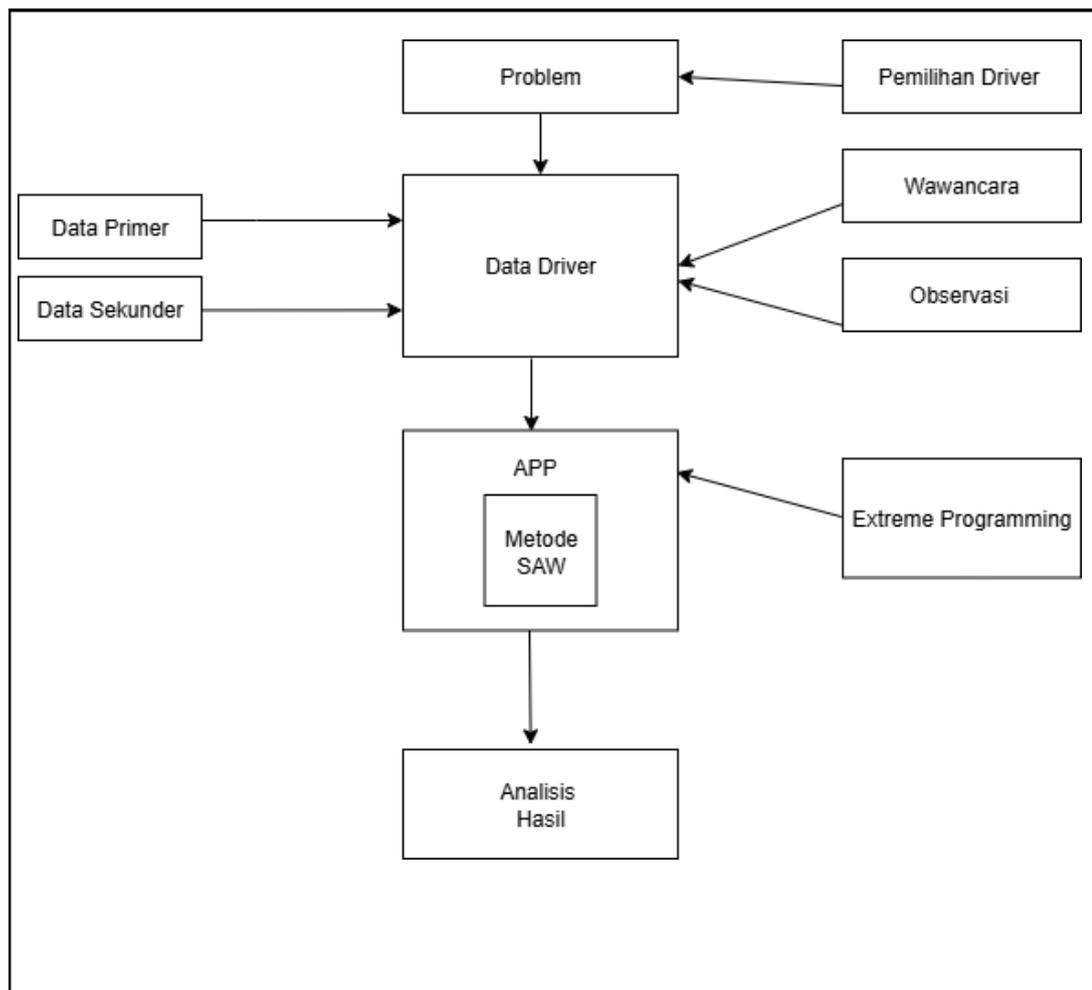


## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Tahapan Penelitian

Berikut adalah rancangan penelitian dari pembangunan Aplikasi Pemilihan *Driver* Terbaik pengangkutan batubara di PT. Kumala Bahtera Utama dapat dilihat pada gambar 3.1.



**Gambar 3.1** Tahapan Penelitian

Tahapan-tahapan rancangan penelitian adalah sebagai berikut :

1. Identifikasi Masalah

Tahap awal dimulai dengan mengidentifikasi permasalahan yang terjadi di lapangan. Selama ini, pemilihan *driver* terbaik dilakukan secara rekomendasi

langsung oleh pimpinan, tanpa sistem pendukung yang terstruktur. Hal ini menyebabkan proses penilaian bersifat subjektif dan kurang akurat.

