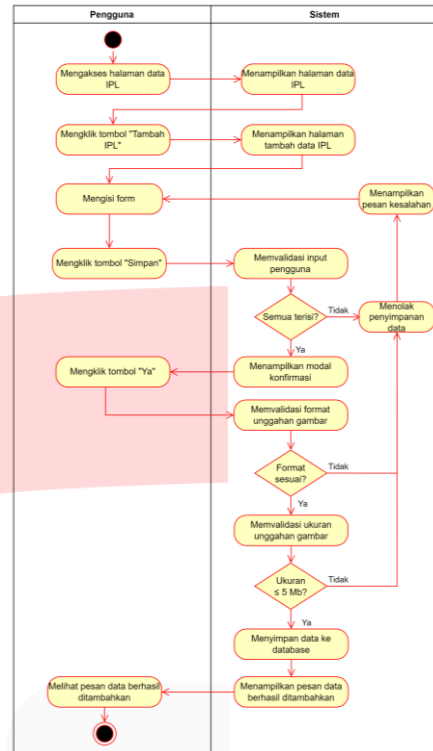
Berikut merupakan *use case diagram* yang menunjukkan fungsionalitas dari aktor sebagai pengguna terhadap sistem.



Gambar IV. 1 Use Case Diagram

2. Activity Diagram

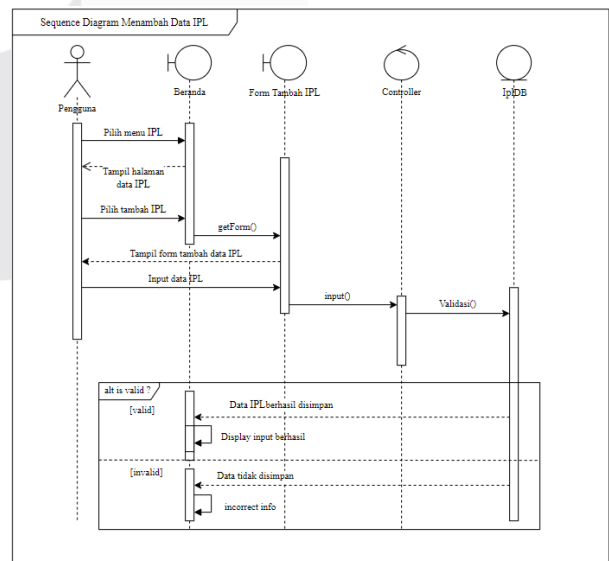
Berikut merupakan gambar *activity diagram* dari proses menambah data IPL yang dapat dilakukan oleh Pemilik, Penyewa, dan Karyawan Tim Finance.



Gambar IV. 2 Activity Diagram Menambah Data IPL

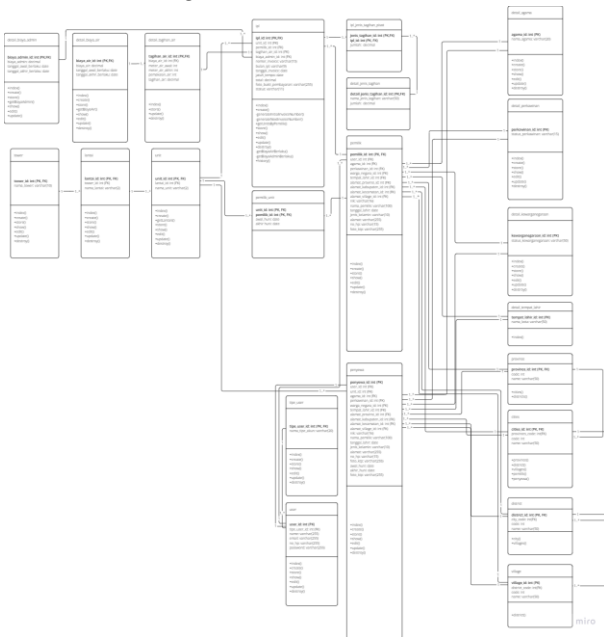
3. Sequence Diagram

Berikut merupakan gambar yang menunjukkan *sequence diagram* dari proses menambah data IPL yang dapat dilakukan oleh Pemilik, Penyewa, dan Karyawan Tim Finance.



