

# **LAPORAN AKHIR**

## **APLIKASI DATA KEPEGAWAIAN**



Disusun Oleh :

2201020091 - Yohani Natalia Simanulang

2201020083 - Winda Aulia Ariyani

2201020068 - Kirana Putri Fercia

2201020071 - Linda Astuti

2201020086 - Tanra Wija Yanti

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN  
UNIVERSITAS MARITIM RAJA ALI HAJI**

**2024**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>DAFTAR GAMBAR.....</b>	<b>4</b>
<b>DAFTAR TABEL.....</b>	<b>5</b>
<b>DAFTAR SIMBOL .....</b>	<b>6</b>
<b>BAB I.....</b>	<b>7</b>
<b>PENDAHULUAN.....</b>	<b>7</b>
<b>1.1 Latar Belakang .....</b>	<b>7</b>
<b>1.2 Tujuan Proyek .....</b>	<b>8</b>
<b>BAB II.....</b>	<b>9</b>
<b>METODE .....</b>	<b>9</b>
<b>2.1 Pengumpulan Data .....</b>	<b>9</b>
<b>2.2 Jadwal Proyek dan Pembagian Kerja.....</b>	<b>9</b>
<b>2.3 Alur Proyek.....</b>	<b>10</b>
<b>BAB III .....</b>	<b>14</b>
<b>HASIL DAN PEMBAHASAN .....</b>	<b>14</b>
<b>3.1 Deskripsi Perangkat Lunak.....</b>	<b>14</b>
<b>3.2 Metode Pengembangan Perangkat Lunak.....</b>	<b>15</b>
<b>3.3 Analisis Kebutuhan .....</b>	<b>16</b>
<b>3.3.1 Kebutuhan Fungsional.....</b>	<b>16</b>
<b>3.3.2 Kebutuhan Non-Fungsional .....</b>	<b>18</b>
<b>3.4 Perancangan Perangkat Lunak .....</b>	<b>21</b>
<b>3.4.1 Perancangan Sistem .....</b>	<b>21</b>
<b>3.4.2 Perancangan Antarmuka (UX/UI) .....</b>	<b>46</b>

<b>3.5</b>	<b>Implementasi Perangkat Keras dan Lunak .....</b>	<b>52</b>
<b>3.5.1</b>	<b>Implementasi Perangkat Keras.....</b>	<b>52</b>
<b>3.5.2</b>	<b>Implementasi Perangkat Lunak .....</b>	<b>52</b>
<b>3.5.3</b>	<b>Produk Perangkat Lunak .....</b>	<b>53</b>
<b>3.6</b>	<b>Pengujian Perangkat Lunak .....</b>	<b>54</b>
<b>3.6.1</b>	<b>Blackbox Testing .....</b>	<b>54</b>
<b>BAB IV</b>	<b>.....</b>	<b>56</b>
<b>KESIMPULAN DAN SARAN</b>	<b>.....</b>	<b>56</b>
<b>4.1</b>	<b>Kesimpulan .....</b>	<b>56</b>
<b>4.2</b>	<b>Saran.....</b>	<b>56</b>

## DAFTAR GAMBAR

Gambar 3 1 flowchart admin .....	21
Gambar 3 2 Use Case Diagram .....	23
Gambar 3 3 Activity Diagram mencari .....	25
Gambar 3 4 Activity Diagram Detail .....	26
Gambar 3 5 Activity Diagram Edit .....	27
Gambar 3 6 Activity Diagram Tambah .....	29
Gambar 3 7 Activity Diagram Hapus .....	31
Gambar 3 8 Class Diagram .....	33
Gambar 3 9 Object Diagram .....	36
Gambar 3 10 Sequence Diagram login .....	39
Gambar 3 11 Sequence Diagram melihat data pegawai .....	40
Gambar 3 12 Sequence Diagram melihat riwayat kerja .....	40
Gambar 3 13 Sequence Diagram mengelola data pegawai .....	41
Gambar 3 14 Tampilan Login .....	46
Gambar 3 15 Tampilan Register .....	46
Gambar 3 16 Tampilan SIMPEG .....	47
Gambar 3 17 Tampilan myProfile Admin .....	47
Gambar 3 18 Tampilan Data Pegawai .....	48
Gambar 3 19 form Tambah Data Pegawai .....	48
Gambar 3 20 Tampilan Detail Data Pegawai .....	49
Gambar 3 21 Form Ubah Data Pegawai .....	49
Gambar 3 22 Tampilan List Riwayat Kerja .....	50
Gambar 3 23 Form Tambah Data Riwayat Kerja .....	50
Gambar 3 24 Form Ubah Riwayat Kerja .....	51

## DAFTAR TABEL

tabel 2 1Jadwal Proyek .....	10
tabel 2 2Pembagian Kerja Tim .....	10
tabel 3 1 data_pribadi .....	43
tabel 3 2 riwayat_kerja .....	43
tabel 3 3 login .....	44
tabel 3 4 user_role .....	45
tabel 3 5 profile_perusahaan .....	45
tabel 3 6 blackbox testing.....	55

## **DAFTAR SIMBOL**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perangkat lunak adalah sekumpulan instruksi atau program yang dirancang untuk menjalankan tugas tertentu pada komputer atau perangkat lainnya. Dalam era digital saat ini, perangkat lunak telah menjadi bagian penting dalam berbagai sektor, termasuk bisnis, pendidikan, kesehatan, dan pemerintahan. Kemajuan teknologi seperti komputasi awan, kecerdasan buatan, dan analitik berbasis data mendorong perkembangan perangkat lunak yang lebih inovatif dan efisien. Dalam dunia kerja, perangkat lunak manajemen sumber daya manusia (HRM) menjadi salah satu solusi yang sangat dibutuhkan untuk mendukung transformasi digital, memungkinkan pengelolaan data karyawan yang lebih terstruktur, cepat, dan aman.

Untuk memastikan pengembangan perangkat lunak yang sesuai dengan kebutuhan pengguna, proyek ini menggunakan metode Agile. Metode ini memungkinkan proses pengembangan dilakukan secara iteratif dan kolaboratif, dengan pembagian pekerjaan ke dalam sprint atau iterasi yang lebih kecil. Agile juga memungkinkan adanya umpan balik yang cepat dari pengguna, sehingga perangkat lunak dapat berkembang secara adaptif sesuai perubahan kebutuhan. Pendekatan ini menjadikan pengembangan perangkat lunak lebih fleksibel dan fokus pada penyampaian nilai secara berkelanjutan.

Aplikasi Data Kepegawaian yang dikembangkan dalam proyek ini bertujuan untuk membantu perusahaan mengelola data karyawan secara efektif dan efisien. Aplikasi ini dirancang untuk mengelola informasi personal dan riwayat kerja karyawan, menyediakan fitur otomatisasi untuk tugas-tugas rutin HR, serta memberikan akses cepat ke data melalui antarmuka yang user-friendly dan berbasis web.

## **1.2 Tujuan Proyek**

Tujuan dari proyek Aplikasi Data Kepegawaian ini adalah untuk mengembangkan sebuah solusi perangkat lunak yang dapat membantu perusahaan dalam mengelola data karyawan secara efisien, akurat, dan terstruktur. Dengan adanya aplikasi ini, perusahaan dapat menyederhanakan proses manajemen sumber daya manusia, sehingga tim HR dapat lebih fokus pada tugas-tugas strategis daripada pekerjaan administratif yang rutin.

Aplikasi ini dirancang untuk memberikan kemudahan dalam mengakses informasi personal, riwayat kerja, dan data lain yang relevan dengan karyawan, melalui antarmuka yang user-friendly dan berbasis web. Selain itu, fitur otomatisasi dalam aplikasi ini bertujuan untuk mengurangi kesalahan manual, mempercepat proses pembaruan data, serta meningkatkan produktivitas tim HR.

Secara keseluruhan, tujuan utama dari proyek ini adalah untuk memberikan nilai tambah bagi perusahaan dengan cara meningkatkan produktivitas, efisiensi, dan efektivitas dalam pengelolaan data karyawan serta mendukung strategi bisnis yang lebih matang.



## METODE

Metode pengumpulan data yang digunakan dalam proyek ini adalah **Studi Dokumentasi**, yang dilakukan pada tahap awal pengembangan perangkat lunak, yaitu pada fase perencanaan dan analisis kebutuhan. Pada fase ini, data dikumpulkan dengan cara menganalisis berbagai dokumen dan informasi yang sudah ada, seperti dokumen kebutuhan sistem, spesifikasi perangkat lunak, serta dokumen terkait lainnya. Fokus utama dari pengumpulan data ini adalah untuk memahami secara mendalam kebutuhan dan spesifikasi sistem yang harus dikembangkan. Informasi yang diperoleh dari studi dokumentasi ini akan menjadi dasar bagi tahap desain dan implementasi sistem, memastikan bahwa pengembangan perangkat lunak berjalan sesuai dengan kebutuhan pengguna dan tujuan proyek.

Menjelaskan jadwal proyek dan disertai dengan pembagian kerja tim. Tabel jadwal proyek disesuaikan dengan jadwal pengerjaan proyek tim.

[illegible]

Implementasi Source Code																
Pembuatan Laporan Bab IV																
Pengujian Blackbox																
Presentasi Akhir																

*tabel 2 .1 Jadwal Proyek*

Pembagian kerja tim dijelaskan menggunakan **tabel** sebagai berikut:

No	Nama	Posisi Tim	Deskripsi Kerja
1	Yohani Natalia.S	Ketua	Mengerjakan bagian UI dan UX (FrontEnd) dan Laporan.
2	Winda Aulia Ariyani	Anggota	Mengerjakan bagian BackEnd (Database/phpMyadmin)) dan Laporan.
3	Kirana Putri Fercia	Anggota	-
4.	Linda Astuti	Anggota	-
5.	Tanra Wija Yanti	Anggota	-

*Tabel 2.2 Pembagian Kerja Tim*

## 2.3 Alur Proyek

Menjelaskan alur atau kerangka pengerjaan proyek perangkat lunak

### 2.3.1 Perencanaan Proyek (*Planning*)

**Deskripsi :** Pada tahap ini, tujuan proyek dan kebutuhan sistem didefinisikan secara jelas, berisi analisis kebutuhan pengguna serta studi kelayakan proyek.

#### **Langkah-Langkah:**

##### **1. Identifikasi Tujuan**

Mengidentifikasi tujuan aplikasi, seperti mengelola data karyawan.

## **2. Studi Kelayakan**

Analisis aspek teknis, ekonomis, dan operasional dari proyek.

## **3. Penyusunan Tim**

Menentukan tim proyek (manajer proyek, pengembang, desainer UI/UX, penguji, dll.).

## **4. Pembuatan Jadwal Proyek**

Membuat timeline dengan tonggak (*milestones*) penting.

**Output :** Dokumen rencana proyek (*Project Charter* atau *Project Plan*).

### **2.3.2 Analisis Kebutuhan (*Requirement Analysis*)**

**Deskripsi :** Ditahap ini berfokus pada pengumpulan kebutuhan pengguna dan bisnis. Kebutuhan tersebut akan diterjemahkan menjadi spesifikasi sistem.

#### **Langkah-Langkah:**

#### **1. Pengumpulan Informasi:**

Wawancara dengan Kepala bagian, sekretaris dan staff untuk memahami proses kepegawaian saat ini.

#### **2. Identifikasi Kebutuhan Fungsional:**

Contoh: Pengelolaan data karyawan,.

#### **3. Identifikasi Kebutuhan Non-Fungsional:**

Contoh: Keamanan data, kecepatan respon sistem, skalabilitas.

#### **4. Dokumentasi Kebutuhan:**

Membuat dokumen *Software Requirement Specification* (SRS).

**Output:** Dokumen SRS yang menjadi acuan pengembangan sistem.

### **2.3.3 Desain Sistem (*System Design*)**

**Deskripsi :** Membuat kerangka teknis dan visual untuk aplikasi, termasuk arsitektur sistem, desain database, dan antarmuka pengguna (*UI/UX*).

#### **Langkah-Langkah:**

#### **1. Desain Arsitektur Sistem:**

Memilih pendekatan teknis (misalnya, arsitektur berbasis web) dan menentukan modul utama, seperti modul data karyawan.

