# BAB III

# ANALISIS DAN PERANCANGAN

Pada bab ini akan dijelaskan mengenai analisis sistem yang berjalan, analisis sistem yang akan dibangun, perancangan sistem (UML), dan perancangan database. Aplikasi Website untuk Prediksi Gaji Pegawai ini merupakan aplikasi yang menggunakan beberapa software yaitu Visual Studio Code, Framework Django, Jupyter Notebook, Tableau, PHPMyAdmin, dan XAMPP. Dengan aplikasi ini nantinya dapat membantu admin untuk melakukan prediksi gaji pegawai berdasarkan pengalaman lama bekerja.

## Analisis Sistem

Analisis sistem ialah penjabaran dari suatu sistem informasi yang utuh ke berbagai macam bagian-bagian komponennya yang bertujuan untuk mengevaluasi permasalahan atau kendala yang terjadi pada suatu sistem, sehingga nantinya dapat dilakukan perbaikan ataupun pengembangan pada sistem tersebut.

Perancangan sistem merupakan kegiatan merancang dan mendesain suatu sistem yang baik yang dimana kegiatan tersebut adalah langkah-langkah operasi dalam proses pengolahan data dan prosedur-prosedur untuk mendukung operasi sistem tersebut. Tujuan dari perancangan sistem ialah untuk memenuhi kebutuhan para pemakai sistem serta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada programmer dan ahli-ahli yang terlibat didalamnya.

Pada bagian ini, dibahas tentang analisis prosedur yang digambarkan dalam bentuk flowmap BPMN, pengkodean, analisis sistem fungsional, dan analisis sistem non fungsional yang meliputi perangkat keras dan perangkat lunak yang digunakan. Selain itu pada bagian ini juga akan dibahas mengenai analisis user yang terlibat dalam aplikasi tersebut. Tahapan ini sangat penting dalam membantu melanjutkan tahapan yang selanjutnya yaitu tahapan perancangan.

### Analisis Sistem Yang Sedang Berjalan

Pada sub bab ini akan dijelaskan mengenai proses prediksi gaji pegawai yang sedang berjalan. Analisa sistem yang sedang berjalan bertujuan untuk mengetahui dan menggambarkan lebih lanjut mengenai bagaimana cara kerja sistem tersebut. Sistem yang berjalan saat ini menjelaskan tentang flowmap proses prediksi gaji pegawai. Flowmap tersebut merupakan gambaran alur proses prediksi gaji pegawai yang sedang berjalan pada aplikasi yang nantinya akan dibangun.



Gambar 3. 1 Flowmap Pemrosesan Gaji

### Analisis Sistem Yang Akan Dibangun

Pada sub bab ini akan dijelaskan mengenai Aplikasi Website Prediksi Gaji Pegawai Berdasarkan Pengalaman Lama Bekerja akan dibangun. Analisa sistem yang sedang berjalan memiliki tujuan untuk memberikan gambaran dan mengetahui lebih lanjut bagaimana cara kerja sistem tersebut. Strategi yang digunakan dalam menganalisis sistem yang akan dibangun ini, adalah dengan membongkar atau menterjemahkan dalam bentuk flowmap BPMN.

#### Flowmap *Login*

Pada bagian ini menjelaskan tentang flowmap proses *login* admin untuk masuk ke aplikasi website prediksi gaji. Flowmap tersebut merupakan gambaran alur proses *login* admin yang akan dibangun pada aplikasi ini.

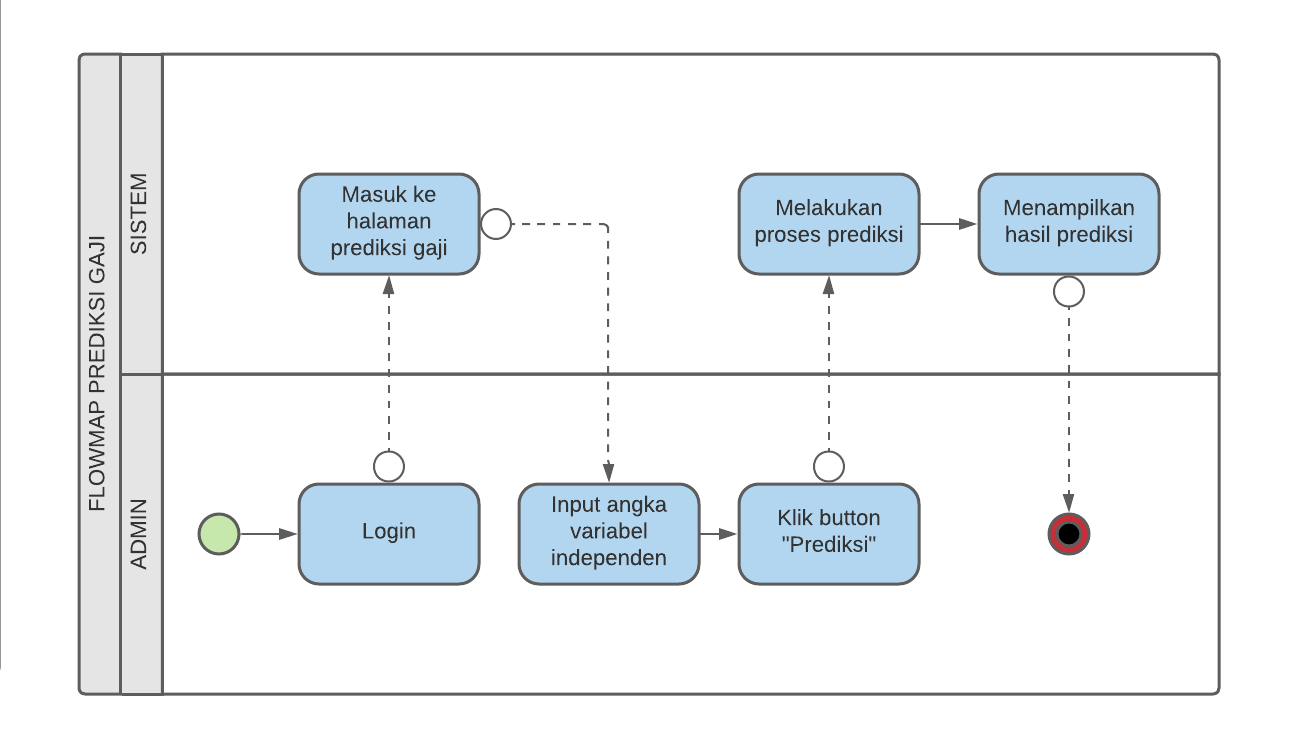
Gambar 3. 2 Flowmap Proses Login Admin

Keterangan :

1. Admin dapat membuka Aplikasi Prediksi Gaji Pegawai.
2. Aplikasi menampilkan halaman *login* .
3. Admin dapat melakukan penginputan username serta password.
4. Setelah itu, sistem dapat melakukan pengecekan apakah username serta password yang diinputkan admin valid atau tidak.
5. Jika username dan password sesuai, maka admin dapat masuk ke halaman .utama aplikasi. Sedangkan jika username dan password tidak sesuai, maka admin akan tetap berada di halaman *login.*

#### Flowmap Prediksi Gaji

Pada bagian ini menjelaskan tentang flowmap proses prediksi gaji pegawai yang dilakukan oleh admin. Flowmap tersebut merupakan gambaran alur proses prediksi gaji pegawai oleh admin yang akan dibangun pada aplikasi ini.



Gambar 3. 3 Flowmap Proses Prediksi Gaji Pegawai

Keterangan :

1. Admin dapat melakukan *login.*
2. Setelah admin berhasil *login*, aplikasi akan menampilkan halaman dashboard.
3. Setelah itu, admin dapat memilih menu Prediksi untuk beralih ke halaman prediksi.
4. Pada halaman prediksi, admin dapat menginputkan angka berupa variabel independent (usia, level pekerjaan, total tahun bekerja, dan tahun di perusahaan) pada form yang disediakan.
5. Setelah diinputkan, aplikasi akan menampilkan hasil prediksi gaji pegawai berdasarkan pengalaman lama bekerja.