## 2. Pengumpulan Data

Pengumpulan data dilakukan untuk mendukung proses penilaian *driver* secara objektif. Data yang digunakan terdiri dari data primer dan data sekunder. Data primer, yaitu data identitas dan informasi dasar masing-masing *driver*. Data sekunder, berupa data yang telah tersedia di perusahaan, meliputi, data absensi, *ritase*, kecelakaan kerja dan surat peringatan

## 3. Pengolahan Data dengan Aplikasi

Data yang telah dikumpulkan kemudian diolah menggunakan *metode Simple Additive Weighting* (SAW). Metode ini digunakan untuk menghitung nilai total dari setiap *driver* berdasarkan bobot kriteria yang telah ditentukan, sehingga menghasilkan peringkat *driver* terbaik secara objektif.

## 4. Pengembangan Sistem

Aplikasi dikembangkan dengan pendekatan *Extreme Programming*, yang memungkinkan pembuatan sistem lebih cepat, *fleksibel*, dan sesuai dengan kebutuhan pengguna.

## 5. Analisis Hasil

Tahapan akhir adalah menganalisis hasil dari sistem. Peringkat *driver* yang dihasilkan dievaluasi untuk memastikan bahwa sistem telah memberikan keputusan yang akurat, adil, dan dapat digunakan sebagai dasar dalam pengambilan keputusan oleh pihak perusahaan.

### 3.2 Waktu dan Tempat Penelitian

Objek penelitian yang menjadi fokus dalam tugas akhir ini adalah PT. Kumala Bahtera Utama. Untuk menyelesaikan tugas akhir ini tepat waktu, penulis telah menyusun jadwal pelaksanaan yang mencakup periode dari bulan Februari 2025 hingga Mei 2025 untuk melakukan penelitian.

### 3.3 Metode Pengumpulan Data

Untuk mendukung penelitian ini diperlukan data awal, untuk mengumpulkan data tersebut dilakukan beberapa cara salah satunya yaitu :

#### 1. Wawancara

Wawancara dilakukan dengan responden 3 orang (Pimpinan *Departemen* Produksi, admin *Departemen* Produksi, dan sebagian *driver*). Data yang didapat dari hasil wawancara adalah data *driver*. Data tersebut merupakan data primer dan data sekunder.

#### 2. Observasi

Tahapan observasi merupakan pengamatan secara langsung dilakukan penulis guna mencari kelengkapan data untuk menyelesaikan laporan tugas akhir ini. Metode observasi ini merupakan metode yang cukup efektif dan efisien untuk mempelajari suatu sistem. Penulis melakukan pengamatan secara langsung di lapangan tempat *driver* dalam pengangkutan Batubara dan mengumpulkan data secara detail di PT. Kumala Bahtera Utama.

#### 3. Studi Pustaka

Penulis mendapatkan informasi dan data dari beberapa sumber referensi, seperti artikel ilmiah, serta dari buku-buku yang mendukung informasi sesuai dengan topik yang diangkat serta informasi terkait dengan sistem yang akan dibangun.

### 3.4 Metode Pengembangan Sistem dan Metode Pemecah Masalah

#### 3.4.1 Metode Pengembang Sistem *Extreme Programming*

Tahapan-Tahapan metode *Extreme Programming* diantaranya yaitu :

##### 1. *Planning* (Perencanaan)

Tahapan *Planning* atau perencanaan merupakan tahap awal dalam pengembangan sistem pada objek yang diteliti, yaitu PT. Kumala Bahtera Utama. Sejumlah tugas perencanaan diselesaikan pada tahap ini, termasuk identifikasi masalah dan analisis kebutuhan untuk menetapkan jadwal implementasi. Langkah awal dalam tahapan ini adalah melakukan pengumpulan data melalui wawancara, observasi, atau diskusi, yang memberikan gambaran yang jelas kepada pengguna mengenai *fitur* utama,

*fungsi*alitas, dan *output* yang diinginkan. Perencanaan juga mencakup pemetaan alur proses bisnis yang ada, sehingga sistem yang dibangun nantinya benar-benar relevan dengan kebutuhan operasional perusahaan. Selain itu, metode pemecahan masalah juga ditetapkan pada tahapan ini, yaitu menggunakan metode *Simple Additive Weighting* (SAW).