## **2. Desain Database:**

Membuat model ERD (*Entity-Relationship Diagram*) untuk memetakan data seperti karyawan dan departemen

## **3. Desain Antarmuka Pengguna (UI/UX):**

Membuat *prototipe* atau *wireframe* untuk aplikasi dan menentukan elemen UI seperti halaman dashboard dan form.

## **4. Dokumentasi Desain:**

Membuat blueprint teknis untuk tim pengembang.

**Output:** Desain UI/UX (*wireframe* atau *mockup*) dan Diagram arsitektur dan database.

### **2.3.4 Pengembangan Sistem (*Development*)**

**Deskripsi :** Pada tahap ini, pengembang mengimplementasikan aplikasi berdasarkan desain dan spesifikasi yang telah dibuat.

#### **Langkah-Langkah:**

##### **1. Setup Lingkungan Pengembangan**

Menyiapkan tools seperti framework : CodeIgniter, database: Phpmyadmin, dan server: XAMPP.

##### **2. Source Code**

Pengembangan modul aplikasi sesuai dengan spesifikasi, seperti modul manajemen data karyawan.

##### **3. Integrasi Modul**

Mengintegrasikan setiap modul yang dikembangkan agar sistem berjalan secara utuh.

##### **4. Version Control**

Menggunakan sistem version control(misalnya Git) untuk manajemen kode.

**Output :** Aplikasi sistem kepegawaian dalam versi awal (beta).

### **2.3.5 Pengujian Sistem (*Testing*)**

**Deskripsi :** Menguji aplikasi untuk memastikan fungsionalitas, performa, dan keamanan sesuai spesifikasi.

#### **Langkah-Langkah:**

### **1. Pengujian Blackbox**

**Output :** Laporan hasil pengujian, daftar bug, dan perbaikan yang dilakukan.

### **2.3.6 Implementasi dan Pelatihan (*Implementation & Training*)**

**Deskripsi :** Setelah sistem diuji dan disetujui, aplikasi akan diimplementasikan dan pengguna akan diberikan pelatihan.

#### **Langkah-Langkah:**

##### **1. Deployment**

Menginstal aplikasi pada server atau perangkat yang digunakan (cloud/on-premise).

##### **2. Data Migration**

Memasukkan data lama (jika ada) ke dalam sistem baru.

##### **3. Pelatihan Pengguna**

Memberikan pelatihan kepada Kepala bagian, sekretaris, dan staff tentang cara menggunakan website.

**Output :** Aplikasi siap digunakan secara resmi.

### **2.3.7 Pemeliharaan dan Perbaikan (*Maintenance & Support*)**

**Deskripsi :** Tahap ini bertujuan untuk memastikan aplikasi tetap berjalan dengan baik dan sesuai kebutuhan pengguna seiring waktu.

#### **Langkah-Langkah:**

##### **1. Perawatan Preventif:**

Memastikan sistem diperbarui secara berkala (patch, update, dll.).

##### **2. Bug Fixing:**

Memperbaiki kesalahan yang ditemukan setelah implementasi.

**Output :** Sistem yang terus terpelihara dan disesuaikan dengan kebutuhan.

## **BAB III**

### **HASIL DAN PEMBAHASAN**

#### **3.1 Deskripsi Perangkat Lunak**

Deskripsi perangkat lunak yang dikembangkan adalah sebuah sistem manajemen yang dirancang untuk mempermudah pengelolaan data pegawai, riwayat kerja, dan profil perusahaan melalui antarmuka web yang intuitif. Sistem ini dilengkapi dengan fitur login dan register untuk memastikan keamanan dan privasi pengguna, memungkinkan hanya pengguna yang terdaftar dan terotorisasi yang dapat mengakses berbagai fitur dalam aplikasi. Setiap pengguna memiliki akses sesuai dengan peran dan tingkat otoritasnya, dengan tampilan dashboard yang menyajikan informasi penting seperti jumlah departemen, pegawai, status pegawai, dan pengguna sistem secara keseluruhan.

Bagian dashboard memberikan gambaran umum yang memudahkan pengguna untuk memonitor status data yang ada dalam sistem. Pada bagian data pegawai, terdapat fitur CRUD (Create, Read, Update, Delete) yang memungkinkan admin untuk menambah, melihat, mengedit, dan menghapus data pegawai dengan mudah. Selain itu, fitur searching bar juga disediakan untuk mempermudah pencarian pegawai berdasarkan kriteria tertentu. Begitu pula dengan bagian data riwayat kerja, yang dilengkapi dengan fitur CRUD dan searching bar untuk mengelola informasi riwayat pekerjaan pegawai secara efisien.

Sistem ini juga memiliki bagian profil perusahaan yang menyajikan informasi penting tentang perusahaan, seperti nama perusahaan, alamat, nomor telepon, dan email. Semua data ini dapat diakses dan diperbarui oleh admin sesuai kebutuhan, untuk memastikan informasi yang ditampilkan selalu up-to-date dan akurat. Dengan fitur-fitur ini, perangkat lunak ini membantu mempermudah pengelolaan data internal perusahaan serta meningkatkan efisiensi operasional dan komunikasi antar pengguna dalam sistem.

### 3.2 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan dalam pengerjaan proyek ini adalah **Metode Waterfall** yang bersifat linear dan berurutan, di mana setiap fase pengembangan perangkat lunak dilakukan secara terstruktur dan dimulai setelah tahap sebelumnya selesai. Metode ini sangat cocok untuk proyek dengan kebutuhan yang sudah jelas dan tidak banyak berubah selama pengembangan. Proses dimulai dengan **perencanaan** dan **analisis kebutuhan**, yang kemudian dilanjutkan ke fase **desain** untuk merancang struktur sistem, termasuk antarmuka pengguna dan database. Setelah desain disetujui, proyek dilanjutkan ke fase **implementasi**, di mana pengkodean perangkat lunak dilakukan sesuai dengan spesifikasi yang telah direncanakan.

Setelah fase implementasi, proyek memasuki tahap **pengujian** (testing), di mana perangkat lunak diuji untuk memastikan bahwa semua fitur berjalan dengan baik dan sesuai dengan kebutuhan pengguna. Pengujian dilakukan secara menyeluruh, baik untuk fungsionalitas maupun untuk mencari potensi bug atau kesalahan yang mungkin terjadi. Setelah perangkat lunak diuji dan diperbaiki, proyek masuk ke tahap **deploy** atau penerapan, di mana sistem siap digunakan oleh pengguna akhir. Pada tahap ini, perangkat lunak akan dipasang pada server atau sistem produksi yang telah disiapkan.

Meskipun metode Waterfall bersifat terstruktur, fleksibilitas dalam pengerjaan proyek ini tetap dijaga, terutama untuk penyesuaian berdasarkan feedback dari pengguna atau perubahan kebutuhan yang terjadi selama pengembangan. Untuk memfasilitasi perubahan yang diperlukan, dilakukan evaluasi secara rutin setelah setiap tahap selesai, dan pengembangan dilakukan dengan pendekatan iteratif yang memungkinkan perbaikan berkelanjutan pada setiap fase. Pendekatan ini memastikan bahwa perangkat lunak yang dikembangkan memenuhi kebutuhan pengguna dengan kualitas yang tinggi dan dapat diandalkan.

### 3.3 Analisis Kebutuhan

#### 3.3.1 Kebutuhan Fungsional

##### User Class 1 – Admin

- Kebutuhan Fungsional 1.1

ID: KF5

TITLE: Log in

DESC: Admin dapat masuk ke dalam sistem menggunakan kredensial untuk mendapatkan akses penuh ke fitur-fitur aplikasi.

RAT: Agar Admin HR dapat mengelola data kepegawaian dalam sistem.

DEP: Tidak ada

- Kebutuhan Fungsional 1.2

ID: KF6

TITLE: Mengelola Data Pegawai

DESC: Admin dapat menambah, mengubah, dan menghapus data pegawai yang mencakup informasi personal dan jabatan.

RAT: Agar Admin HR dapat memastikan data pegawai selalu terbaru dan akurat.

DEP: KF6

- Kebutuhan Fungsional 1.3

ID: KF7

TITLE: Mengelola Riwayat Kerja

DESC: Admin bertanggung jawab mengelola riwayat kerja pegawai, termasuk pencatatan perubahan jabatan.



RAT: Agar informasi karier pegawai dalam sistem tetap teratur dan terperinci.

DEP: KF6

- Kebutuhan Fungsional 1.4

ID: KF9

TITLE: Log Out

DESC: Setelah selesai menggunakan aplikasi, Admin HR dapat keluar dari sistem dengan aman melalui fitur Log Out.

RAT: Agar keamanan data dan akses sistem tetap terjaga.

DEP: KF6

### 3.3.2 Kebutuhan Non-Fungsional

#### 1.Keamanan (Security):

- **Autentikasi dan Otorisasi:** Sistem harus menyediakan mekanisme login yang aman untuk Pegawai dan Admin. Fitur **Login** (KF1, KF5) harus memastikan hanya pengguna yang terautentikasi dengan benar (username dan password yang valid) yang dapat mengakses sistem. Password harus dienkripsi untuk melindungi data pribadi pengguna dan mencegah akses yang tidak sah. Sistem juga harus membatasi akses berdasarkan peran pengguna (Pegawai atau Admin) untuk memastikan bahwa setiap pengguna hanya dapat mengakses fitur yang sesuai dengan hak akses mereka.
- **Log Out** (KF5, KF9) juga harus mengamankan sesi pengguna dengan memastikan bahwa saat pengguna keluar, sesi mereka benar-benar dihentikan, mencegah akses tidak sah ke data pribadi setelah keluar dari aplikasi.

#### 2.Kinerja (Performance):

- **Responsivitas dan Kecepatan Akses:** Semua fitur dalam sistem, seperti **melihat riwayat kerja** (KF2), **melihat data pribadi** (KF3), **melihat profil perusahaan** (KF4), dan **mengelola data pegawai** (KF6), harus diakses dalam waktu singkat tanpa penundaan yang berarti. Pengguna, baik Pegawai maupun Admin, diharapkan dapat mengakses data secara cepat dan efisien tanpa adanya lag yang mengganggu kenyamanan penggunaan.
- **Scalability:** Sistem harus dapat menangani pertumbuhan data yang semakin besar. Misalnya, seiring bertambahnya jumlah pegawai dan riwayat kerja, sistem harus mampu menambah dan mengelola data pegawai dan riwayat kerja dengan tetap mempertahankan performa yang baik. Fitur seperti **mengelola data pegawai** (KF6) dan **mengelola riwayat kerja** (KF7) harus dapat berfungsi optimal meskipun data yang dikelola bertambah.

#### 3.Ketersediaan (Availability):

- Sistem harus selalu tersedia untuk Pegawai dan Admin, terutama saat mereka membutuhkan akses ke informasi pribadi atau riwayat kerja mereka. **Keandalan dan uptime** sistem sangat penting, terutama untuk fitur yang berkaitan dengan data penting

seperti **riwayat kerja** dan **data pegawai**. Keterlambatan dalam pengelolaan data atau downtime sistem dapat mengganggu operasional dan pengalaman pengguna.

#### **4. Usability:**

- **Antarmuka Pengguna:** Sistem harus memiliki antarmuka pengguna yang sederhana dan mudah digunakan oleh Pegawai dan Admin. Pengguna harus dapat dengan mudah mengakses fitur seperti **melihat data pribadi** (KF3), **melihat riwayat kerja** (KF2), dan **mengelola data pegawai** (KF6) tanpa kesulitan. Desain antarmuka harus intuitif dan user-friendly, memastikan pengalaman pengguna yang lancar dan tidak membingungkan. Selain itu, sistem harus mendukung aksesibilitas yang baik untuk berbagai jenis perangkat, baik desktop maupun mobile, untuk memudahkan pengguna dalam mengakses informasi kapan saja dan di mana saja.

#### **5. Kompatibilitas (Compatibility):**

- Sistem harus kompatibel dengan berbagai perangkat dan browser yang umum digunakan oleh pengguna. Misalnya, jika pengguna mengakses sistem melalui browser atau perangkat mobile, fitur seperti **profil perusahaan** (KF4), **melihat riwayat kerja** (KF2), dan **data pegawai** (KF3) harus ditampilkan dengan jelas dan mudah dibaca. Selain itu, sistem harus mendukung standar web modern untuk memastikan tidak ada masalah tampilan atau fungsionalitas di berbagai platform.