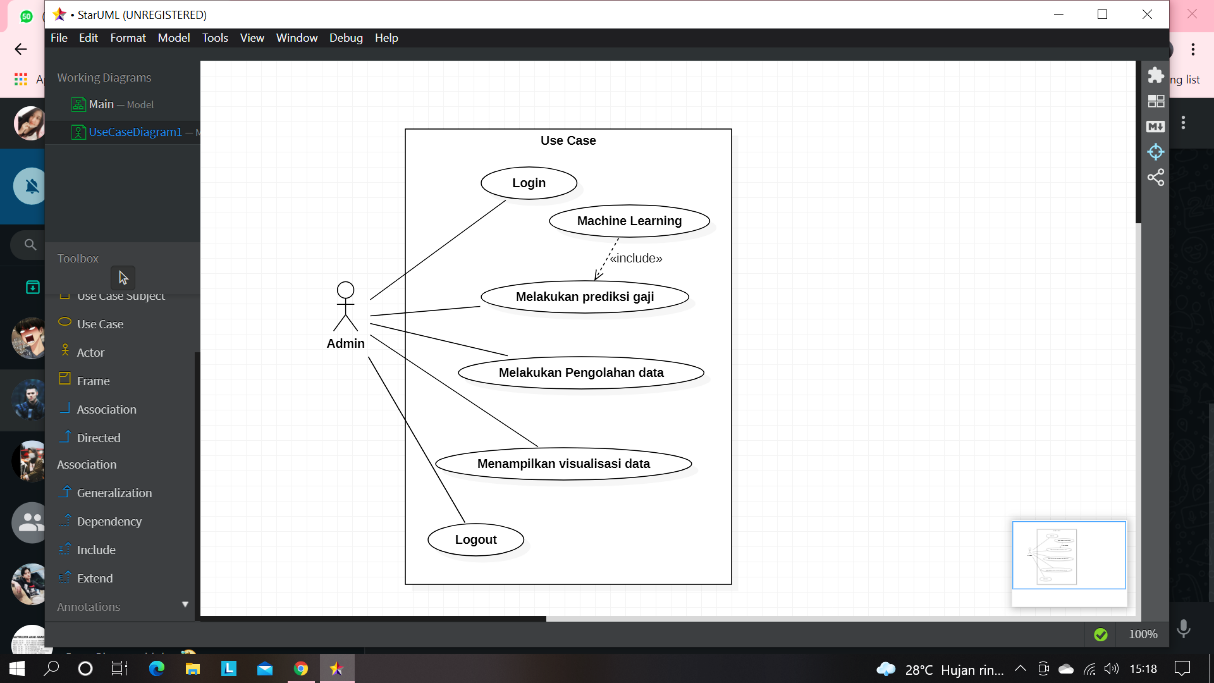
## Perancangan Sistem (UML)

UML adalah singkatan dari Unified Modeling Language yang didefinisikan sebagai sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau *software* berbasis objek. UML juga dapat dikatakan sebagai bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan, dan membangun suatu sistem perangkat lunak.

UML adalah suatu bentuk metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tools* untuk mendukung pengembangan sistem tersebut. Sehingga, UML juga dapat menjadi salah satu cara untuk mempermudah dalam melakukan pengembangan aplikasi yang berkelanjutan. Oleh karenanya, UML juga dapat menjadi suatu alat bantu untuk transfer ilmu tentang sistem yang akan dikembangkan dari satu developer ke developer lainya.

### Use Case Diagram

Use case diagram yaitu suatu gambaran graphical untuk memodelkan seluruh proses bisnis berdasarkan perspektif pengguna sistem dari beberapa atau semua aktor, use case, dan interaksi yang memperkenalkan suatu sistem. Use case diagram secara sederhana merupakan sebuah sarana bantu untuk melakukan pendefinisian apa yang ada di luar sistem (aktor) dan apa yang harus dilakukan oleh sistem yang sedang dikembangkan.



Gambar 3. 4 Use Case Diagram

1. **Definisi Aktor**

Pada bagian ini akan dijelaskan actor-aktor yang terlihat dalam (judul proyek).

Tabel 3. 1 Definisi Aktor

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1 | Admin | * Login * Melakukan Prediksi Gaji * Melakukan Pengolahan Data * Melakukan Visualisasi Data * Logout |

1. **Definisi Use Case**

Tabel 3. 2 Definisi Use Case

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1 | Machine Learning | Merupakan aktivitas interaksi antara manusia dengan mesin. Dalam hal ini, admin membuat model prediksi *Machine Learning*. |
| 2 | Login | Merupakan aktivitas login yang dilakukan oleh admin. Sebelum admin masuk ke aplikasi ia harus melakukan login terlebih dahulu. |
| 3 | Melakukan Prediksi Gaji | Merupakan aktivitas memprediksi gaji karyawan yang dilakukan oleh admin dengan parameter lama bekerja seorang karyawan. |
| 4 | Melakukan Pengolahan Data | Merupakan aktivitas insert, read, update dan delete data karyawan yang dilakukan oleh admin. |
| 5 | Menampilkan Visualisasi Data | Merupakan aktivitas visualisasi grafik data karyawan yang dilakukan oleh admin. |
| 6 | Logout | Merupakan aktivitas logout yang dilakukan oleh admin. Admin dapat keluar dari aplikasi jika telah selesai melakukan pekerjaan pada aplikasi. |

1. **Skenario Use Case**

Skenario use case diharapkan setelah berjalannya fungsional use case. Selain itu juga diberikan ulasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang dilakukan oleh actor. Setiap use case akan diberikan sebuah scenario yang akan menjelaskan secara detail interaksi yang ada di dalamnya.

Tabel 3. 3 Skenario Use Case Machine Learning

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Machine Learning |
| **Tujuan** | Pembuatan model prediksi |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Membuat model prediksi | * Melakukan olah data gaji karyawan |
| **Kondisi Akhir** | Masuk pada aplikasi |

Tabel 3. 4 Skenario Use case Login Admin

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Login |
| **Tujuan** | Sebelum masuk pada aplikasi |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Input username dan password | * Mengecek data dan menentukan aktor tersebut admin atau user |
| **Kondisi Akhir** | Masuk pada aplikasi |

Tabel 3. 5 Skenario Use case Melakukan Prediksi Gaji Karyawan

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Melakukan Predikis Gaji Karyawan |
| **Tujuan** | Melakukan Predikis Gaji Karyawan |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Menginput Data | * Melakukan Prediksi Gaji |
| **Kondisi Akhir** | Admi dapat memprediksi gaji |

Tabel 3. 6 Skenario Use case Melakukan Visualisasi Data

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Melakukan Visualisasi Data |
| **Tujuan** | Melakukan visualisasi data karyawan |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Input data karyawan | * Menampilkan grafik visualisasi data karyawan |
| **Kondisi Akhir** | Admin dapat memvisualisasikan data |

Tabel 3. 7 Skenario Use case Melakukan Pengolahan Data

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Melakukan Pengolahan Data |
| **Tujuan** | Melakukan pengolahan data karyawan |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Insert, read, update, dan delete data karyawan. | * Mengubah data karyawan yang diubah oleh admin |
| **Kondisi Akhir** | Admin dapat mengelola data |

Tabel 3. 8 Skenario Use case Logout Admin

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Logout |
| **Tujuan** | Keluar dari aplikasi |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Menekan tombol logout | * Melakukan proses keluar dari aplikasi |
| **Kondisi Akhir** | Keluar dari aplikasi |

### Class Diagram

Class diagram ialah diagram UML yang mendeskripsikan suatu struktur dari sebuah sistem yang dibuat dari kelas-kelas dengan relasi - relasinya. Class diagram juga dapat menggambarkan jenis-jenis objek yang terdapat pada sistem dan berbagai hubungan statis yang terdapat di antara mereka. Class diagram ini dapat menunjukkan properti dan operasi sebuah kelas serta batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

### Sequence Diagram

Sequence diagram adalah diagram yang mendeskripsikan serta menjelaskan suatu interaksi objek yang berdasarkan urutan waktu. Interaksi tersebut diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Sequence diagram dapat digunakan untuk menggambarkan urutan atau prosedur yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada use case diagram.

1. ***Sequence* Diagram *Login* Admin**

Berikut ini merupakan *sequence* diagram Login Admin menjelaskan hubungan antara admin pada aplikasi.

Berikut Sequence Diagram Login Admin pada gambar.



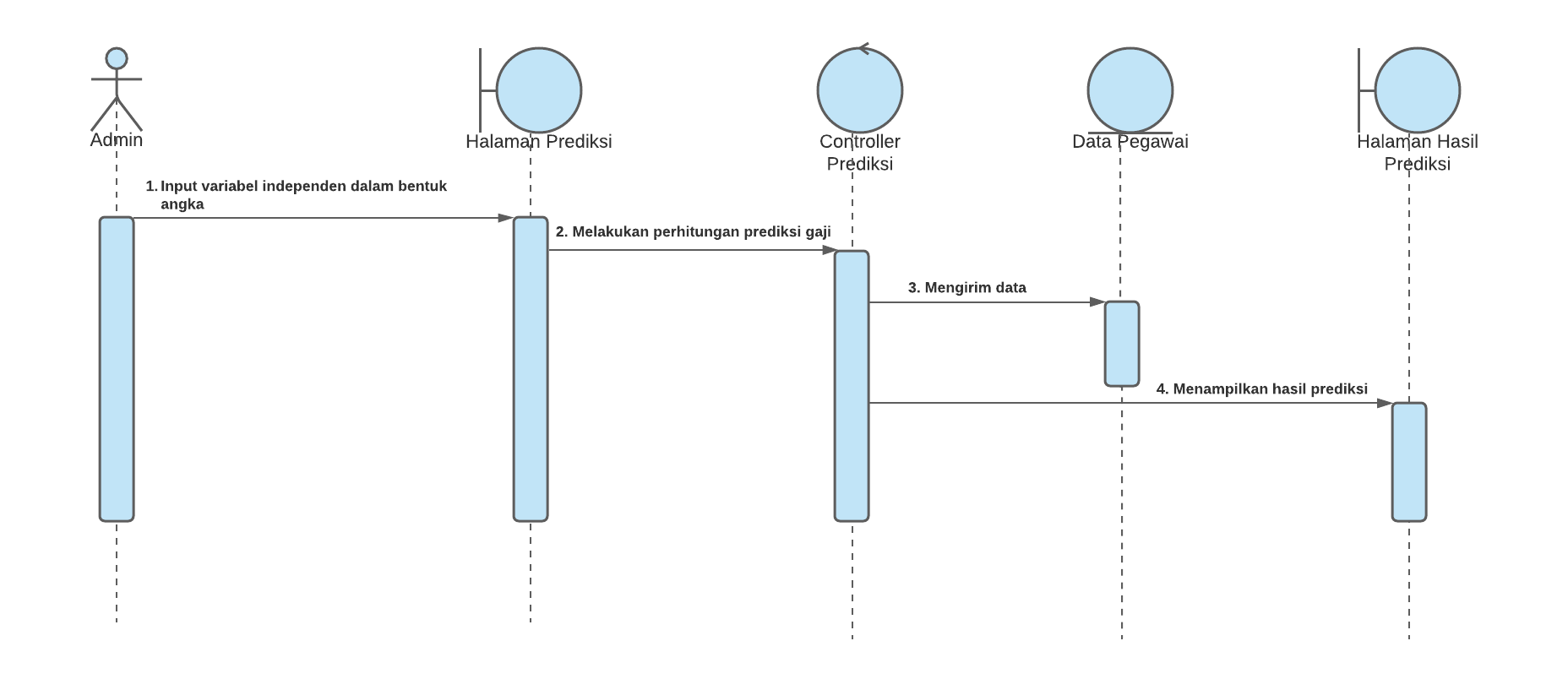
Gambar 3. 5 Sequence Diagram Login Admin