#### 2. *Desain* (Perancangan)

Pada tahapan ini dilakukan proses perancangan yang meliputi pemodelan sistem, pemodelan arsitektur, dan pemodelan basis data. Pemodelan sistem dan arsitektur menggunakan alat bantu visual seperti *Data Flow Diagram* (DFD), *Block Chart*, *Entity Relationship Diagram* (ERD), dan *Flowchart*. Sedangkan untuk pemodelan basis data, digunakan aplikasi *PhpMyAdmin* sebagai alat bantu dalam merancang struktur basis data secara terstruktur.

#### 3. *Coding* (Pengkodean)

Pada tahapan ini, hasil pemodelan diterapkan ke dalam antarmuka pengguna (*user interface*) dengan menggunakan bahasa pemrograman PHP. Selain itu, sistem manajemen basis data yang digunakan untuk menyimpan dan mengelola data adalah *MySQL*.

#### 4. *Testing* (Pengujian)

Setelah tahap pengkodean selesai, dilakukan pengujian sistem untuk mengidentifikasi dan memperbaiki kesalahan yang muncul selama aplikasi dijalankan, sekaligus memastikan sistem telah memenuhi kebutuhan pengguna. Jika terdapat modul yang belum sesuai dengan spesifikasi, perbaikan akan dilakukan hingga modul tersebut memenuhi persyaratan. Pengujian menggunakan metode *Black-box* untuk memverifikasi bahwa seluruh fungsi aplikasi berjalan sesuai dengan kebutuhan yang telah ditetapkan.

### 3.4.2 Metode Pemecah Masalah *Simple Additive Weigting* (SAW)

Langkah-langkah metode SAW (Noviana *et al.*, 2024)

#### 1. Menentukan alternatif, dengan kode $A_i$ .

Alternatif merupakan objek atau pilihan yang akan dievaluasi dan dibandingkan dalam proses pengambilan keputusan. Pada penelitian ini, alternatif yang dianalisis adalah sepuluh kandidat *driver*, yaitu Sulaiman, Suparman, Asmedi, Isharyadi, Safril Hadi, Dedy Aprinaldi, Setiawan Mahendra, Hardianto, Tria Prastia, dan Bobby. Setiap kandidat ini akan dinilai berdasarkan beberapa kriteria yang telah ditentukan.

2. Menentukan kriteria yang dijadikan acuan perhitungan dengan kode *C<sub>i</sub>*.

Dalam penelitian ini, digunakan empat kriteria penilaian dalam pemilihan *driver*, yaitu produktivitas dan absensi sebagai atribut benefit, serta kecelakaan kerja dan surat peringatan sebagai atribut cost. Penetapan jenis atribut tersebut sangat penting dalam metode SAW untuk melakukan normalisasi data sehingga evaluasi dapat dilakukan secara objektif dan akurat.

3. Menentukan bobot sesuai dengan tingkat kepentingan.

Bobot kriteria menunjukkan tingkat kepentingan relatif dari masing-masing kriteria dalam proses pengambilan keputusan. Bobot ini harus mencerminkan prioritas dari setiap aspek yang dinilai. Dalam penelitian ini, bobot yang digunakan adalah Produktivitas (0,4), Absensi (0,2), Kecelakaan Kerja (0,2), dan Surat Peringatan (0,2). Penetapan bobot dilakukan berdasarkan analisis kebutuhan dan tujuan pengambilan keputusan.

4. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.