#### **6. Integritas Data (Data Integrity):**

- **Keakuratan dan Konsistensi:** Data yang dimasukkan ke dalam sistem, seperti data pegawai dan riwayat kerja, harus dijaga keakuratannya. Fitur seperti **mengelola data pegawai** (KF6) dan **mengelola riwayat kerja** (KF7) harus memastikan bahwa data yang diubah atau diperbarui tidak menyebabkan ketidakkonsistenan dalam sistem. Sistem harus memiliki mekanisme untuk menghindari duplikasi atau kesalahan dalam pengelolaan data yang penting.

## **7. Pemeliharaan dan Pembaruan (Maintainability and Updatability):**

- Sistem harus dirancang sedemikian rupa sehingga dapat diperbarui atau dipelihara dengan mudah. Pembaruan pada sistem, baik untuk memperbaiki bug atau menambah fitur baru, harus dilakukan tanpa mengganggu kinerja dan stabilitas sistem. Sebagai contoh, fitur **mengelola riwayat kerja (KF7)** dan **data pegawai (KF6)** harus memungkinkan admin untuk dengan mudah menambah, mengubah, atau menghapus data tanpa menimbulkan masalah pada data yang ada sebelumnya.

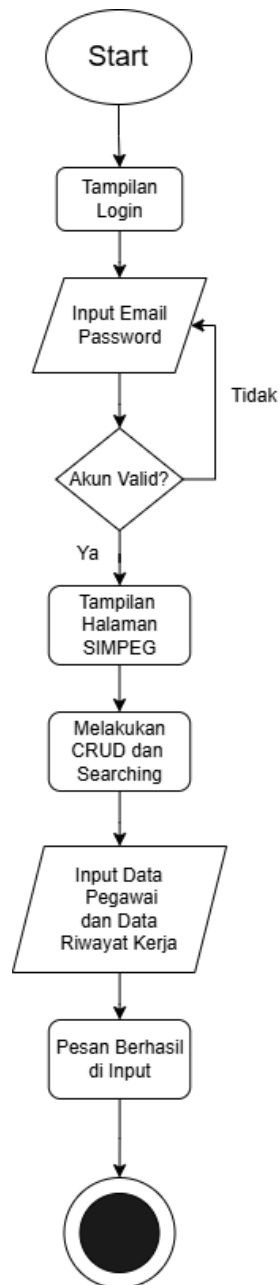
### 3.4 Perancangan Perangkat Lunak

Menjelaskan perancangan perangkat lunak yang meliputi tahapan:

#### 3.4.1 Perancangan Sistem

##### 3.4.1.1 Flowchart

###### a. Admin



*Gambar 3 1 flowchart admin*

Berikut adalah penjelasan per langkah dalam flowchart diatas:

1. **Start**

Proses dimulai dari langkah "Start".

2. **Tampilan Login**

Sistem menampilkan halaman login untuk admin.

3. **Input Email dan Password**

Admin memasukkan email dan password ke dalam form login.

4. **Akun Valid?**

- **Decision Point:** Sistem memverifikasi kredensial (email dan password) admin.
  - Jika **tidak valid**, proses akan kembali ke langkah **Input Email dan Password**.
  - Jika **valid**, proses berlanjut ke langkah berikutnya.

5. **Tampilan Halaman SIMPEG**

Setelah login berhasil, sistem menampilkan halaman utama SIMPEG untuk admin.

6. **Melakukan CRUD dan Searching**

Admin dapat melakukan operasi **CRUD (Create, Read, Update, Delete)** dan **Searching** pada data pegawai.

7. **Input Data Pegawai dan Data Riwayat Kerja**

Admin melakukan input data pegawai dan riwayat kerja pegawai ke dalam sistem.

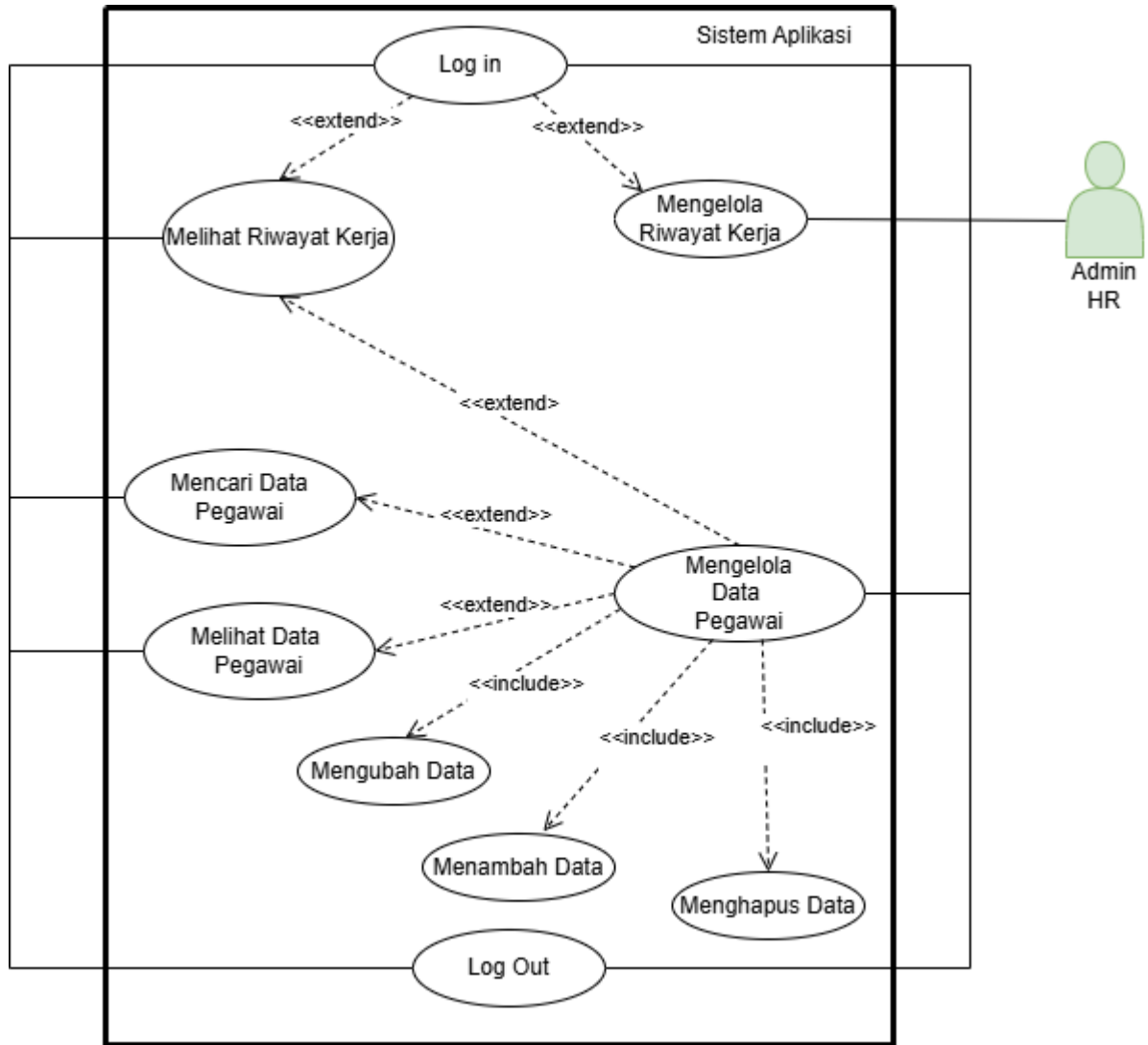
8. **Pesan Berhasil di Input**

Sistem memberikan pesan notifikasi bahwa data berhasil diinput.

9. **End**

Proses selesai.

### 3.4.1.2 Use Case Diagram



Gambar 3 2 Use Case Diagram

Sistem aplikasi ini mencakup berbagai fitur atau **use case** yang dapat digunakan oleh Admin HR. Berikut penjelasan masing-masing use case:

#### 1. **Log in**

- Admin HR harus melakukan login terlebih dahulu untuk mengakses fitur dalam sistem.

- Use case ini dihubungkan dengan fitur lain melalui hubungan <<extend>>, artinya login adalah langkah tambahan yang diperlukan sebelum mengelola data.

## 2. Mengelola Data Pegawai

- Fitur utama yang memungkinkan Admin HR untuk melakukan berbagai operasi pada data pegawai.
- Use case ini mencakup (<<include>>) fitur berikut:
  - **Menambah Data:** Menambahkan data pegawai baru.
  - **Mengubah Data:** Memperbarui informasi pegawai yang sudah ada.
  - **Menghapus Data:** Menghapus data pegawai tertentu.
- Selain itu, fitur ini diperluas (<<extend>>) ke use case lain seperti:
  - **Melihat Data Pegawai:** Untuk melihat informasi pegawai tertentu.
  - **Mencari Data Pegawai:** Untuk mencari data pegawai berdasarkan kriteria tertentu.

## 3. Mengelola Riwayat Kerja

- Fitur ini memungkinkan Admin HR untuk mengelola informasi riwayat kerja pegawai.
- Fitur ini diperluas (<<extend>>) dari fitur melihat riwayat kerja.

## 4. Melihat Riwayat Kerja

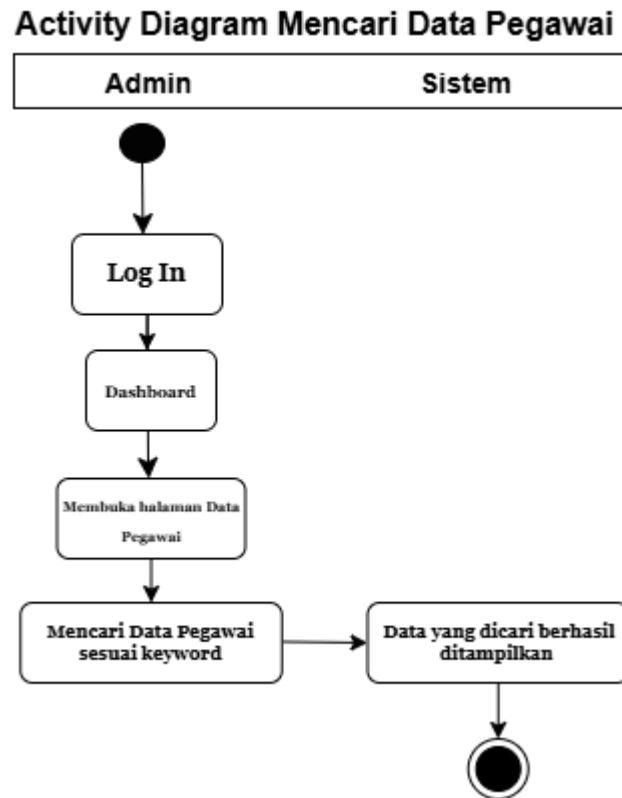
- Admin HR dapat melihat riwayat kerja pegawai tertentu.
- Use case ini saling terkait dengan Mengelola Riwayat Kerja.

## 5. Log Out

- Setelah selesai menggunakan sistem, Admin HR dapat keluar dari aplikasi menggunakan fitur log out.



### 3.4.1.3 Activity Diagram



Gambar 3 3 Activity Diagram mencari

- **Admin Melakukan Login:**

Aktivitas dimulai dengan admin melakukan proses login ke dalam sistem.

Sistem memverifikasi kredensial admin.

- **Dashboard:**

Setelah berhasil login, admin diarahkan ke halaman dashboard utama.

- **Membuka Halaman Data Pegawai:**

Dari dashboard, admin memilih untuk membuka halaman yang berisi daftar data pegawai.

- **Mencari Data Pegawai Sesuai Keyword:**

Admin memasukkan kata kunci (keyword) tertentu untuk mencari data pegawai di sistem.

Contoh keyword: nama, ID pegawai, atau departemen.