Keterangan :

1. Admin menginputkan *username* dan *password* di halaman login.
2. Validasi username dan password diteruskan ke Controller login.
3. Controller login mengambil data ke database data admin untuk validasi login.
4. Jika username dan password valid pada data admin maka akan ditampilkan halaman dashboard.
5. Jika username dan password tidak valid pada data admin maka akan ditampilkan halaman login.
6. **Sequence Diagram Melakukan Prediksi Gaji**

Berikut ini merupakan sequence diagram melakukan predikis gaji menjelaskan hubungan antara admin dan aplikasi. Admin dapat melakukan prediksi gaji dengan cara menginput data.

Berikut Sequence Diagram Melakukan Prediksi Gaji pada gambar.



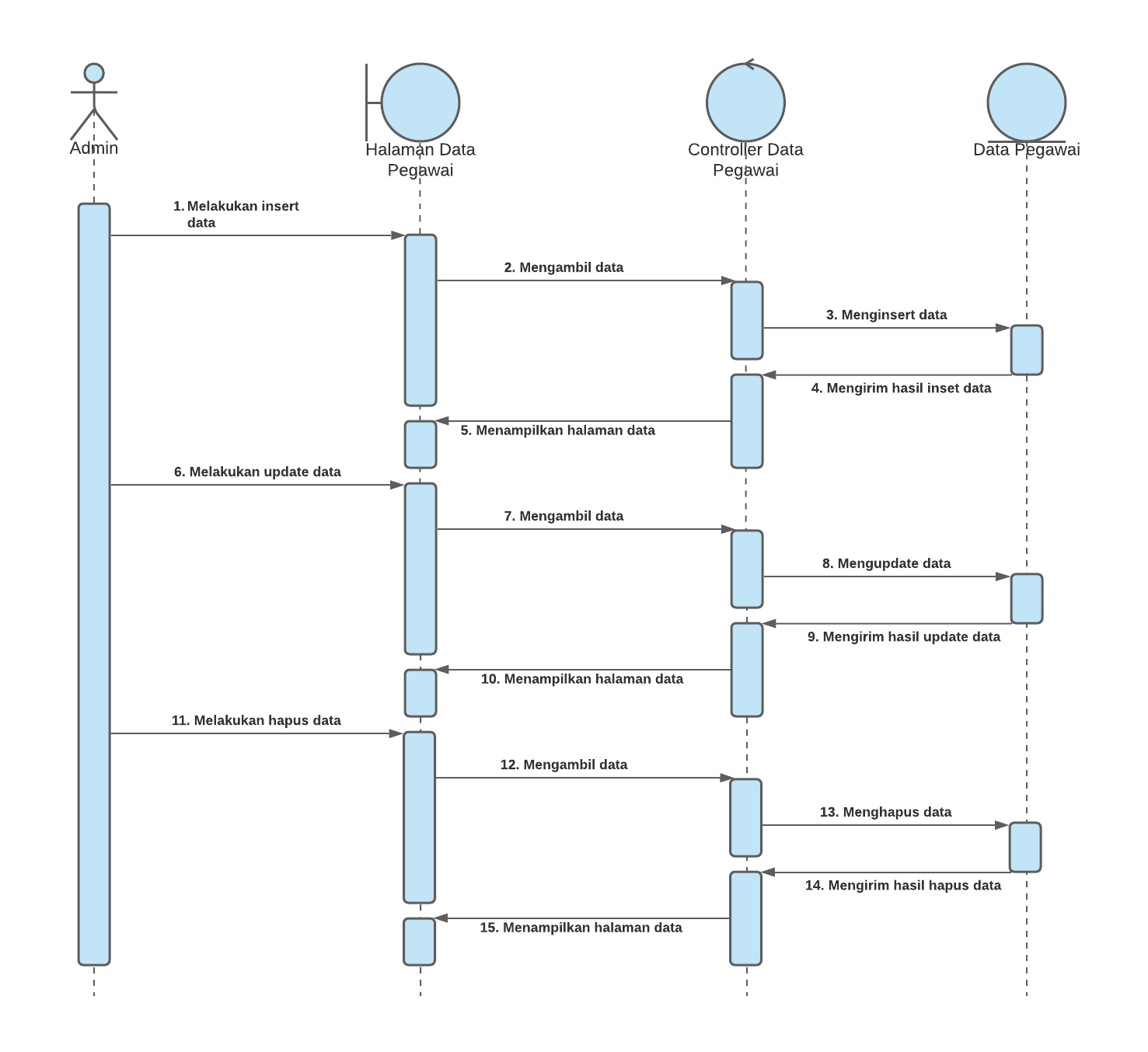
Gambar 3. 6 Sequence Diagram Melakukan Prediksi Gaji

Keterangan :

1. Admin menginputkan independen dalam bentuk angka.
2. Sistem melakukan perhitungan prediksi gaji.
3. Controller prediksi mengirim data ke database data pegawai dan memproses data.
4. Hasil data yang dihitung akan ditampilkan pada halaman hasil prediksi.
5. **Sequence Diagram Melakukan Pengolahan Data**

Berikut ini merupakan sequence diagram mengelola data menjelaskan hubungan antara admin pada aplikasi. Admin dapat insert, read, update, dan delete pada data alat.

Berikut Sequence Diagram Melakukan Pengolahan Data pada gambar.



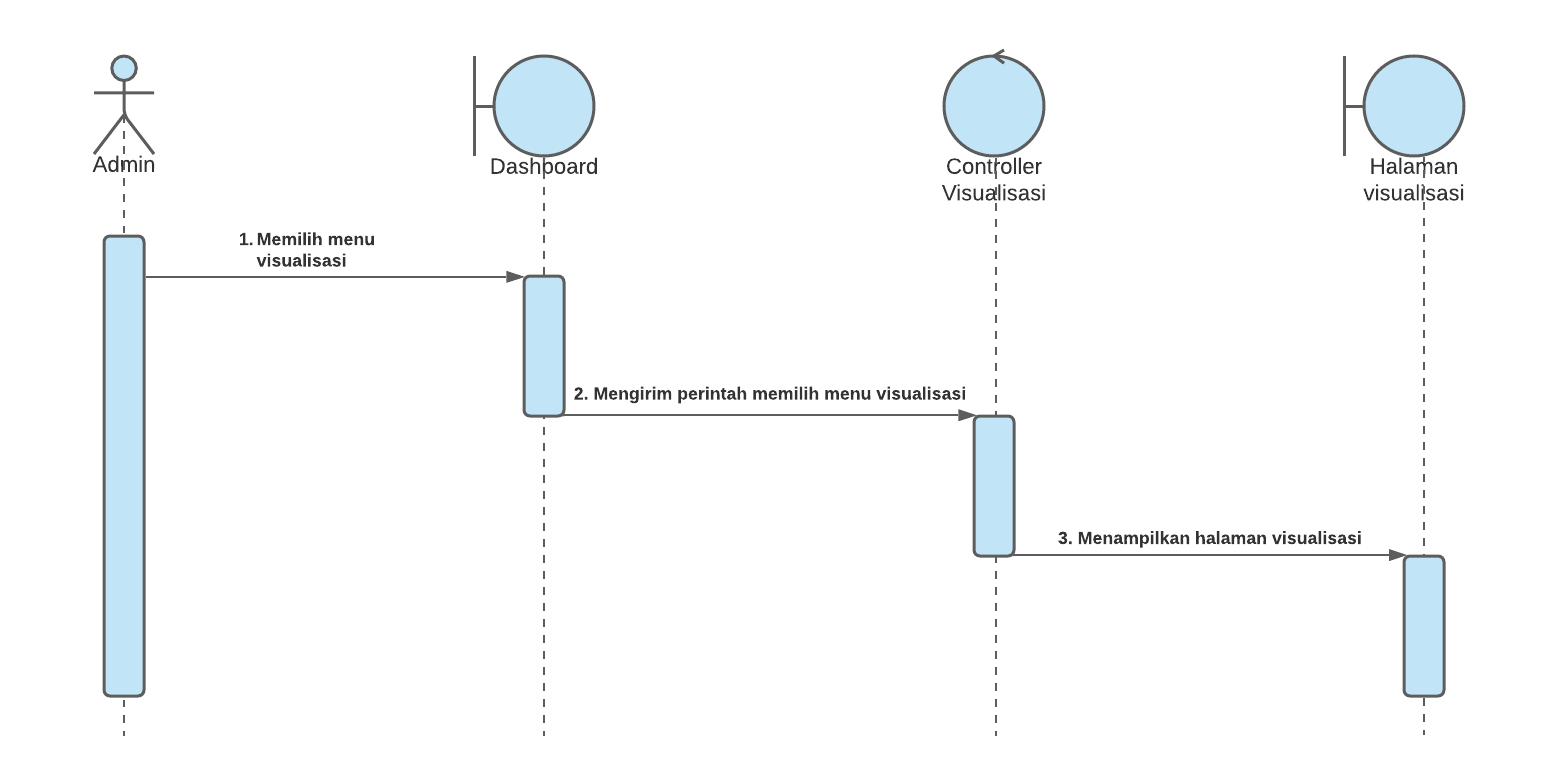
Gambar 3. 7 Sequence Diagram Melakukan Pengolahan Data