Gambar IV. 3 Sequence Diagram Menambah Data IPL

4. Class Diagram



Gambar IV. 4 Class Diagram

V. IMPLEMENTASI DAN PENGUJIAN

A. Tahap Perencanaan

Tahap perencanaan merupakan langkah pertama yang dilakukan pada suatu iterasi. Identifikasi kebutuhan yang dipilih dalam iterasi kedua mencakup fungsi yang belum dipenuhi di iterasi sebelumnya

1. Iterasi Pertama

Berikut merupakan kebutuhan fungsional pada iterasi pertama :

Tabel V. 1 Kebutuhan Fungsional Iterasi Pertama

| Kebutuhan | Deskripsi |
|------------------------|--|
| Login | Proses semua pengguna masuk ke akun yang dimiliki dengan menyertakan informasi login yang sesuai. |
| Registrasi Akun | Proses pemilik dan penyewa membuat akun baru |
| Logout | Proses semua pengguna keluar dari akun yang dimiliki. |
| Menambah Data Pemilik | Proses pemilik, karyawan Tim <i>Tenant Relations</i> , dan admin menambahkan data pemilik. |
| Melihat Data Pemilik | Proses pemilik, karyawan Tim <i>Tenant Relations</i> , karyawan Tim <i>Finance</i> , dan admin melihat data pemilik. |
| Mengubah Data Pemilik | Proses karyawan Tim <i>Tenant Relations</i> dan admin mengubah data pemilik. |
| Menghapus Data Pemilik | Proses karyawan Tim <i>Tenant Relations</i> dan admin menghapus data pemilik. |
| Menambah Data Penyewa | Proses penyewa, karyawan Tim <i>Tenant Relations</i> , dan admin menambahkan data penyewa. |

| | |
|----------------------------------|--|
| Melihat Data Penyewa | Proses penyewa, karyawan Tim <i>Tenant Relations</i> , karyawan Tim <i>Finance</i> , dan admin melihat data penyewa. |
| Mengubah Data Penyewa | Proses karyawan Tim <i>Tenant Relations</i> dan admin mengubah data penyewa. |
| Menghapus Data Penyewa | Proses karyawan Tim <i>Tenant Relations</i> dan admin menghapus data penyewa. |
| Menambah Data Tower | Proses karyawan Tim <i>Tenant Relations</i> dan admin menambahkan data tower. |
| Melihat Data Tower | Proses karyawan Tim <i>Tenant Relations</i> dan admin melihat data tower. |
| Mengubah Data Tower | Proses karyawan Tim <i>Tenant Relations</i> dan admin mengubah data tower. |
| Menghapus Data Tower | Proses karyawan Tim <i>Tenant Relations</i> dan admin menghapus data tower. |
| Menambah Data Lantai | Proses karyawan Tim <i>Tenant Relations</i> dan admin menambahkan data lantai. |
| Melihat Data Lantai | Proses karyawan Tim <i>Tenant Relations</i> dan admin melihat data lantai. |
| Mengubah Data Lantai | Proses karyawan Tim <i>Tenant Relations</i> dan admin mengubah data lantai. |
| Menghapus Data Lantai | Proses karyawan Tim <i>Tenant Relations</i> dan admin menghapus data lantai. |
| Menambah Data Unit | Proses karyawan Tim <i>Tenant Relations</i> dan admin menambahkan data unit. |
| Melihat Data Unit | Proses karyawan Tim <i>Tenant Relations</i> dan admin melihat data unit. |
| Mengubah Data Unit | Proses karyawan Tim <i>Tenant Relations</i> dan admin mengubah data unit. |
| Menghapus Data Unit | Proses karyawan Tim <i>Tenant Relations</i> dan admin menghapus data unit. |
| Menambah Data Status Perkawinan | Proses admin menambahkan data status perkawinan. |
| Melihat Data Status Perkawinan | Proses admin melihat data status perkawinan. |
| Mengubah Data Status Perkawinan | Proses admin mengubah data status perkawinan. |
| Menghapus Data Status Perkawinan | Proses admin menghapus data status perkawinan. |
| Menambah Data Agama | Proses admin menambahkan data agama. |
| Melihat Data Agama | Proses admin melihat data agama. |
| Mengubah Data Agama | Proses admin mengubah data agama. |
| Menghapus Data Agama | Proses admin menghapus data agama. |

2. Iterasi Kedua

Berikut merupakan kebutuhan fungsional pada iterasi kedua, dimana terdapat penambahan fitur sebagai berikut:

Tabel V. 2 Kebutuhan Fungsional Iterasi Kedua

| Kebutuhan | Deskripsi |
|----------------------------|---|
| Menambah Data IPL | Proses karyawan Tim <i>Finance</i> dan admin menambahkan data IPL. |
| Melihat Data IPL | Proses pemilik, penyewa, karyawan Tim <i>Finance</i> dan admin melihat data IPL. |
| Mengubah Data IPL | Proses pemilik, penyewa, karyawan Tim <i>Finance</i> , dan admin mengubah data IPL. |
| Menghapus Data IPL | Proses karyawan Tim <i>Finance</i> dan admin menghapus data IPL. |
| Menambah Data Biaya Air | Proses karyawan Tim <i>Finance</i> dan admin menambahkan data biaya air. |
| Melihat Data Biaya Air | Proses karyawan Tim <i>Finance</i> dan admin melihat data biaya air. |
| Mengubah Data Biaya Air | Proses karyawan Tim <i>Finance</i> dan admin mengubah data biaya air. |
| Menghapus Data Biaya Air | Proses karyawan Tim <i>Finance</i> dan admin menghapus data biaya air. |
| Menambah Data Biaya Admin | Proses karyawan Tim <i>Finance</i> dan admin menambahkan data biaya admin. |
| Melihat Data Biaya Admin | Proses karyawan Tim <i>Finance</i> dan admin melihat data biaya admin. |
| Mengubah Data Biaya Admin | Proses karyawan Tim <i>Finance</i> dan admin mengubah data biaya admin. |
| Menghapus Data Biaya Admin | Proses karyawan Tim <i>Finance</i> dan admin menghapus data biaya admin. |

B. Tahap Design

Tahap desain merupakan langkah kedua pada suatu iterasi. Setelah kebutuhan sudah ditentukan pada tahap perencanaan, selanjutnya menganalisis dan merancang sistem sesuai dengan kebutuhan. Hasil dari tahap ini mencakup rancangan UML (*Unified Modeling Language*) yang dijelaskan lebih lanjut dalam Bab IV.

C. Tahap Pengembangan

Tahap pengembangan merupakan langkah ketiga pada suatu iterasi. Tahap ini dilakukan untuk mengembangkan web sesuai hasil yang diperoleh dari tahap desain.

1. Iterasi Pertama

Berikut merupakan halaman menambah data pemilik pada iterasi pertama:

Gambar V. 1 Halaman Menambah Data Pemilik

2. Iterasi Kedua

Berikut merupakan halaman menambah data IPL pada iterasi kedua:

Gambar V. 2 Halaman Menambah Data IPL

D. Tahap Pengujian

Tahap pengujian merupakan langkah keempat pada suatu iterasi. Pengujian yang dilakukan antara lain pengujian *blackbox*, pengujian *load*, dan pengujian *cross browser*.

1. Iterasi Pertama

Pengujian *load* dilakukan untuk memastikan kemampuan sistem dalam menghadapi beban tertentu. Berikut merupakan hasil pengujian *load* pada fitur menambah data pemilik pada iterasi pertama:

```

C:\Users\asus>ab -n 900 -c 45 http://127.0.0.1:8080/penyewa/create
This is ApacheBench, Version 2.3 <Revision: 1913912 >
Copyright 1996 Adan 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: 8080

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

Concurrency Level: 45
Time taken for tests: 148.766 seconds
Complete requests: 900
Failed requests: 0
Non-2xx responses: 900
Total transferred: 1377000 bytes
HTML transferred: 318600 bytes
Requests per second: 6.05 [#/sec] (mean)
Time per request: 2430.276 [ms] (mean)
Time per request: 165.235 [ms] (mean, across all concurrent requests)
Transfer rate: 9.04 [Kbytes/sec] received

Connection Times (ms)
min mean[+-sd] median max
Connect: 0 0 0.4 0 2
Processing: 170 7255 959.5 7417 8082
Waiting: 169 7253 959.6 7415 8080
Total: 170 7255 959.5 7417 8083

Percentage of the requests served within a certain time (ms)
50% 7417
66% 7447
75% 7469
80% 7494
90% 7536
95% 7673
99% 7950
100% 8083 (longest request)

```

Gambar V. 3 Hasil Pengujian Load Menambah Data Pemilik

2. Iterasi Kedua

Berikut merupakan hasil pengujian *load* pada fitur menambah data biaya air pada iterasi kedua:

```

C:\Users\ASUS>ab -n 900 -c 45 http://127.0.0.1:8080/detail_biaya_air/create
This is ApacheBench, Version 2.3 <Revision: 1913912 >
Copyright 1996 Adan 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: 8080

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

Concurrency Level: 45
Time taken for tests: 166.899 seconds
Complete requests: 900
Failed requests: 0
Non-2xx responses: 900
Total transferred: 1377000 bytes
HTML transferred: 318600 bytes
Requests per second: 5.39 [#/sec] (mean)
Time per request: 8304.974 [ms] (mean)
Time per request: 185.444 [ms] (mean, across all concurrent requests)
Transfer rate: 8.06 [Kbytes/sec] received

Connection Times (ms)
min mean[+-sd] median max
Connect: 0 0 0.4 0 2
Processing: 202 8143 1329.2 8846 11264
Waiting: 200 8141 1329.2 8845 11262
Total: 202 8143 1329.3 8847 11264

Percentage of the requests served within a certain time (ms)
50% 8847
66% 8396
75% 8668
80% 8838
90% 9486
95% 10825
98% 10667
99% 10885
100% 11264 (longest request)