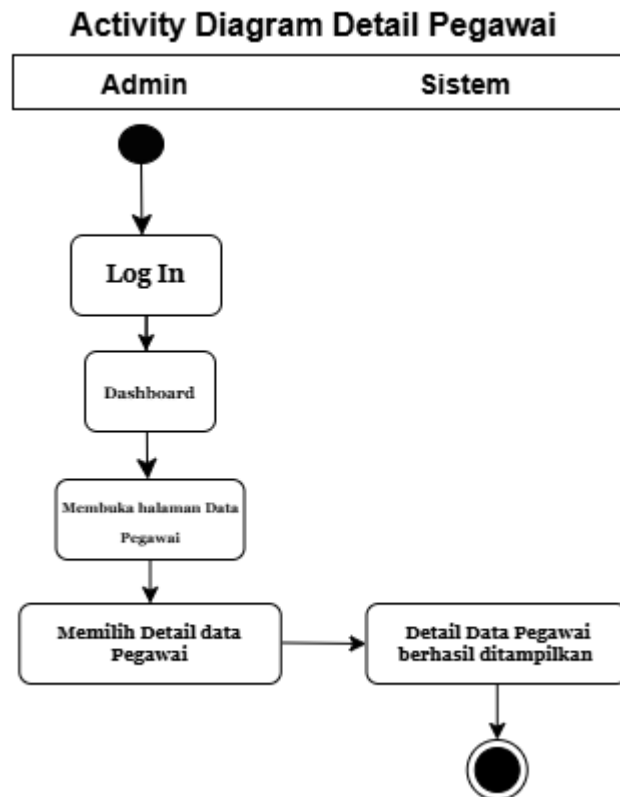
- **Sistem Menampilkan Data yang Dicari:**

Sistem memproses permintaan pencarian dan menampilkan data pegawai yang sesuai dengan keyword yang dimasukkan.

Jika data ditemukan, hasilnya akan muncul di layar admin.

- **Proses Selesai:**

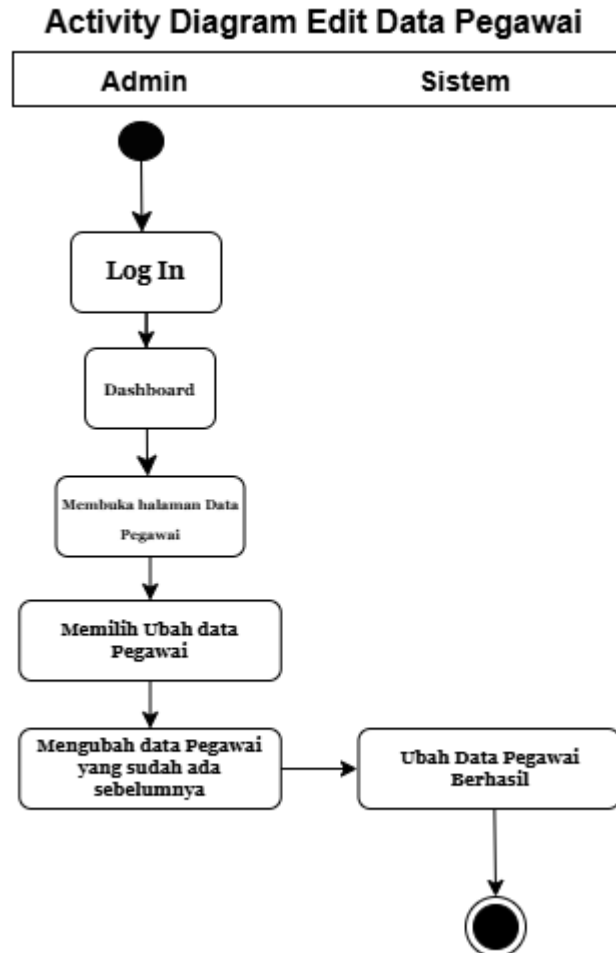
Aktivitas berakhir setelah data pegawai berhasil ditampilkan.



*Gambar 3 4 Activity Diagram Detail*

- **Mulai Proses (Lingkaran Hitam)**
  - Proses dimulai ketika Admin memulai interaksinya dengan sistem.
- **Admin Log In**
  - Admin melakukan login ke dalam sistem menggunakan kredensialnya.
- **Masuk ke Dashboard**
  - Setelah login berhasil, Admin diarahkan ke halaman Dashboard, tempat berbagai fitur sistem tersedia.
- **Membuka Halaman Data Pegawai**

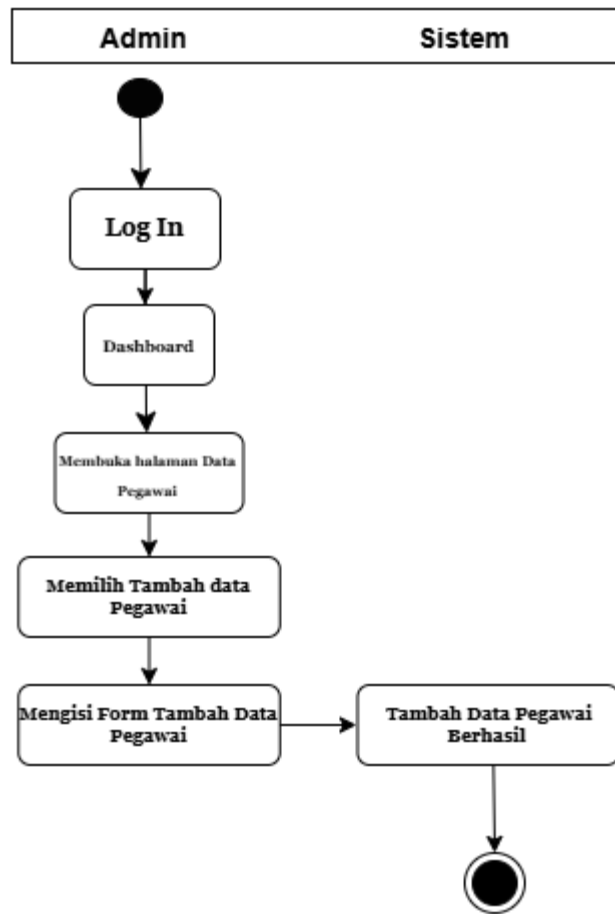
- Dari Dashboard, Admin memilih menu atau opsi untuk membuka halaman Data Pegawai.
- **Memilih Detail Data Pegawai**
  - Admin memilih salah satu pegawai untuk melihat detail informasinya.
- **Sistem Menampilkan Detail Data Pegawai**
  - Sistem memproses permintaan Admin dan menampilkan informasi detail pegawai yang dipilih.
- **Proses Berakhir (Lingkaran Hitam dengan Lingkaran Putih di Tengah)**
  - Proses selesai setelah data pegawai berhasil ditampilkan.



*Gambar 3 5 Activity Diagram Edit*

- **Admin Melakukan Login:**
  - Proses dimulai dengan admin melakukan login ke dalam sistem untuk mendapatkan akses ke fitur yang ada.
  - Sistem memvalidasi kredensial admin.
- **Dashboard:**
  - Setelah berhasil login, admin diarahkan ke halaman utama atau dashboard.
- **Membuka Halaman Data Pegawai:**
  - Dari dashboard, admin memilih menu untuk membuka halaman yang berisi daftar data pegawai.
- **Memilih Ubah Data Pegawai:**
  - Admin memilih fitur atau opsi untuk mengubah data pegawai tertentu.
- **Mengubah Data Pegawai yang Sudah Ada Sebelumnya:**
  - Admin memasukkan perubahan yang diinginkan pada data pegawai yang sudah ada, seperti memperbarui nama, jabatan, atau departemen.
  - Input perubahan ini kemudian dikirimkan ke sistem.
- **Sistem Memproses dan Mengubah Data:**
  - Sistem memvalidasi data baru yang dimasukkan admin, kemudian memperbarui data pegawai dalam database.
- **Ubah Data Pegawai Berhasil:**
  - Sistem memberikan konfirmasi bahwa perubahan data pegawai berhasil dilakukan.
  - Proses selesai.

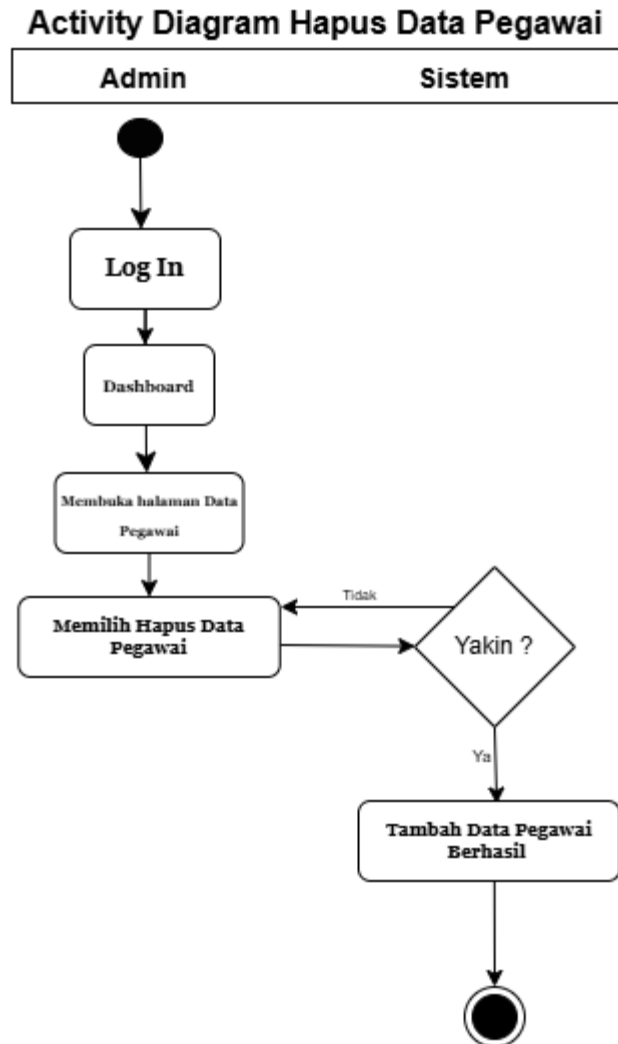
### Activity Diagram Tambah Data Pegawai



*Gambar 3 6 Activity Diagram Tambah*

- **Mulai Proses (Lingkaran Hitam)**
  - Proses dimulai saat Admin memulai interaksinya dengan sistem.
- **Admin Log In**
  - Admin masuk ke dalam sistem menggunakan kredensialnya (username dan password).
- **Masuk ke Dashboard**
  - Setelah login berhasil, Admin diarahkan ke halaman Dashboard, tempat berbagai fitur sistem tersedia.
- **Membuka Halaman Data Pegawai**
  - Dari Dashboard, Admin memilih menu atau opsi untuk membuka halaman Data Pegawai.
- **Memilih Tambah Data Pegawai**

- Admin memilih fitur untuk menambahkan data pegawai baru.
- **Mengisi Form Tambah Data Pegawai**
  - Admin memasukkan informasi yang diperlukan (seperti nama pegawai, posisi, kontak, dll.) ke dalam formulir yang disediakan oleh sistem.
- **Sistem Memproses Tambah Data Pegawai**
  - Setelah formulir diisi, sistem memproses data yang dimasukkan Admin dan menyimpan data pegawai baru ke dalam database.
- **Data Pegawai Berhasil Ditambahkan**
  - Sistem memberikan notifikasi bahwa data pegawai berhasil ditambahkan.
- **Proses Berakhir (Lingkaran Hitam dengan Lingkaran Putih di Tengah)**
  - Proses selesai setelah data pegawai baru berhasil disimpan ke dalam sistem.