Keterangan :

1. Admin memilih menu yang akan dikelola pada menu di halaman data pegawai. Menu yang ada yaitu tambah data, update data, dan hapus data.
2. Halaman data pegawai mengambil data (baik antara input, update atau hapus) untuk diteruskan ke controller data pegawai.
3. Controller data pegawai mengirim hasil olah data ke data pegawai.
4. Data pegawai mengirim hasil olah data ke controller data pegawai.
5. Controller data pegawai menampilkan halaman data pegawai.
6. **Sequence Diagram Melakukan Visualisasi Data**

Berikut ini merupakan sequence diagram melakukan visualisasi data menjelaskan hubungan antara admin pada aplikasi.

Berikut Sequence Diagram Melakukan Visualisasi Data pada gambar.



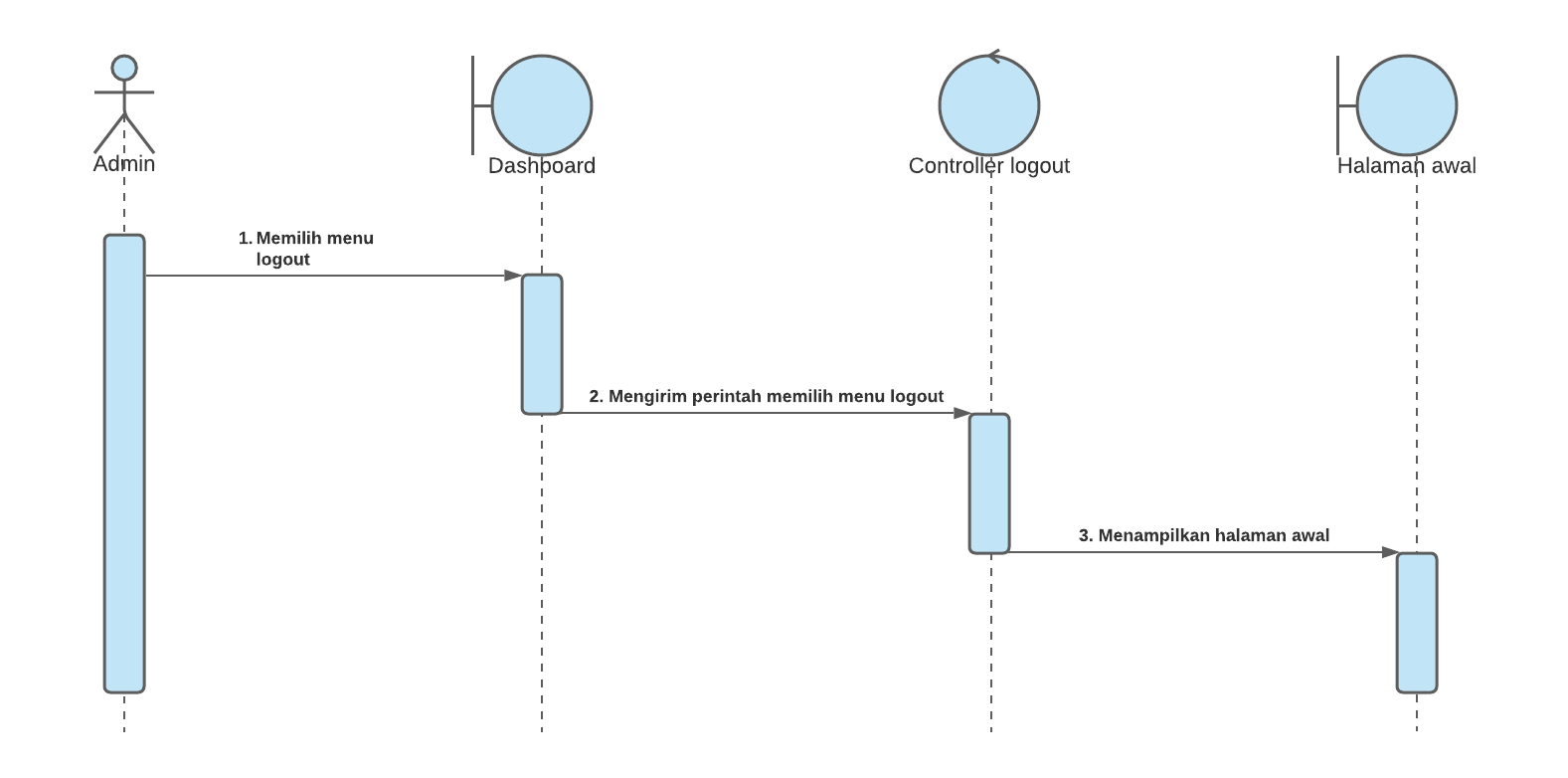
Gambar 3. 8 Sequence Diagram Melakukan Visualisasi Data

Keterangan :

1. Admin memilih menu halaman visualisasi pada dashboard.
2. Dashboard mengirim perintah memilih menu visualisasi ke controller visualisasi.
3. Controller visualisasi menampilkan halaman visualisasi.
4. **Sequence Diagram Melakukan Visualisasi Data**

Berikut ini merupakan sequence diagram *logout* Admin menjelaskan hubungan antara admin pada aplikasi.

Berikut Sequence Diagram *logout* Admin pada gambar.



Gambar 3. 9 Sequence Diagram logout admin

Keterangan :

1. Admin memilih menu logout pada dashboard.
2. Dashboard mengirim perintah memilih menu logout ke controller logout.
3. Controller logout melakukan proses logout kemudian menuju untuk menampilkan halaman awal.

### Collaboration Diagram

Collaboration diagram dapat dipakai untuk memodelkan interaksi antar objek yang ada di dalam sistem. Berbeda dari sequence diagram yang lebih menunjukkan kronologis dari operasi-operasi yang dilakukan, collaboration diagram ini lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.

1. **Collaboration Diagram Melakukan Login**



Gambar 3. 10 Collaboration Diagram Melakukan Login

1. **Collaboration Diagram Melakukan Prediksi Gaji**



Gambar 3. 11 Collaboration Diagram Melakukan Prediksi Gaji

1. **Collaboration Diagram Melakukan Pengolahan Data**



Gambar 3. 12 Collaboration Diagram Melakukan Visualisasi Data

1. **Collaboration Diagram Melakukan Visualisasi Data**

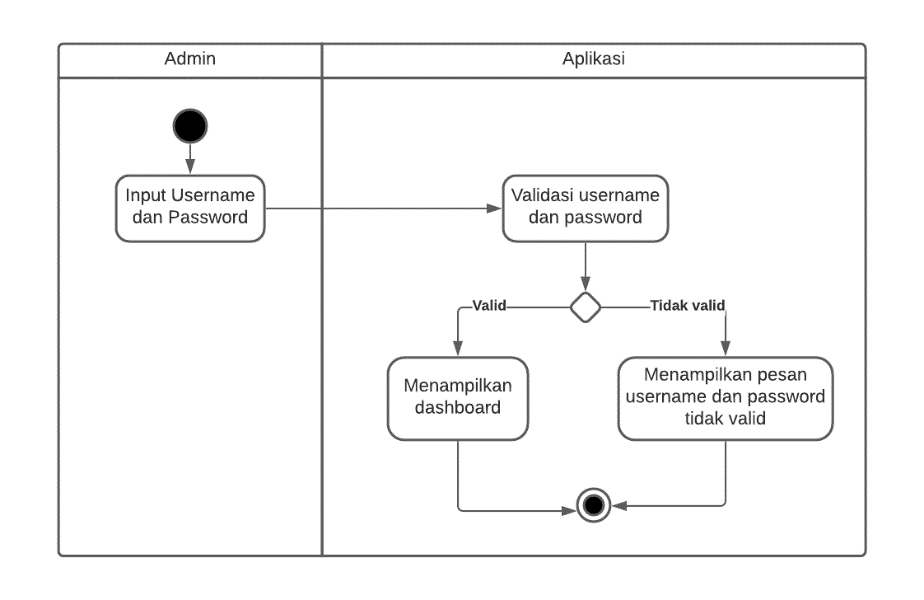


Gambar 3. 13 Collaboration Diagram Melakukan Visualisasi Data

### Activiy Diagram

Activity diagram dapat didefinisikan sebagai diagram yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Activity diagram adalah suatu teknik yang digunakan untuk mendeskripsikan logika procedural, proses bisnis serta aliran kerja dalam banyak kasus. Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, hasil akhir yang mungkin terjadi, hingga bagaimana mereka berakhir.