Setiap alternatif diberikan nilai atau rating pada setiap kriteria dengan menggunakan skala tertentu. Skala ini biasanya dalam rentang 1 hingga 3. Penilaian ini disesuaikan dengan kondisi dan kategori masing-masing kriteria untuk memudahkan perhitungan lebih lanjut. Skala benefit semakin tinggi penilainya maka semakin bagus sedangkan untuk skala cost semakin rendah penilainya semakin bagus.

5. Mempresentasikan rating kecocokan menjadi matriks keputusan (X).

Nilai-nilai rating yang diperoleh dari tahap sebelumnya kemudian disusun dalam sebuah matriks keputusan, di mana baris mewakili alternatif dan kolom mewakili kriteria. Matriks ini menjadi basis data utama untuk proses

analisis dan perhitungan dalam metode SAW. Untuk menentukan ranting kecocokan matriks keputusan menggunakan rumus berikut :

$$X = \begin{bmatrix} X_{11} & X_{12} & X_j \\ \dot{X}_{i1} & \dot{X}_{i2} & \dot{X}_j \end{bmatrix} \dots\dots\dots(1)$$

6. Melakukan normalisasi matriks (x) berdasarkan persamaan yang disesuaikan dengan jenis atribut dan membuat nilai rating kinerja ternormalisasi ( $r_{ij}$ ). Untuk kriteria *benefit*, nilai normalisasi dihitung dengan membagi nilai tiap alternatif dengan nilai maksimum pada kriteria tersebut. Untuk kriteria *cost*, nilai normalisasi dihitung dengan membagi nilai minimum pada kriteria tersebut dengan nilai tiap alternatif. Untuk melakukan normalisasi matriks menggunakan rumus berikut :

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\max X_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min X_{ij}}{X_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases} \dots\dots (2)$$

Keterangan :

- rij : Nilai rating kinerja ternormalisasi
- Xij : Nilai atribut yang dimiliki dari setiap kriteria
- Max Xij : Nilai terbesar dari setiap kriteria
- Min;Xij : Nilai terkecil dari setiap kriteria
- Benefit : Jika nilai terbesar adalah terbaik
- Cost : Jika nilai terkecil adalah terbaik

7. Hasil dari perhitungan rating kinerja ternormalisasi  $r_{ij}$  dibuat menjadi matriks ternormalisasi (R) dengan menggunakan rumus berikut :

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{1j} \\ \dot{r}_{i1} & \dot{r}_{i2} & \dot{r}_j \end{bmatrix} \dots\dots\dots(3)$$

8. Melakukan Perangkingan ( $V_i$ ) Nilai ( $V_i$ ) diperoleh dari perhitungan penjumlahan dan perkalian elemen baris matrik ternormalisasi (R) dengan bobot masing-masing kriteria (W) dengan menggunakan rumus berikut :