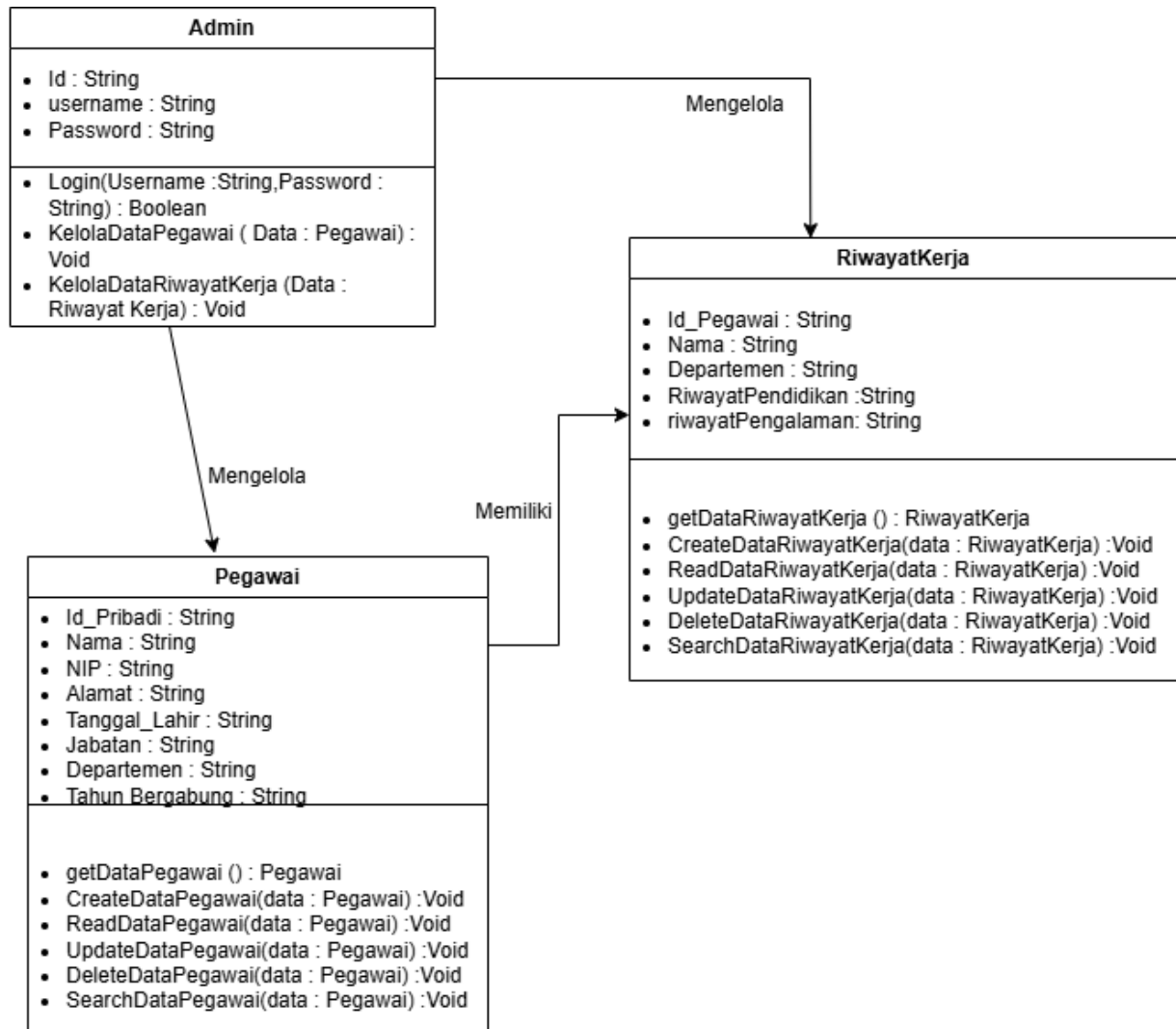
*Gambar 3 7 Activity Diagram Hapus*

- **Mulai Proses (Lingkaran Hitam)**
  - Proses dimulai ketika Admin memulai interaksinya dengan sistem.
- **Admin Log In**
  - Admin masuk ke sistem dengan menggunakan kredensial yang valid (username dan password).
- **Masuk ke Dashboard**
  - Setelah berhasil login, Admin diarahkan ke halaman Dashboard.
- **Membuka Halaman Data Pegawai**
  - Admin memilih menu atau opsi untuk membuka halaman Data Pegawai.
- **Memilih Hapus Data Pegawai**

- Admin memilih pegawai tertentu dari daftar yang ingin dihapus.
- **Konfirmasi Penghapusan (Sistem)**
  - Sistem menampilkan pesan konfirmasi seperti: "**Yakin?**", untuk memastikan Admin benar-benar ingin menghapus data pegawai tersebut.
  - **Jika Admin memilih "Tidak":**
    - Proses kembali ke langkah **Memilih Hapus Data Pegawai**, dan data pegawai tidak dihapus.
  - **Jika Admin memilih "Ya":**
    - Sistem memproses permintaan penghapusan data pegawai.
- **Data Pegawai Berhasil Dihapus**
  - Sistem memberikan notifikasi bahwa data pegawai telah berhasil dihapus.
- **Proses Berakhir (Lingkaran Hitam dengan Lingkaran Putih di Tengah)**
  - Proses selesai setelah data pegawai dihapus dari sistem.



### 3.4.1.5 Class Diagram



Gambar 3 8 Class Diagram

#### Kelas Admin

- **Atribut:**
  - Id: Identitas unik admin.
  - username: Nama pengguna untuk login.
  - Password: Kata sandi untuk login.

- **Metode/Fungsi:**

- `Login(username: String, Password: String): Boolean:` Memvalidasi login admin dengan mengembalikan nilai boolean (true jika berhasil).
- `KelolaDataPegawai(data: Pegawai): Void:` Mengelola data pegawai (mencakup CRUD).
- `KelolaDataRiwayatKerja(data: RiwayatKerja): Void:` Mengelola data riwayat kerja pegawai.

Admin bertanggung jawab untuk mengelola seluruh data pegawai dan riwayat kerja, termasuk validasi akses melalui login.

---

## Kelas Pegawai

- **Atribut:**

- `Id_Pribadi:` Identitas unik untuk setiap pegawai.
- `Nama:` Nama pegawai.
- `NIP:` Nomor Induk Pegawai.
- `Alamat:` Alamat pegawai.
- `Tanggal_Lahir:` Tanggal lahir pegawai.
- `Jabatan:` Posisi atau jabatan pegawai.
- `Departemen:` Departemen tempat pegawai bekerja.
- `Tahun_Bergabung:` Tahun pegawai mulai bergabung.

- **Metode/Fungsi:**

- `getDataPegawai(): Pegawai:` Mengambil data pegawai.
- `CreateDataPegawai(data: Pegawai): Void:` Menambahkan data pegawai baru.
- `ReadDataPegawai(data: Pegawai): Void:` Membaca data pegawai tertentu.

- `UpdateDataPegawai(data: Pegawai): Void`: Memperbarui data pegawai.
- `DeleteDataPegawai(data: Pegawai): Void`: Menghapus data pegawai.
- `SearchDataPegawai(data: Pegawai): Void`: Mencari data pegawai berdasarkan kriteria tertentu.

Kelas ini mendeskripsikan entitas pegawai dan operasinya, termasuk fitur CRUD (Create, Read, Update, Delete) data pegawai.

---

## Kelas RiwayatKerja

- **Atribut:**

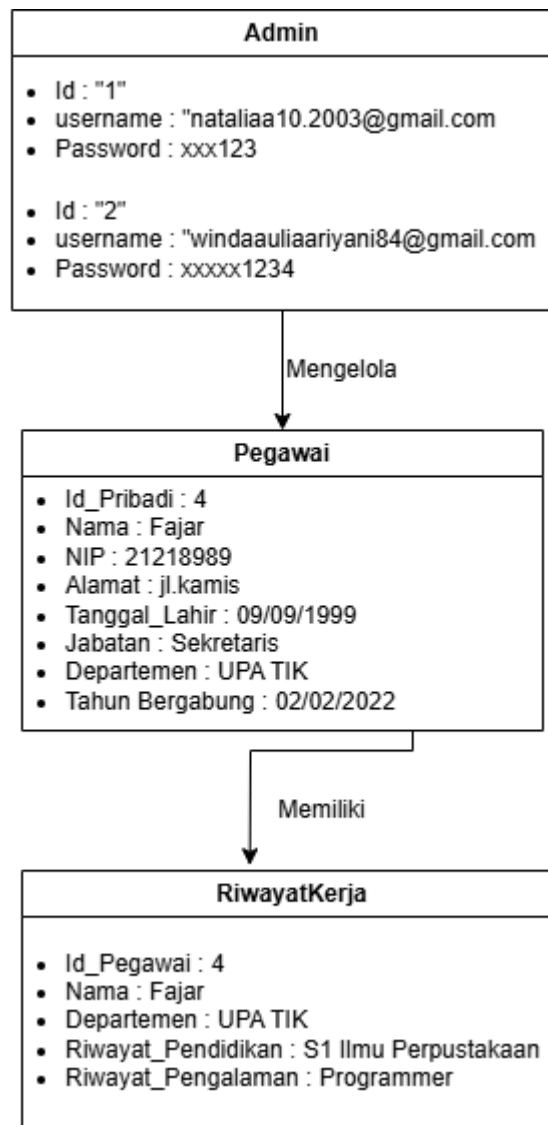
- `Id_Pegawai`: Identitas pegawai yang memiliki riwayat kerja ini.
- `Nama`: Nama pegawai terkait.
- `Departemen`: Departemen tempat pegawai bekerja di suatu waktu tertentu.
- `RiwayatPendidikan`: Riwayat pendidikan pegawai.
- `riwayatPengalaman`: Pengalaman kerja pegawai.

- **Metode/Fungsi:**

- `getDataRiwayatKerja(): RiwayatKerja`: Mengambil data riwayat kerja pegawai.
- `CreateDataRiwayatKerja(data: RiwayatKerja): Void`: Menambahkan riwayat kerja baru.
- `ReadDataRiwayatKerja(data: RiwayatKerja): Void`: Membaca data riwayat kerja tertentu.
- `UpdateDataRiwayatKerja(data: RiwayatKerja): Void`: Memperbarui data riwayat kerja.
- `DeleteDataRiwayatKerja(data: RiwayatKerja): Void`: Menghapus data riwayat kerja.

- SearchDataRiwayatKerja(data: RiwayatKerja): Void: Mencari data riwayat kerja berdasarkan kriteria tertentu.

### 3.4.1.6 Object Diagram



Gambar 3 9 Object Diagram

Berikut adalah penjelasan alur object diagram:

#### 1. Entitas admin

- **Atribut:**

- `id = "1"`
- `username = "nataliaa10.2003@gmail.com"`
- `password = "xxx123"`
  
- `id = "2"`
- `username = "windaauliaariyani84@gmail.com"`
- `password = "xxxxx1234"`

- **Peran:**

Admin berperan dalam **mengelola** data pegawai.

## 2. Entitas pegawai1

- **Atribut:**

- `idPegawai = "4"`
- `nama = "Fajar"`
- `nip = "21218989"`
- `alamat = "jl.kamis "`
- `tanggallahir = "1999-09-09"`
- `jabatan = "Sekretaris"`
- `departemen = "UPA TIK"`
- `tahunBergabung = "2022-02-02"`

- **Hubungan:**

- **Dikelola oleh admin** → Admin memiliki akses untuk menambah, memperbarui, atau menghapus data pegawai.
- **Memiliki** → pegawai1 memiliki **riwayatkerja** tertentu.

## 3. Entitas riwayatkerja1

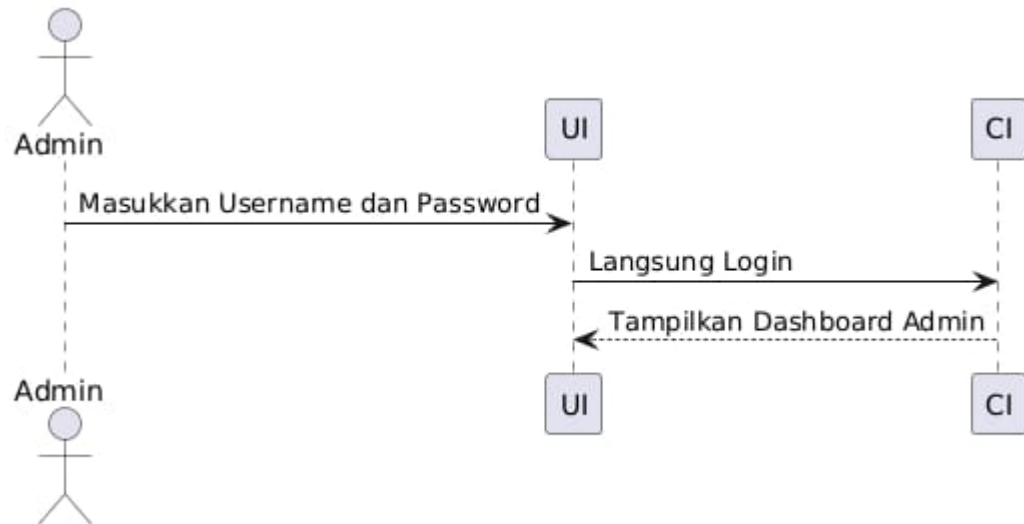
- **Atribut:**
  - `idPegawai` = "4" → Referensi ke pegawai (`pegawai1`).
  - `Nama` = "Fajar"
  - `departemen` = "UPA TIK"
  - `riwayatPendidikan` = "S1 Ilmu Perpustakaan"
  - `riwayatPengalaman` = "Programmer"
- **Hubungan:**
  - Berelasi dengan **pegawai1** melalui `idPegawai`.  
Ini berarti setiap pegawai memiliki riwayat yang mencakup pendidikan dan pengalaman.