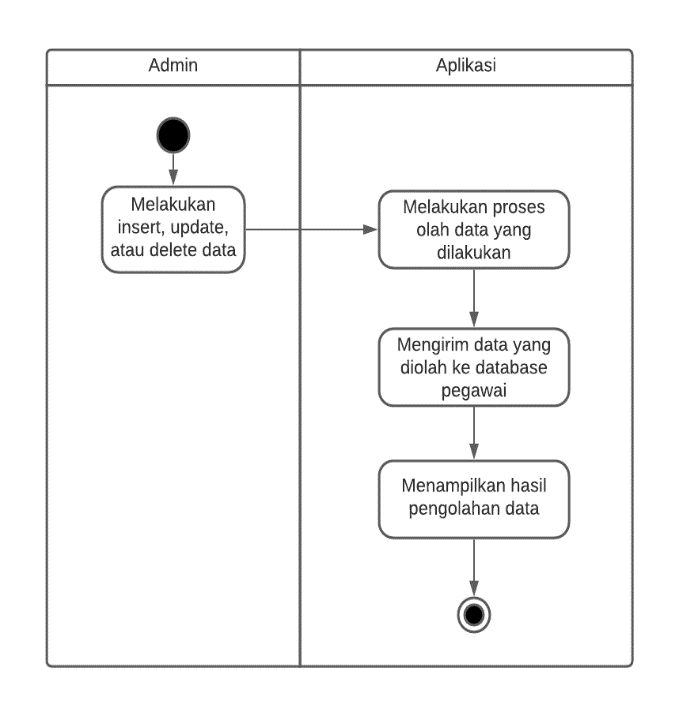
1. **Activity Diagram Login Admin**



Gambar 3. 14 Activity Diagram Login Admin

Keterangan :

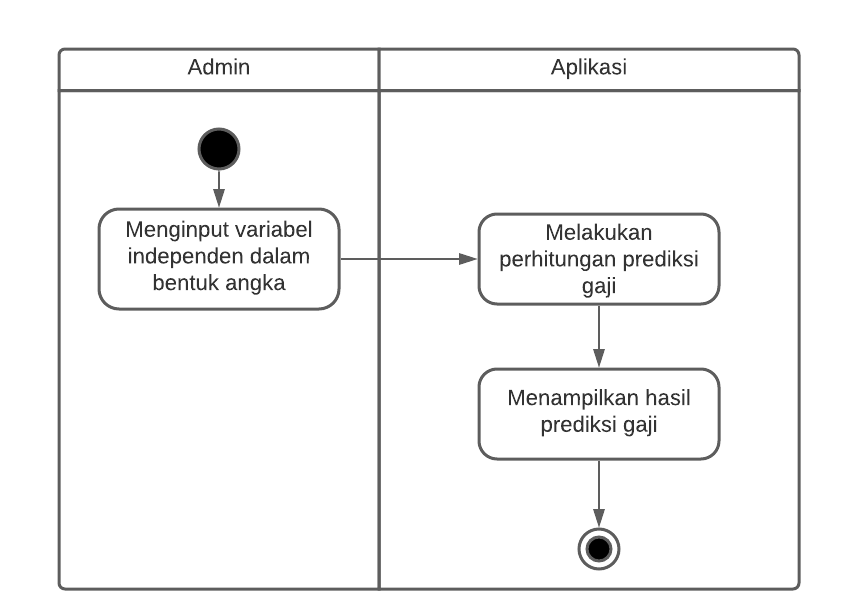
1. Admin menginput username dan password.
2. Aplikasi melakukan validasi username dan password.
3. Jika valid akan menampilkan halaman dashboard.
4. Jika tidak valid menampilkan pesan username dan password tidak valid.
5. **Activity Diagram Melakukan Pengolahan Data**



Gambar 3. 15 Activity Diagram Melakukan Pengolahan Data

Keterangan :

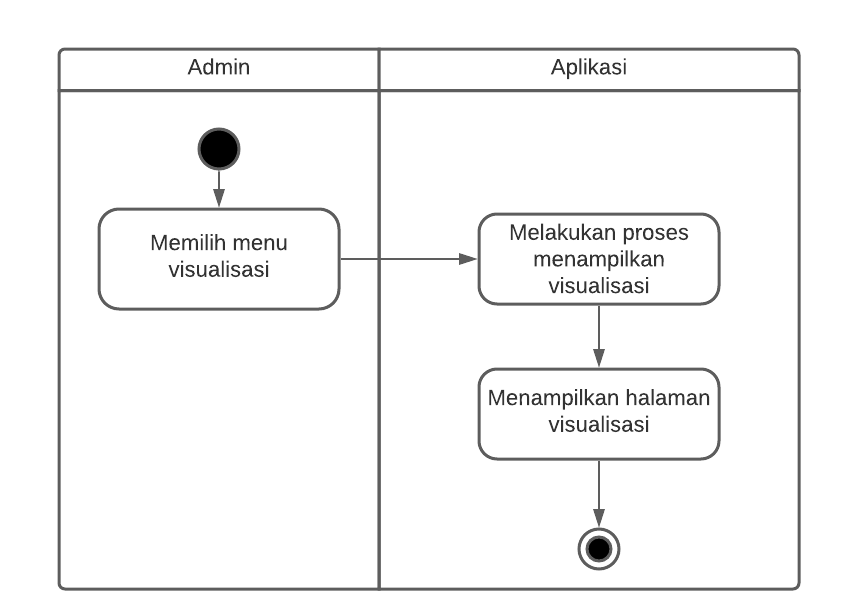
1. Admin melakukan olah data seperti insert, update, dan delete data.
2. Aplikasi melakukan proses olah data yang dilakukan.
3. Aplikasi mengirim data yang diolah ke database pegawai.
4. Aplikasi menampilkan data hasil create/update/delete.
5. **Activity Diagram Melakukan Prediksi Gaji**



Gambar 3. 16 Activity Diagram Melakukan Prediksi Gaji

Keterangan :

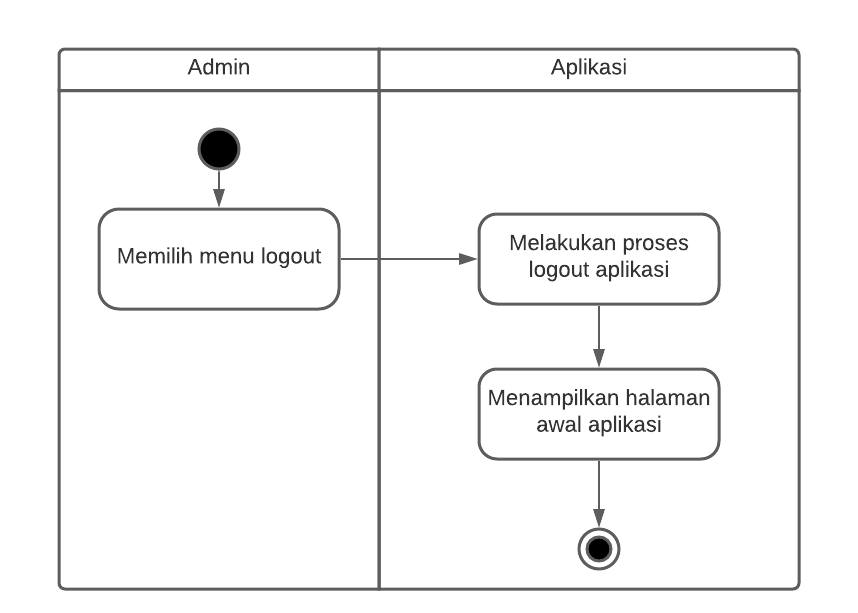
1. Admin menginputkan variabel independent dalam bentuk angka pada form prediksi.
2. Aplikasi melakukan proses perhitungan prediksi gaji.
3. Aplikasi menampilkan hasil prediksi.
4. **Activity Diagram Melakukan Visualisasi**



Gambar 3. 17 Activity Diagram Melakukan Visualisasi

Keterangan :

1. Admin memilih menu visualisasi.
2. Aplikasi melakukan proses menampilkan visualisasi.
3. Aplikasi menampilkan halamann visualisasi.
4. **Activity Diagram Logout Admin**



Gambar 3. 18 Activity Diagram Logout Admin

### Statechart Diagram

Statechart Diagram dapat menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) dari suatu objek pada sistem sebagai akibat dari stimuli yang diterimanya. Pada umumnya statechart diagram dapat menjelaskan atau menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).