```

Gambar V. 4 Hasil Pengujian Load Menambah Data Biaya Air

E. Tahap Evaluasi

Pada tahap evaluasi menghasilkan keberhasilan iterasi pertama dan kedua berjalan dengan baik dan sesuai harapan. Pengembangan sistem dan semua fitur dapat digunakan oleh pengguna sesuai dengan fungsinya. Fitur-fitur yang dibangun berhasil diimplementasikan dan diuji dengan baik, dan respons yang diperoleh sesuai dengan spesifikasi yang ditetapkan. Dengan keberhasilan ini, fase iterasi dicukupkan sampai fase kedua.

F. Tahap Deploy

Tahap *deployment* atau penyebaran merupakan tahap terakhir dari metode iteratif inkremental. Tahapan ini melibatkan proses *hosting* aplikasi web pendataan pembayaran IPL yang dibangun. Proses *hosting* menggunakan layanan dari Rumahweb Indonesia yang memungkinkan web dapat diakses oleh pengguna.

VI. KESIMPULAN

Berdasarkan hasil pengembangan aplikasi web sistem pendataan pembayaran IPL, dapat disimpulkan bahwa sistem berhasil dibangun dalam dua iterasi hingga di-*deploy*. Proses administrasi pendataan pemilik dan penyewa berhasil dikembangkan dan berjalan dengan efisien dan terorganisir. Selain itu, proses administrasi pendataan pembayaran IPL berhasil dikembangkan dan berjalan dengan efisien. Semua fitur yang dibangun untuk memenuhi kebutuhan pengguna dapat berfungsi dengan baik. Hal ini disebabkan telah dilakukan beragam pengujian, seperti pengujian *blackbox*, *load*, dan *cross browser*. Ketiga pengujian tersebut dilakukan di kedua iterasi dan hasil dari tiap pengujian menunjukkan hasil yang sesuai dan baik. *Test case* pada pengujian *blackbox* memperoleh hasil semua sesuai. Dalam pengujian *load*, dengan menempatkan beban yang tinggi, sistem masih dapat berjalan dengan baik. Selain itu, pengujian *cross browser* berhasil membuktikan dan menampilkan konsistensi tampilan halaman dan respons dari input yang dibuktikan pada 2 *browser* berbasis Chromium.

REFERENSI

- [1] F. A. Saputra and A. Iskandar, "Sistem Pendukung Keputusan Pemilihan Apartemen Terbaik Menerapkan Metode TOPSIS dan Pembobotan ROC," *Journal of Information System Research (JOSH)*, vol. 5, no. 1, pp. 142–150, Oct. 2023, doi: 10.47065/josh.v5i1.4434.
- [2] C. A. P. Kusuma, "Perancangan Apartemen pada Area Kabupaten Karawang dengan Prinsip Healthy Building," Universitas Atma Jaya Yogyakarta, Yogyakarta, 2023.
- [3] F. P. Andini, "Biophilic High-End Apartment yang Berfungsi sebagai Hutan Jakarta," Institut Teknologi Sepuluh Nopember, Surabaya, 2018.
- [4] R. G. A. D. Atmadja, "Rancangan Apartemen Comfy di Kabupaten Bandung," Institut Teknologi Nasional Bandung, Bandung, 2019.
- [5] M. I. Akbar and T. U. Atmoko, "Iuran pemeliharaan lingkungan apartemen = Apartment service charge," Universitas Indonesia, Depok, 2019.
- [6] M. V. Al Hasri and E. Sudarmilah, "Sistem Informasi Pelayanan Administrasi Kependudukan Berbasis Website Kelurahan Banaran," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 2, pp. 249–260, May 2021, doi: 10.30812/matrik.v20i2.1056.
- [7] M. Stauffer, *Laravel: Up & Running, 3rd Edition*, 3rd ed. O'Reilly Media, Inc, 2023.
- [8] C. Larman and V. R. Basili, "Iterative and Incremental Developments. A Brief History,"

- [9] *Computer (Long Beach Calif)*, vol. 36, no. 6, pp. 47–56, Jun. 2003, doi: 10.1109/MC.2003.1204375.
- A. Nugroho, *Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek*. Bandung: Informatika Bandung, 2005.
- [10] Hevner, March, Park, and Ram, “Design Science in Information Systems Research,” *MIS Quarterly*, vol. 28, no. 1, p. 75, 2004, doi: 10.2307/25148625.