$$V_i = \sum_{j=1}^n W_j r_{ij} \dots \dots \dots (4)$$

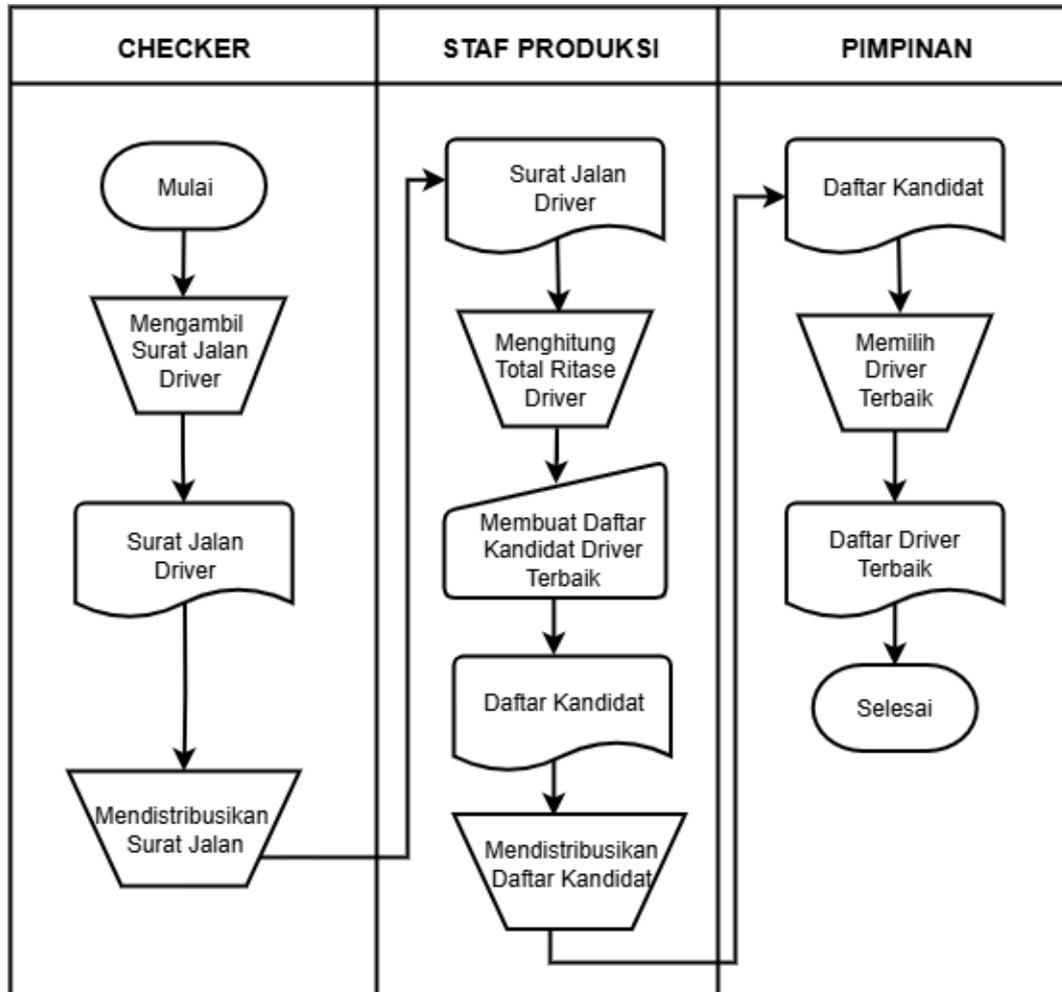
Hasil akhir perhitungan nilai  $V_i$  lebih besar dari nilai lain dapat diidentifikasi bahwa alternatif  $A_i$  merupakan alternatif yang direkomendasikan.

### 3.5 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem menjelaskan mengenai sistem yang sedang berjalan saat ini di tempat penelitian, kemudian menjelaskan juga mengenai sistem yang akan dibangun, serta spesifikasi kebutuhan dalam membangun sistem, baik itu *hardware* atau *software*.

### 3.5.1 Analisis Sistem Berjalan

Berikut adalah analisis sistem yang sedang berjalan di PT. Kumala Bhatara Utama dapat di lihat pada gambar 3.2.