1. **Statechart Diagram Login Admin**



Gambar 3. 19 Statechart Diagram Login Admin

1. **Statechart Diagram Melakukan Prediksi**



Gambar 3. 20 Statechart Diagram Melakukan Prediksi

1. **Statechart Diagram Pengolahan Data**



Gambar 3. 21 Statechart Diagram Pengolahan Data

1. **Statechart Diagram Visualisasi**



Gambar 3. 22 Statechart Diagram Visualisasi

### Component Diagram

Component diagram dapat menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) diantaranya. Komponen piranti lunak atau yang biasa disebut dengan perangkat lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun runtime. Pada umumnya komponen dapat terbentuk dari beberapa class dan atau package, tapi dapat juga dibentuk dari komponen-komponen yang lebih kecil.



Gambar 3. 23 Component Diagram

### Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana suatu komponen di-*deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server atau piranti keras apa), dan bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi, server, serta hal-hal lain yang bersifat fisik. Sebuah *node* terdiri dari server, *workstation*, atau piranti keras lain yang digunakan untuk melakukan *deploy* komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat pula didefinisikan dalam diagram ini. (Dharwiyanti : 2003)



Gambar 3. 24 Deployment Diagram

## Perancangan Database

Perancangan database merupakan proses untuk menentukan dan pengaturan data yang dibutuhkan untuk mendukung rancangan sistem, agar teciptanya pemrosesan data yang lebih efisien. Struktur tabel meliputi nama tabel, tipe data, nama atribut dan data relasi seperti primary dan foreign key.

### CDM (Conceptual Data Model)



Gambar 3. 25 Conceptual Data Model

### PDM (Physical Data Model)



Gambar 3. 26 Physical Data Model

### ERD (Entity Relationship Diagram)



Gambar 3. 27 Entity Relationship Diagram

## Struktur Menu

Berikut adalah struktur menu dari aplikasi Prediksi Gaji Pegawai secara keseluruhan :



Gambar 3. 28 Struktur Menu Aplikasi

## Antarmuka (*Interface*)

Impelementasi sistem memiliki bagian berupa pemaparan mengenai tampilan pada sistem serta kegunaan dari setiap halaman. Implementasi dari hasil perancanggan menggunakan bahasa pemrograman python dengan framework Django.

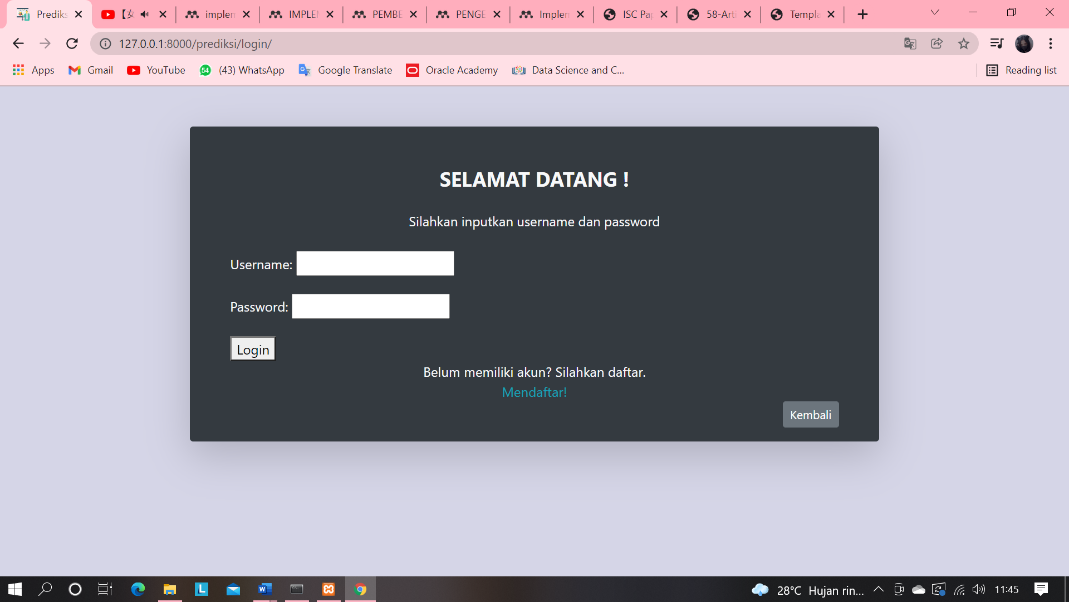
### Antarmuka Halaman Depan



Gambar 3. 29 Halaman Depan Sistem

Gambar diatas merupakan implementasi dari halaman depan sistem. Pada halaman ini, *user* dapat menekan button “Login” pada navbar untuk masuk ke dalam sistem.

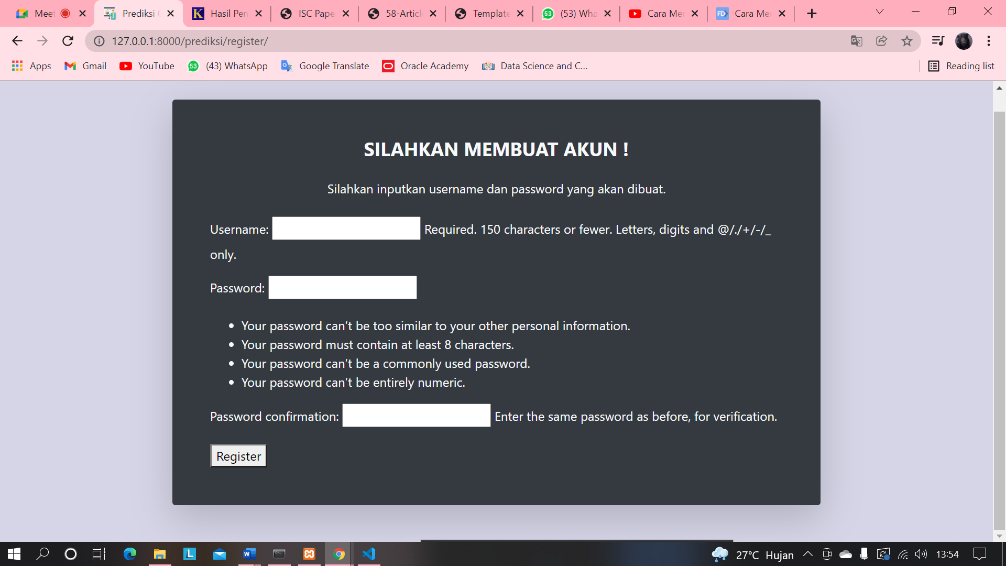
### Antarmuka Halaman *Login*



Gambar 3. 30 Halaman Login Sistem

Gambar diatas merupakan implementasi dari halaman *login*. Sebelum masuk ke halaman *dashboard*, *user* harus menginputkan *username* dan *password* yang sesuai. Jika sesuai, maka *user* akan di arahkan ke halaman *dashboard*. Jika tidak *user* akan diminta kembali memasukkan *username* dan *password* yang sesuai.

### Antarmuka Halaman Registrasi



Gambar 3. 31 Halaman Registrasi Sistem

Gambar diatas merupakan implementasi dari halaman registrasi. Jika *user* belum memiliki akun untuk masuk sistem, maka *user* dapat melakukan registrasi dengan cara menginputkan *username* dan *password* lalu menekan tombol register.

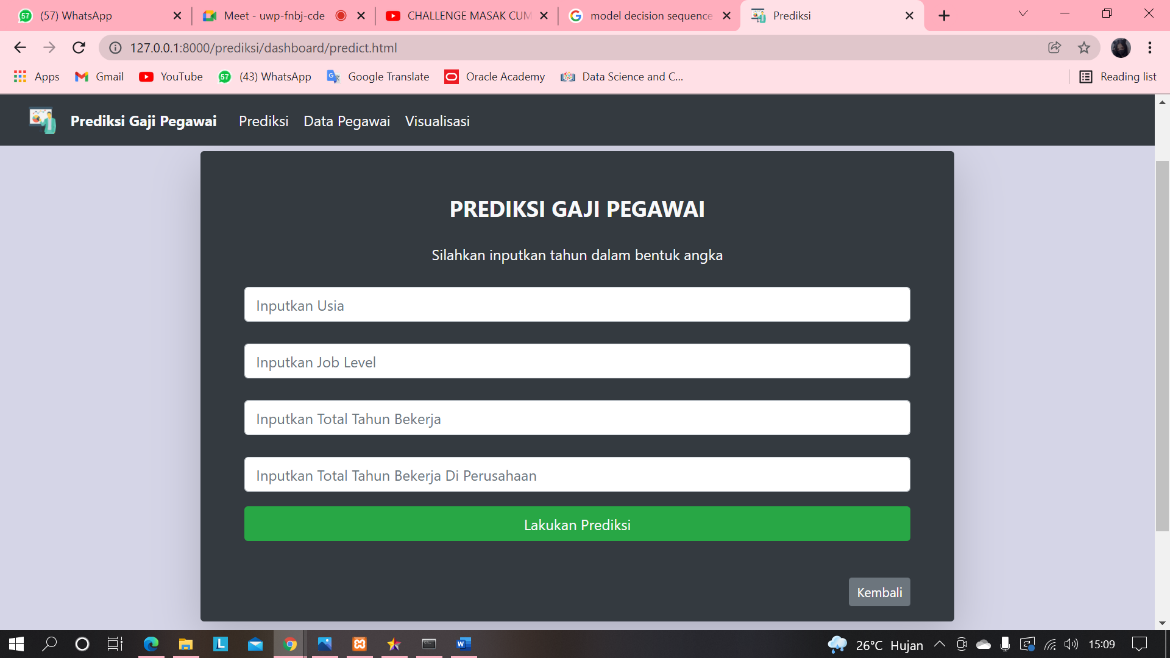
### Antarmuka Halaman *Dashboard*



Gambar 3. 32 Halaman Dashboard Sistem

Gambar diatas merupakan implementasi dari halaman *dashboard* dari sistem yang dirancang. Pada bagian navbar, terdapat pilihan submenu untuk mengakses suatu halaman yang lainnya. Pilihan submenu tersebut terdiri dari Prediksi, Data Pegawai, serta Visualisasi.