#### 4. Alur Relasi Antar Entitas:

- **Admin (`admin1`)** mengelola data pegawai dalam sistem.
  - Ini menunjukkan bahwa admin bertanggung jawab atas manajemen data pegawai.
- **Pegawai (`pegawai1`)** berisi informasi personal seperti nama, NIP, alamat, jabatan, departemen, dan tahun bergabung.
  - Data pegawai merupakan objek utama yang dikelola admin.
- **RiwayatKerja (`riwayatkerja1`)** merupakan bagian dari data pegawai yang berisi informasi riwayat pendidikan dan pengalaman.
  - Hubungan ini ditunjukkan dengan atribut `idPegawai` yang mengaitkan `riwayatkerja1` dengan `pegawai1`.

### 3.4.1.6 Sequence Diagram

#### a. Proses Login (Admin)

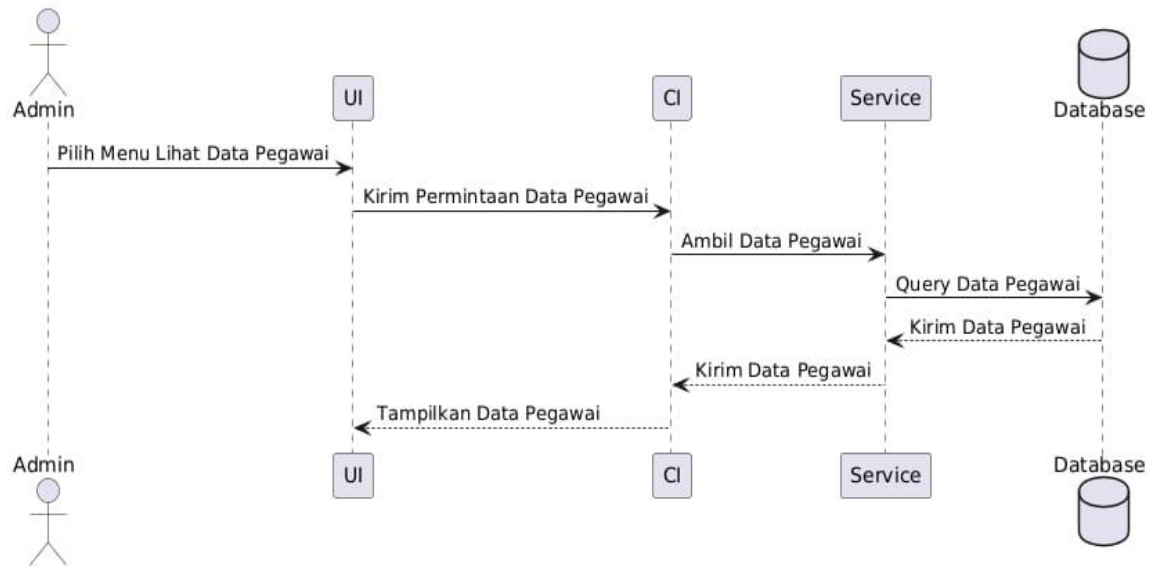


*Gambar 3 10 Sequence Diagram login*

Berikut adalah penjelasan alur proses diagram sequence:

1. Admin langsung login ke sistem tanpa verifikasi tambahan.
2. Tidak ada interaksi dengan Service atau Database karena tidak diperlukan validasi kredensial.

b. Melihat Data Pegawai ( Admin)

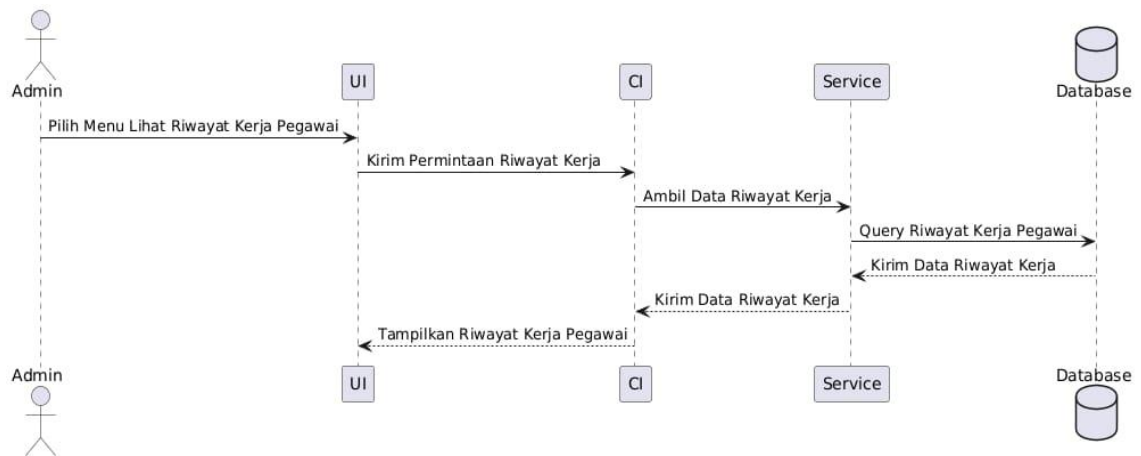


*Gambar 3 11 Sequence Diagram melihat data pegawai*

Berikut adalah penjelasan alur dari diagram sequence:

1. Admin mengakses menu untuk melihat data pegawai.
2. CI memproses permintaan melalui Service dan mengambil data dari Database.
3. Setelah data pegawai diambil, sistem menampilkan informasi di antarmuka Admin.

c. Melihat Riwayat Kerja Pegawai (Admin)



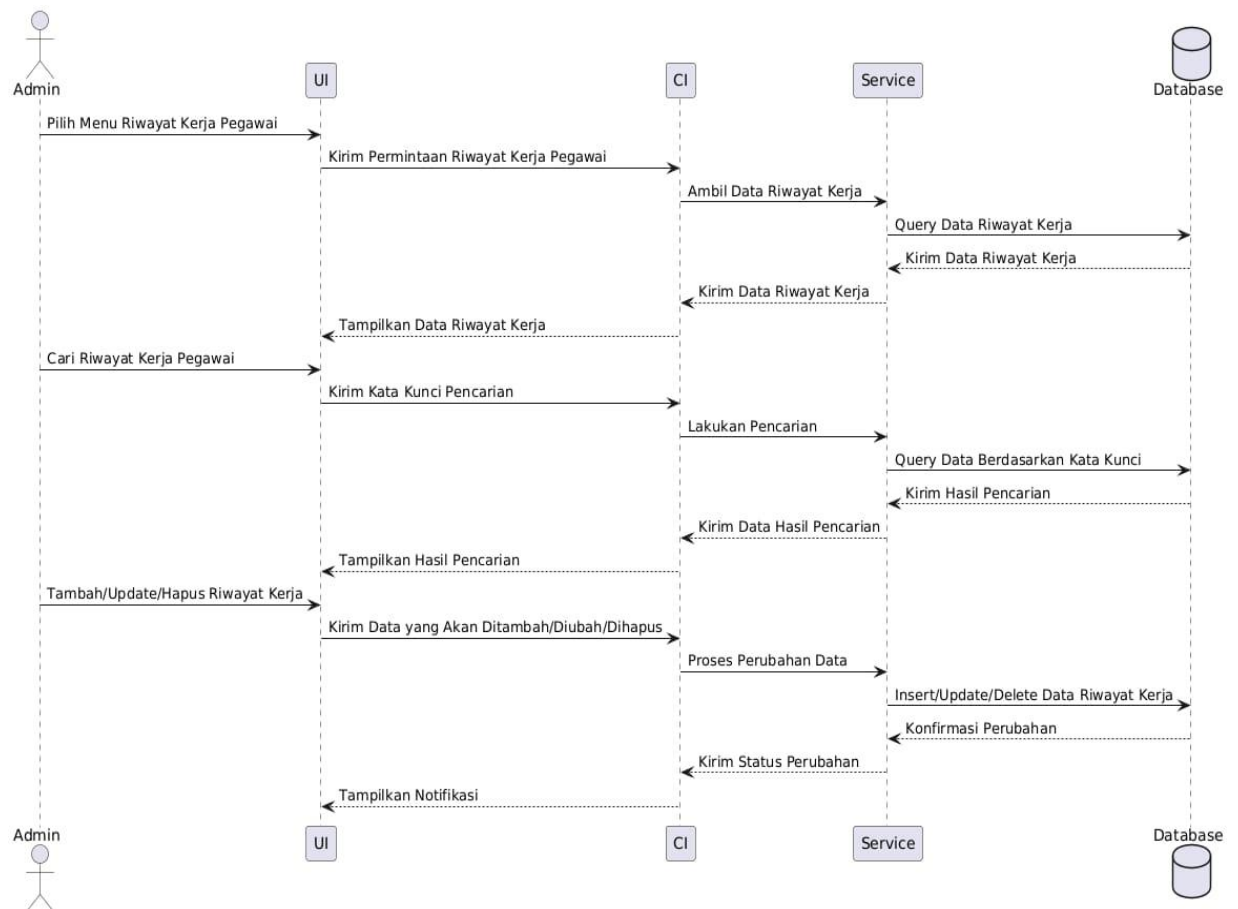
*Gambar 3 12 Sequence Diagram melihat riwayat kerja*



Berikut adalah penjelasan alur proses dari diagram sequence:

1. Admin memilih menu Lihat Riwayat Kerja Pegawai melalui antarmuka UI.
2. UI mengirimkan permintaan ke CI (Controller).
3. CI meminta Service untuk mengambil data riwayat kerja dari database.
4. Service berkomunikasi dengan Database untuk melakukan query data.
5. Database mengembalikan data riwayat kerja ke Service.
6. Service mengirimkan data ke CI, yang kemudian mengirimkannya ke UI untuk ditampilkan ke Admin.

d. Mengelola Data Pegawai (Admin)



*Gambar 3 13 Sequence Diagram mengelola data pegawai*

Berikut adalah penjelasan alur dari diagram sequence:

1. (Melihat Data Pegawai): Admin meminta data pegawai, sistem mengambil data dari database, dan menampilkan data di UI.
2. Search (Pencarian): Admin memasukkan kata kunci pencarian, sistem memproses query pencarian, dan hasilnya ditampilkan di UI.
3. CRUD (Tambah, Update, Hapus): Admin dapat menambah, memperbarui, atau menghapus data pegawai. Sistem memproses perubahan dan menyimpan hasilnya ke database

#### 3.4.1.7 Perancangan Database

Tabel 1 <data\_pribadi>

Identifikasi>Nama : tbdata\_pribadi

Deskripsi isi : Menyimpan data pribadi yang didalam ada nama, NIP, alamat, tanggal lahir,jabatan, departemen dan tahun bergabung

Jenis : table data induk

Primary key : id\_pribadi

Id Field	Tipe & length	NULL	Deskripsi
id_pribadi	Primary Key-INT(5)	No	Identifikasi unik untuk setiap pegawai
nama	Varchar(100)	No	Nama Lengkap pegawai
nip	Varchar(100)	No	Nomor Induk Pegawai
alamat	Varchar(100)	Yes	Alamat Pegawai
tanggal_lahir	Date	Yes	Tanggal lahir pegawai
jabatan	Varchar(100)	Yes	Jabatan pegawai

departemen	Varchar(128)	No	Departemen Pegawai
tahun_bergabung	date	Yes	Tanggal Bergabung pegawai

*tabel 3 1 data\_pribadi*

Tabel 2 <riwayat\_kerja>

Identifikasi>Nama : tbriwayat\_kerja

Deskripsi isi : Menyimpan data riwayat kerja pegawai yang mencakup informasi nama,departemen,riwayat pendidikan dan riwayat pengalaman.