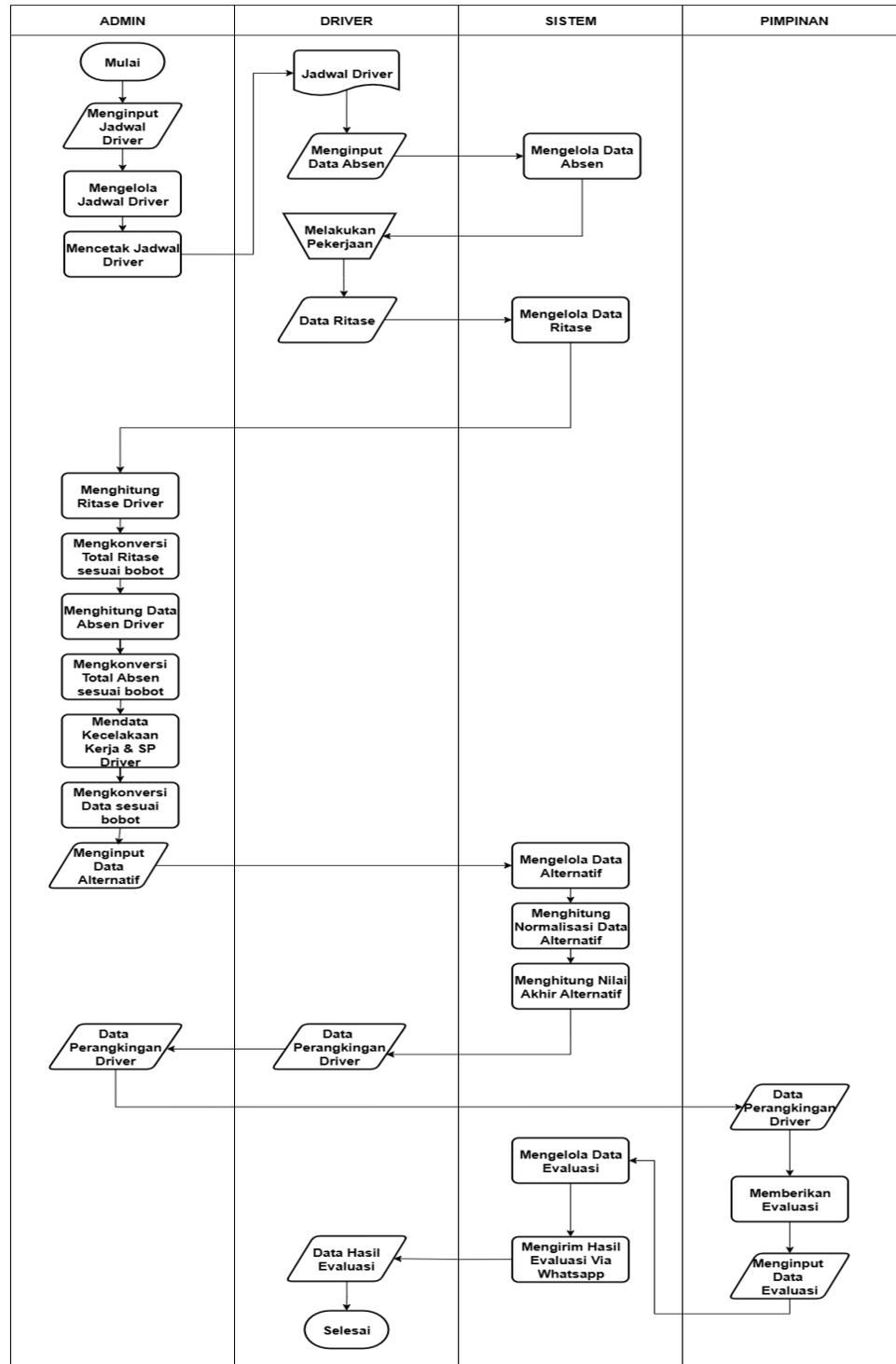


**Gambar 3. 2** Analisis Sistem Berjalan

Proses operasional dalam pengangkutan batubara yang terjadi saat ini masih menghadapi sejumlah permasalahan akibat belum didukung oleh sistem informasi yang terintegrasi. Proses yang tidak terintegrasi menyebabkan komunikasi dan koordinasi antar pihak, seperti *checker*, *staf* produksi, dan pimpinan, menjadi kurang efektif. tidak adanya sistem otomatisasi dalam pengolahan data *ritase* dan pemilihan kandidat *driver* terbaik berisiko menimbulkan ketidaktepatan dalam pengambilan keputusan. Distribusi laporan dan daftar kandidat yang masih dilakukan secara manual juga menimbulkan potensi keterlambatan dan risiko kehilangan data penting.

### 3.5.2 Analisis Sistem Usulan

Berikut analisis sistem yang sedang diusulkan di PT. Kumala Bahtera Utama dapat di lihat pada gambar 3.3.