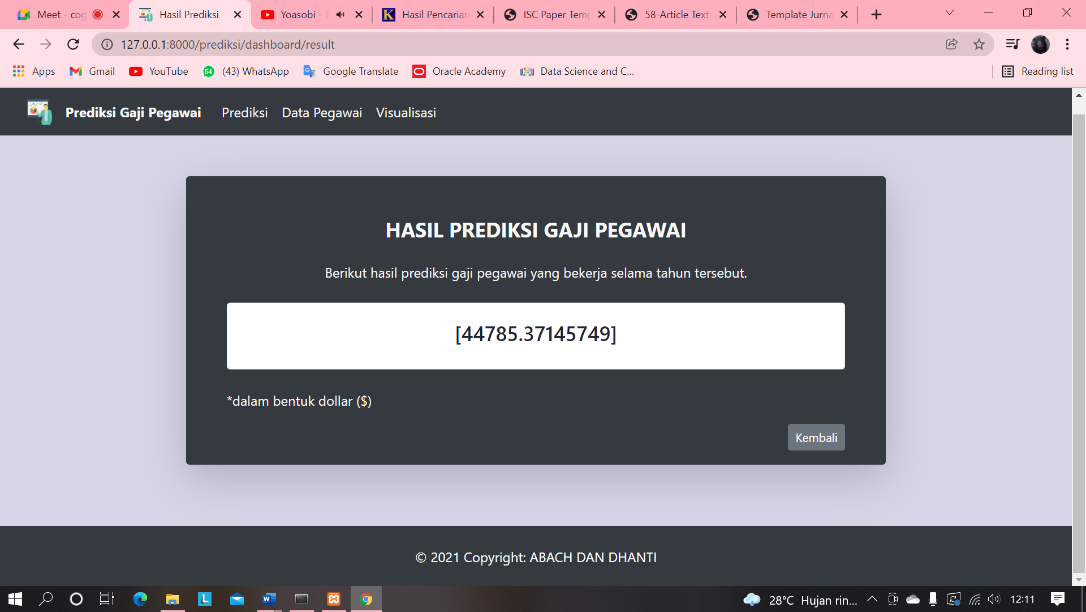
### Antarmuka Halaman Prediksi



Gambar 3. 33 Halaman Prediksi Sistem

Gambar diatas merupakan implementasi dari halaman prediksi dari sistem yang dirancang. Pada halaman ini, terdapat form untuk melakukan prediksi gaji pegawai. *User* dapat menginputkan angka berupa usia, job level, total tahun bekerja, dan total tahun bekerja di perusahan ke dalam form untuk mengetahui hasil prediksi gaji pegawai. Kemudian *user* dapat menekan button “Lakukan Prediksi” untuk melihat hasil prediksi berdasarkan tahun yang diinputkan.

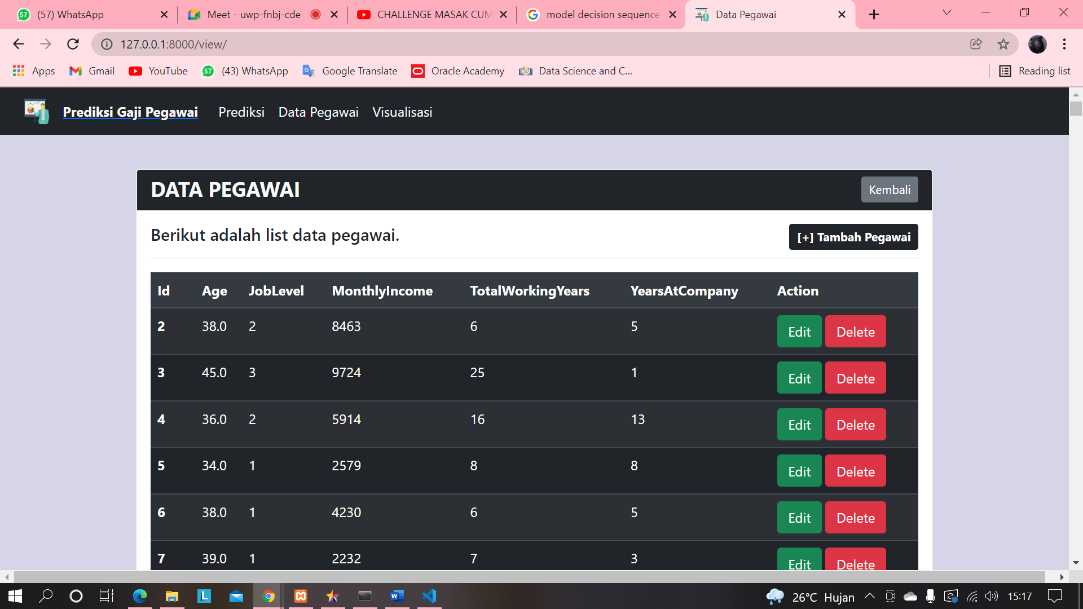
### Antarmuka Halaman Hasil Prediksi



Gambar 3. 34 Halaman Hasil Prediksi Sistem

Gambar diatas merupakan implementasi dari halaman hasil prediksi dari sistem yang dirancang. Pada halaman ini, akan ditampilkan hasil prediksi gaji dari user yang sebelumnya telah menginputkan angka tahun pada form prediksi.

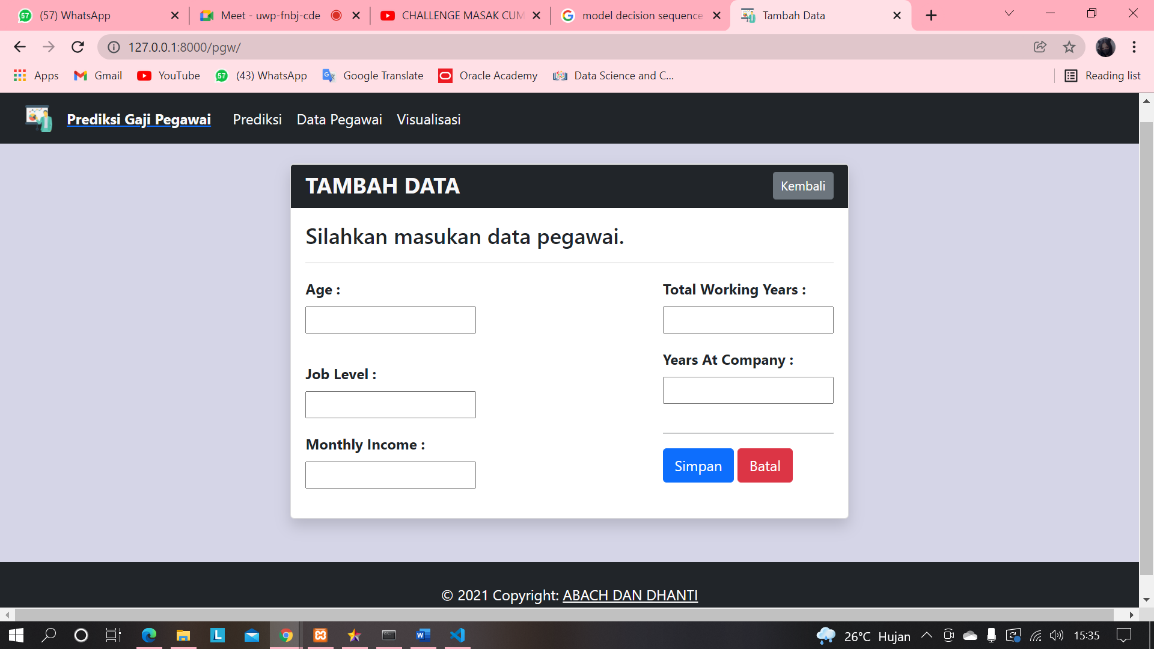
### Antarmuka Halaman Data Pegawai



Gambar 3. 35 Halaman Data Pegawai Sistem

Gambar diatas merupakan implementasi dari halaman data pegawai dari sistem yang dirancang. Pada halaman ini, akan ditampilkan data dalam bentuk tabel yang terdiri dari atribut Id, Age, JobLevel, MonthlyIncome, dan Action. *Record* yang ditampilkan terdiri dari 439 *record*. Terdapat *button* Tambah Pegawai yang berfungsi untuk menambah data pegawai, *button* Edit berfungsi untuk mengubah suatu *record* yang dipilih, dan *button* Delete untuk menghapus suatu *record* yang dipilih.