Jenis : table data induk

Primary key : id\_pegawai

Id Field	Tipe & length	NULL	Deskripsi
id_pegawai	Primary Key, INT(9)	No	Identifikasi unik untuk setiap data riwayat kerja
nama	Varchar(255)	No	Identifikasi nama pegawai
departemen	Varchar(255)	No	Identifikasi departemen pegawai
riwayat_pendidikan	Varchar(255)	No	Jam saat pegawai memasuki tempat kerja.
riwayat_pengalaman	Varchar(255)	No	Jam saat pegawai meninggalkan tempat kerja.

*tabel 3 2 riwayat\_kerja*

Tabel 3 <login>

Identifikasi>Nama : tblogin

Deskripsi isi : Menyimpan informasi login

Jenis : table data induk

Primary key : id

Id Field	Tipe & length	NULL	Deskripsi
id	Primarykey & INT(11)	No	Identifikasi unik untuk setiap id login
name	Varchar(128)	No	Identifikasi nama login
email	Varchar(128)	No	Email data user dan admin
image	Varchar(128)	No	Gambar
password	Varchar(128)	No	Identifikasi unik password
role_id	INT(11)	No	
is_active	INT(1)	No	
date_created	INT(11)	No	

*tabel 3 3 login*

Tabel 4 <user\_role>

Identifikasi>Nama : tbuser\_role  
 Deskripsi isi : Menyimpan data role  
 Jenis : table data transaksi  
 Primary key : id

Id Field	Tipe & length	NULL	Deskripsi
id	INT(11)	No	Identifikasi unik untuk setiap data user role
role	Varchar(128)	No	Role pekerjaan

*tabel 3 4 user\_role*

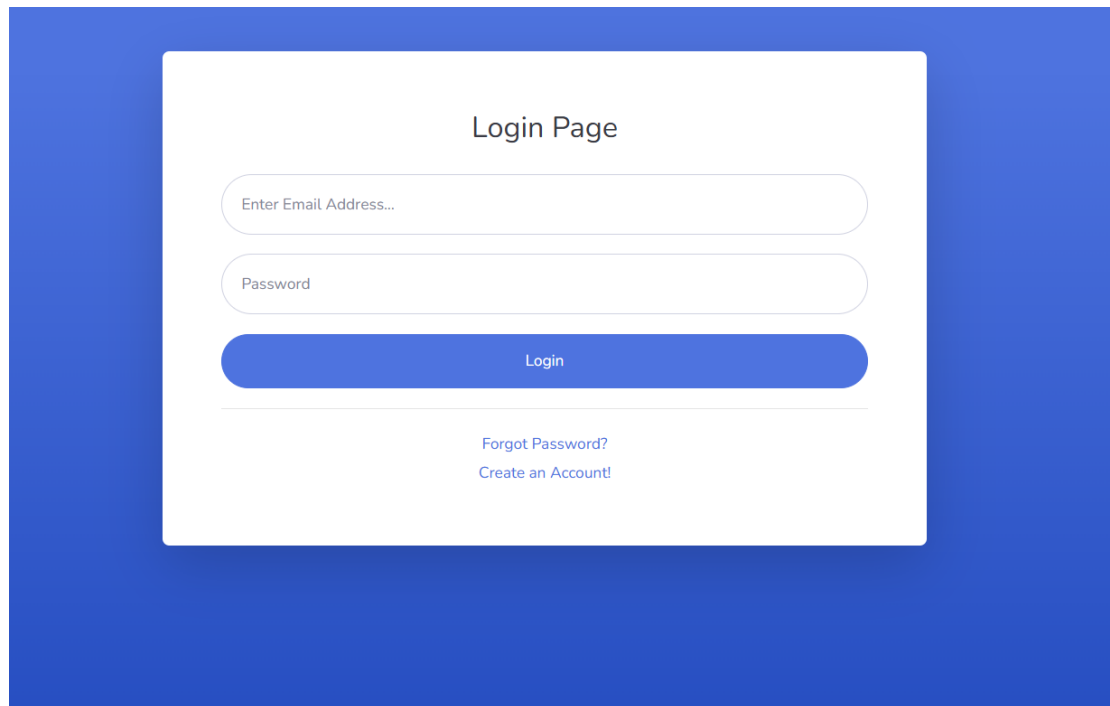
Tabel 4 <profile\_perusahaan>

Identifikasi>Nama : tbprofile\_perusahaan  
 Deskripsi isi : Menyimpan informasi profil perusahaan  
 Jenis : table data induk  
 Primary key : id

Id Field	Tipe & length	NULL	Deskripsi
id	INT(11)	No	Identifikasi unik untuk setiap data profil perusahaan
nama_perusahaan	Varchar(255)	No	Nama perusahaan
alamat	Varchar(255)	No	Alamat perusahaan
telepon	INT(12)	No	Nomor telepon perusahaan
email	Varchar(255)	No	Email perusahaan

*tabel 3 5 profile\_perusahaan*

### 3.4.2 Perancangan Antarmuka (UX/UI)



The image shows a login page with a white background and a blue border. The title "Login Page" is centered at the top. Below it are two input fields: "Enter Email Address..." and "Password". A blue "Login" button is centered below the fields. At the bottom, there are two links: "Forgot Password?" and "Create an Account!".

Login Page

Enter Email Address...

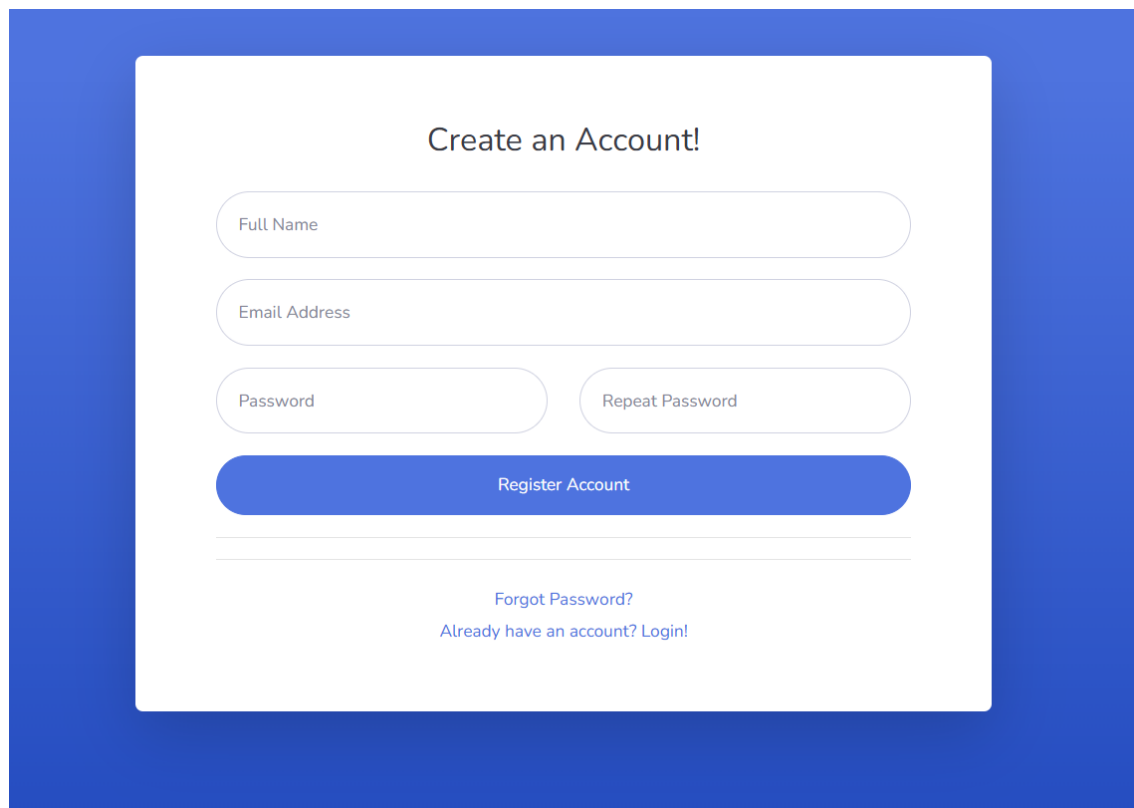
Password

Login

[Forgot Password?](#)

[Create an Account!](#)

*Gambar 3 14 Tampilan Login*



The image shows a registration page with a white background and a blue border. The title "Create an Account!" is centered at the top. Below it are four input fields: "Full Name", "Email Address", "Password", and "Repeat Password". A blue "Register Account" button is centered below the fields. At the bottom, there are two links: "Forgot Password?" and "Already have an account? Login!".

Create an Account!