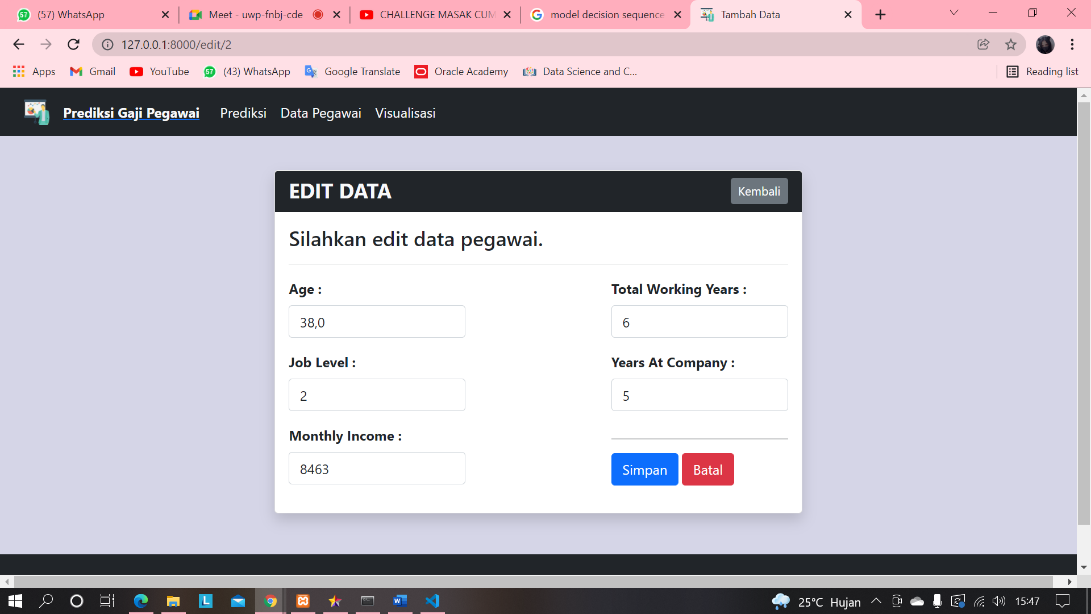
### Antarmuka Halaman Tambah Data Pegawai



Gambar 3. 36 Halaman Tambah Data Pegawai Sistem

Gambar diatas merupakan implementasi dari halaman tambah data pegawai dari sistem yang dirancang. Pada halaman ini, akan ditampilkan form tambah data untuk menambah data pegawai. *User* dapat menginputkan Age, Job Level, Monthly Income, Total Working Years, dan Years At Company pada form. Lalu *user* dapat menekan *button* Simpan untuk menyimpan data yang diinputkan. *User* dapat menekan *button* Batal untuk me-*reset* data yang diinputkan pada form.

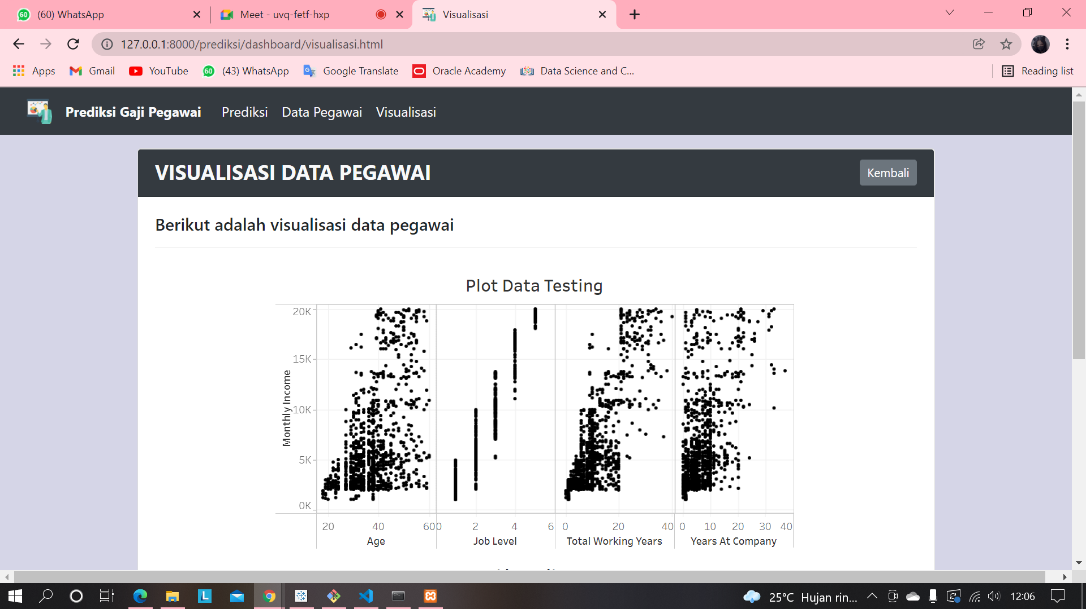
### Antarmuka Halaman Edit Data Pegawai



Gambar 3. 37 Halaman Edit Data Pegawai Sistem

Gambar diatas merupakan implementasi dari halaman edit data pegawai dari sistem yang dirancang. Pada halaman ini, akan ditampilkan form edit data untuk mengubah data pegawai. *User* dapat menginputkan Age, Job Level, Monthly Income, Total Working Years, dan Years At Company pada form. Lalu *user* dapat menekan *button* Simpan untuk menyimpan data yang telah diubah. *User* dapat menekan *button* Batal untuk me-*reset* data yang diinputkan pada form.

### Antarmuka Halaman Visualisasi



Gambar 3. 38 Halaman Visualisasi Sistem

Gambar diatas merupakan implementasi dari halaman visualisasi dari sistem yang dirancang. Pada halaman ini, akan ditampilkan bentuk visualisasi prediksi antara Monthly Income dengan Age, Job Level, Total Working Years, Years At Company. Visualisasi terdiri dari Plot Data Testing, dan Linearitas

## Analisis Kebutuhan

### Kebutuhan Fungsional (*Functional Requirements*)

Analisis kebutuhan fungsional merupakan suatu kebutuhan yang berhubungan dengan berbagai kebutuhan sistem yang akan dirancang. Dimana kebutuhan ini menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sistem, adapun kebutuhan fungsional yang akan dibuat yaitu terdiri dari 3 (tiga) proses sesuai dengan urutan sebagai berikut:

1. *Login* admin (user masuk ke aplikasi menggunakan username dan password);
2. Prediksi gaji pegawai yang dilakukan oleh Admin.
3. Melakukan *crate*, *read*, *update* dan *delete* pada data pegawai.

### Kebutuhan Non-Fungsional (Non-Functional Requirements)

Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan non-fungsional ini melibatkan analisis perangkat keras/*hardware*, analisis perangkat lunak/*software*, serta analisis pengguna/*user*. Adapun kebutuhan non-fungsional yang didapatkan adalah sebagai berikut :

1. Perangkat Keras (Hardware) Yang Digunakan

Tabel 3. 9 Perangkat Keras Yang Digunakan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** | **Keterangan** |
| 1 | *Hardisk* | 200 GB | Media untuk menyimpan data  aplikasi yang dibuat |
| 2 | *Memory* | 4 GB | *Memory System* yang digunakan |
| 3 | *Processor* | *AMD A4-9120 RADEON R3, 4 COMPUTE CORES 2C+2G*  *2.20 GHz* | Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer |
| 4 | Infrastruktur jaringan | *PC* | Mengolah, menginput serta menghasilkan *output* data ataupun informasi yang sesuai dengan keinginan pengguna (*user*) |

1. Perangkat Lunak (Software) Yang Digunakan

Tabel 3. 10 Perangkat Lunak Yang Digunakan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Tools / Software** | **Fungsi** | **Keterangan** |
| 1. | *Windows 10* | Sistem Operasi | Sistem Operasi yang  digunakan |
| 2. | *XAMPP 3.2.4* | Web server | Membuka web server |
| 3. | *Python* | Bahasa Pemograman | Bahasa pemograman yang  digunakan |
| 4. | *Lucidchart* | *Software* Pendukung | Media dalam pembuatan  flowmap |
| 5. | *Visual Studio Code* | *Software* Pendukung | Media penulisan *coding* |
| 6. | *Jupyter Notebook* | *Software* Pendukung | Media penulisan *coding* |
| 7. | *PDF, Microsoft Office*  *Word* | *Document* | Media untuk membuat  laporan |
| 8. | *Google Crome* | *Browser* | Media untuk mencari  informasi |

1. Pengguna (User)

Aplikasi yang akan dirancang ini digunakan dalam lingkup bisnis sebuah perusahaan dibagian pengelolaan data gaji pegawai. Aplikasi ini melibatkan Admin sebagai pengelola data gaji pegawai pada perusahaan tersebut.