Full Name

Email Address

Password

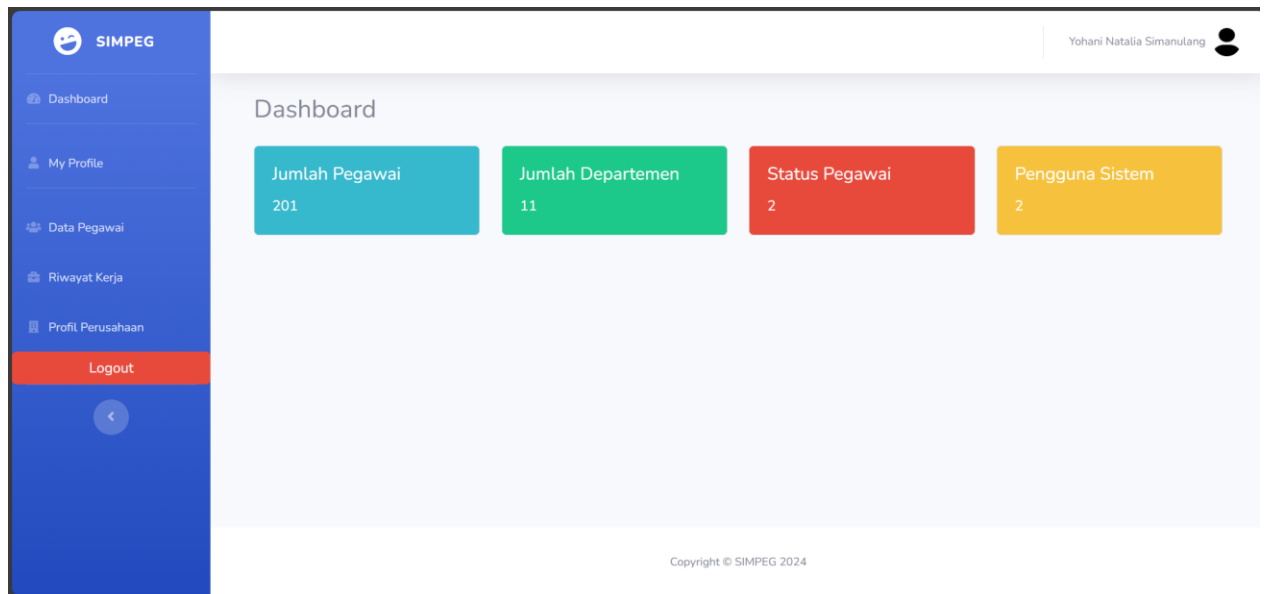
Repeat Password

Register Account

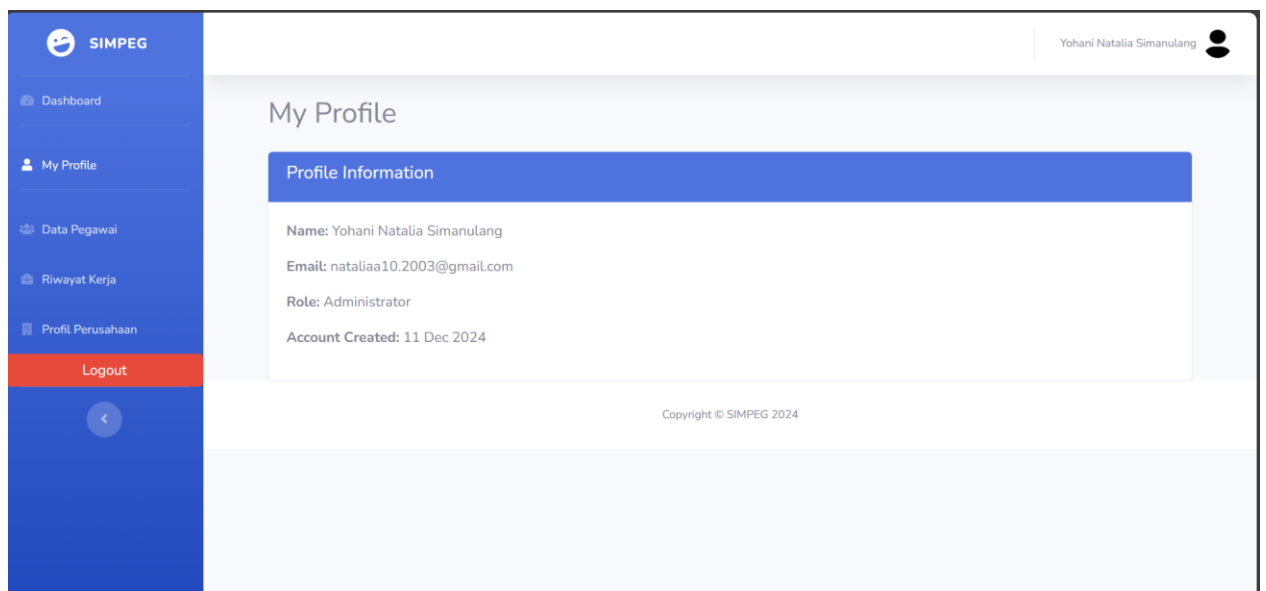
[Forgot Password?](#)

[Already have an account? Login!](#)

*Gambar 3 15 Tampilan Register*



**Gambar 3 16 Tampilan SIMPEG**



**Gambar 3 17 Tampilan myProfile Admin**

- Dashboard
- My Profile
- Data Pegawai
- Riwayat Kerja
- Profil Perusahaan
- Logout

Yohani Natalia Simanulang

Tambah Data Pegawai

Cari Data Pegawai... Cari

### Daftar Pegawai

No	Nama	Aksi
1	Winda Aulia Ariyani	<a href="#">Detail</a> <a href="#">Ubah</a> <a href="#">Hapus</a>
2	Yohani Natalia Simanulang	<a href="#">Detail</a> <a href="#">Ubah</a> <a href="#">Hapus</a>
3	Yohana Raisa S	<a href="#">Detail</a> <a href="#">Ubah</a> <a href="#">Hapus</a>
4	Fajar	<a href="#">Detail</a> <a href="#">Ubah</a> <a href="#">Hapus</a>
5	Toni Christopher.d	<a href="#">Detail</a> <a href="#">Ubah</a> <a href="#">Hapus</a>
6	Hendra Putra	<a href="#">Detail</a> <a href="#">Ubah</a> <a href="#">Hapus</a>

**Gambar 3 18 Tampilan Data Pegawai**

Form Tambah Data Pegawai

Nama

NIP

Alamat

Tanggal Lahir

dd/mm/yyyy

Jabatan

Kepala

Departemen

BAPKK

Tahun Bergabung

dd/mm/yyyy

Tambah Data

**Gambar 3 19 form Tambah Data Pegawai**



Detail Data Pegawai

Yohani Natalia Simanulang

2201020091

jl.antasari

2003-12-10

Admin

UPA TIK

2020-01-13

Kembali

Copyright © SIMPEG 2024

**Gambar 3 20 Tampilan Detail Data Pegawai**

Form Ubah Data Pegawai

Nama

Winda Aulia Ariyani

NIP

2201020083

Alamat

jl.perintis no

Tanggal Lahir

03/03/2004

Jabatan

Manager Departemen

Departemen

Tahun Bergabung

01/01/2022

Ubah Data

**Gambar 3 21 Form Ubah Data Pegawai**

- Dashboard
- My Profile
- Data Pegawai
- Riwayat Kerja
- Profil Perusahaan
- Logout

Yohani Natalia Simanulang

### List Riwayat Kerja Pegawai

Tambah Data Riwayat Kerja

Results : 201

No	Nama	Departemen	Riwayat Pendidikan	Riwayat Pengalaman	Aksi
1	Yohani Natalia Simanulang	BAPKK	S1 Ekonomi	Kepala Bagian	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2	Winda Aulia Ariyani	LPPM	S1 Teknik Informatika	Peneliti	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
3	Yohana Raisa S	LPMPP	S2 Sistem Informasi	Koordinator	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
4	Fajar	UPA TIK	S1 Ilmu Perpustakaan	Programmer	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
5	Toni Christopher.d	UPA PERPUSTAKAAN	S2 Linguistik	Pustakawan	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
6	Hendra Putra	UPA BAHASA	S1 Psikologi	Trainer	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

**Gambar 3 22 Tampilan List Riwayat Kerja**

Form Tambah Riwayat Kerja

Nama

Departemen

BAPKK

Riwayat Pendidikan

S1 Ekonomi

Riwayat Pengalaman

Tambah Data

**Gambar 3 23 Form Tambah Data Riwayat Kerja**

Form Ubah Riwayat Kerja

Nama

Yohani Natalia Simanulang

Departemen

BAPKK

Riwayat Pendidikan

S1 Ekonomi

Riwayat Pengalaman

Kepala Bagian

Ubah Data

*Gambar 3 24 Form Ubah Riwayat Kerja*

### 3.5 Implementasi Perangkat Keras dan Lunak

#### 3.5.1 Implementasi Perangkat Keras

Perangkat keras yang dibutuhkan dalam penelitian ini adalah laptop yang digunakan sebagai penempatan program dan database pada aplikasi berbasis web tersebut. Spesifikasinya sebagai berikut :

##### 1. Laptop 1

- Device : LENOVO
- Processor : 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.80GHz (8 CPUs)
- Sistem Operasi : Windows 11 Home Single Language 64-bit (10.0, Build 26100)
- RAM : 8192MB RAM
- SSD : 512 GB
- System Type : 64-bit

##### 2. Laptop 2

- Device : ASUS
- Processor : Intel(R) Celeron(R) N4020 CPU @ 1.10GHz(2 CPUs),
  - ~1.1GHz
- Sistem Operasi : Windows 11 Home Single Language 64-bit (10.0, Build 22631)
- RAM : 8192MB RAM
- SSD : 256 GB
- System Type : 64-bit

#### 3.5.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan website ini terdiri dari beberapa jenis aplikasi

##### 1. Aplikasi Pengolah Kata

Mencatat dan menyusun laporan dalam bentuk dokumen. Pada perancangan ini, proses dokumentasi dilakukan dengan menggunakan Google Document dan Microsoft Word

## 2. Aplikasi Pembangunan Sistem atau Bahasa Pemrograman

Aplikasi bahasa pemrograman digunakan untuk menerapkan perancangan ke dalam program yang dapat digunakan. Dalam perancangan implementasi ini, bahasa pemrograman yang dipakai adalah HTML, PHP JavaScript dan CSS yang dilakukan di VSC(Visual Studio Code).

## 3. Aplikasi Web Server

Aplikasi yang digunakan sebagai server adalah XAMPP dan untuk mengelola database adalah MySQL/phpMyAdmin yang bersifat open source dan mendukung dynamic web.

## 4. Aplikasi Pengelola Desain

Aplikasi ini digunakan untuk menggambar dan merancang bentuk model desain website dan database. Pada implementasi perancangan ini menggunakan Bootstrap versi 3.1.9 untuk membuat gambar rancangan desain UI/UX aplikasi berbasis website dan SBAdmin sebagai tampilan Login dan Register.

### 3.5.3 Produk Perangkat Lunak

Menampilkan hasil produk perangkat lunak yang telah dibuat beserta **link GITHUB**.  
<https://github.com/WindaAulia83/ProjectPIPL.git>

### 3.6 Pengujian Perangkat Lunak

Menjelaskan metode pengujian perangkat lunak yang digunakan

#### 3.6.1 Blackbox Testing

Jika menggunakan metode *blackbox*, bentuk pengujian dilakukan dengan cara:

No	Komponen Pengujian	Langkah Pengujian	Hasil yang Diharapkan
1	“Register” untuk melakukan pendaftaran akun admin	Menampilkan tampilan pendaftaran akun untuk login ke website SIMPEG	Berhasil menampilkan tampilan pendaftaran akun
2	“Login” untuk masuk ke aplikasi	Menampilkan tampilan login untuk masuk ke website SIMPEG	Berhasil menampilkan tampilan login
3	“Halaman SIMPEG” adalah halaman utama	Menampilkan user information, sidebar berupa dashboard, my profile, data pegawai, data riwayat kerja serta profile perusahaan.	Berhasil menampilkan halaman homepage
4	“Halaman My Profile”	Menampilkan deskripsi detail Admin	Berhasil menampilkan tampilan my profile
5	“Halaman Data Pegawai”	Menampilkan data pegawai yang bisa melakukan CRUD serta Searching	Berhasil melakukan CRUD dan Searching
6	“Halaman Data Riwayat Kerja”	Menampilkan data riwayat kerja yang bisa melakukan create, update, delete dan searching.	Berhasil melakukan CRD dan Searching

7	“Halaman Profile Perusahaan”	Menampilkan data detail profile perusahaan	Berhasil menampilkan detail profile perusahaan
8	“ Tombol Logout”	Menampilkan tombol untuk keluar	Berhasil melakukan Logout

*tabel 3 6 blackbox testing*

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Perangkat lunak ini dirancang sebagai sistem manajemen yang intuitif untuk mempermudah pengelolaan data pegawai, riwayat kerja, dan profil perusahaan. Dengan dilengkapi fitur keamanan login dan register, sistem memastikan bahwa hanya pengguna yang terdaftar dan terotorisasi dapat mengakses aplikasi. Pembagian peran sesuai otoritas membuat pengguna dapat menggunakan fitur yang relevan dengan tanggung jawabnya.

Dashboard interaktif memberikan ringkasan data penting seperti jumlah pegawai, departemen, dan pengguna sistem. Fitur CRUD yang tersedia untuk pengelolaan data pegawai dan riwayat kerja memberikan kemudahan dalam menambah, mengedit, membaca, dan menghapus data. Ditambah dengan fitur searching bar, pencarian informasi menjadi lebih efisien. Selain itu, bagian profil perusahaan menyajikan informasi penting yang dapat diperbarui sesuai kebutuhan, sehingga mendukung pengelolaan data yang akurat dan terkini. Secara keseluruhan, perangkat lunak ini dirancang untuk meningkatkan efisiensi operasional dan mempermudah komunikasi antar pengguna dalam perusahaan.

#### **4.2 Saran**

Untuk meningkatkan kualitas sistem manajemen ini, disarankan untuk menambahkan fitur notifikasi yang dapat mengingatkan admin terkait perubahan penting, seperti penambahan pegawai baru atau pembaruan riwayat kerja. Selain itu, sistem dapat dikembangkan lebih lanjut dengan mengintegrasikannya ke modul lain seperti penggajian, absensi, dan penilaian kinerja untuk memberikan solusi HR yang lebih menyeluruh. Fitur audit log juga sebaiknya disertakan untuk melacak aktivitas pengguna, sehingga keamanan dan pelacakan data lebih terjamin. Demi meningkatkan keamanan, perlu diterapkan enkripsi data sensitif seperti kata sandi dan sertifikat SSL. Penambahan fitur analisis data, seperti grafik statistik jumlah pegawai per departemen atau tren riwayat kerja, juga akan membantu pengambilan keputusan berbasis data. Selain itu, memastikan antarmuka web responsif agar dapat diakses dari berbagai perangkat, termasuk ponsel



dan tablet, akan mendukung fleksibilitas pengguna. Dengan implementasi saran-saran ini, sistem akan menjadi lebih komprehensif, aman, dan bermanfaat bagi perusahaan.