Gambar 3.3 Analisis Sistem Berjalan

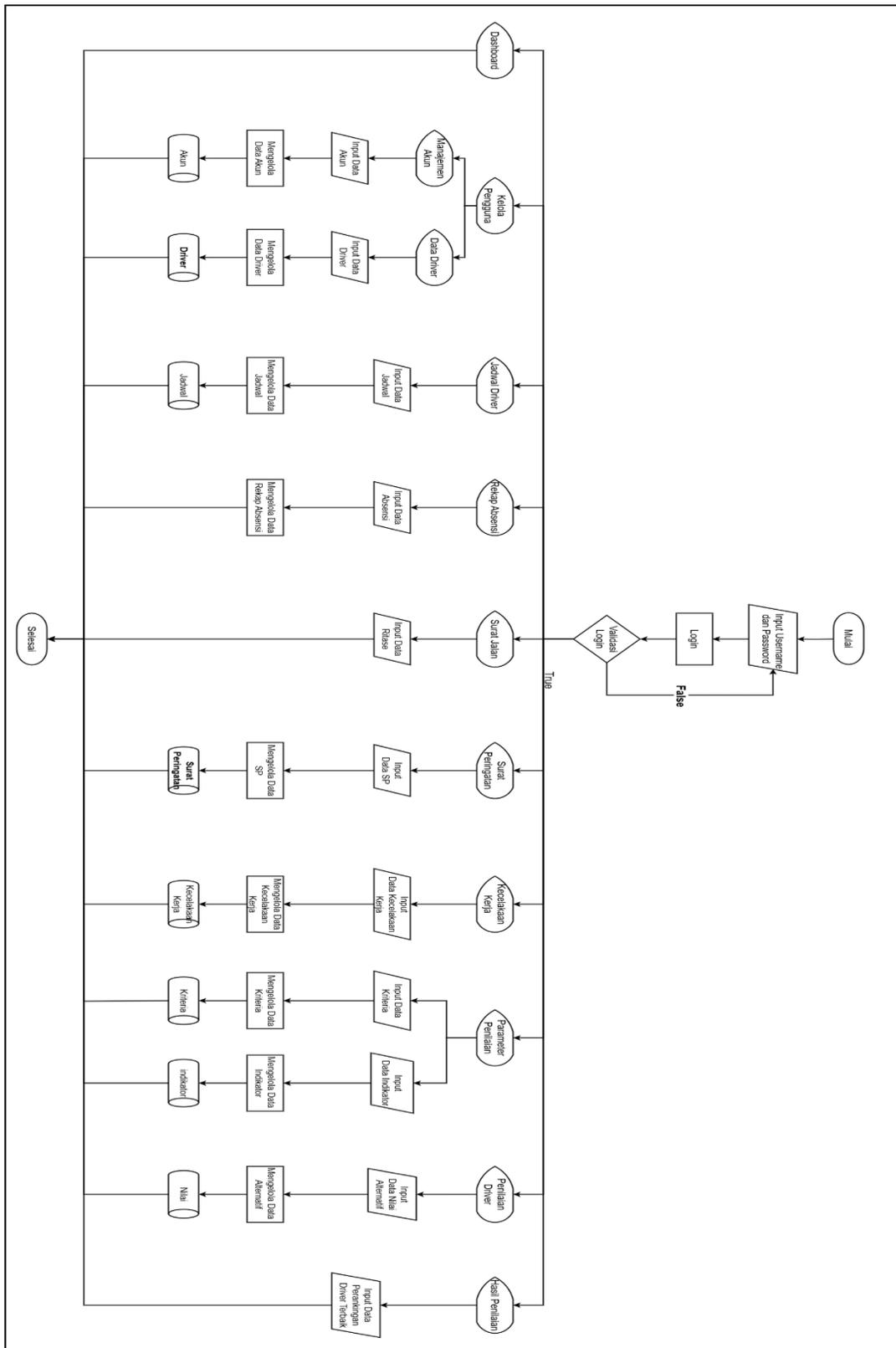


---

### 3.5.3 *Flowchart* Sistem yang di Usulkan

a. *Flowchart* Admin

*Flowchart* Admin menggambarkan alur proses yang dilakukan oleh admin dalam sistem pemilihan *driver* terbaik. Diagram ini menunjukkan langkah-langkah dari proses login hingga akses ke berbagai fitur yang tersedia dalam sistem yang dapat dilihat pada gambar 3.4.



Gambar 3. 4 Flowchart Admin



---

**Event List :**

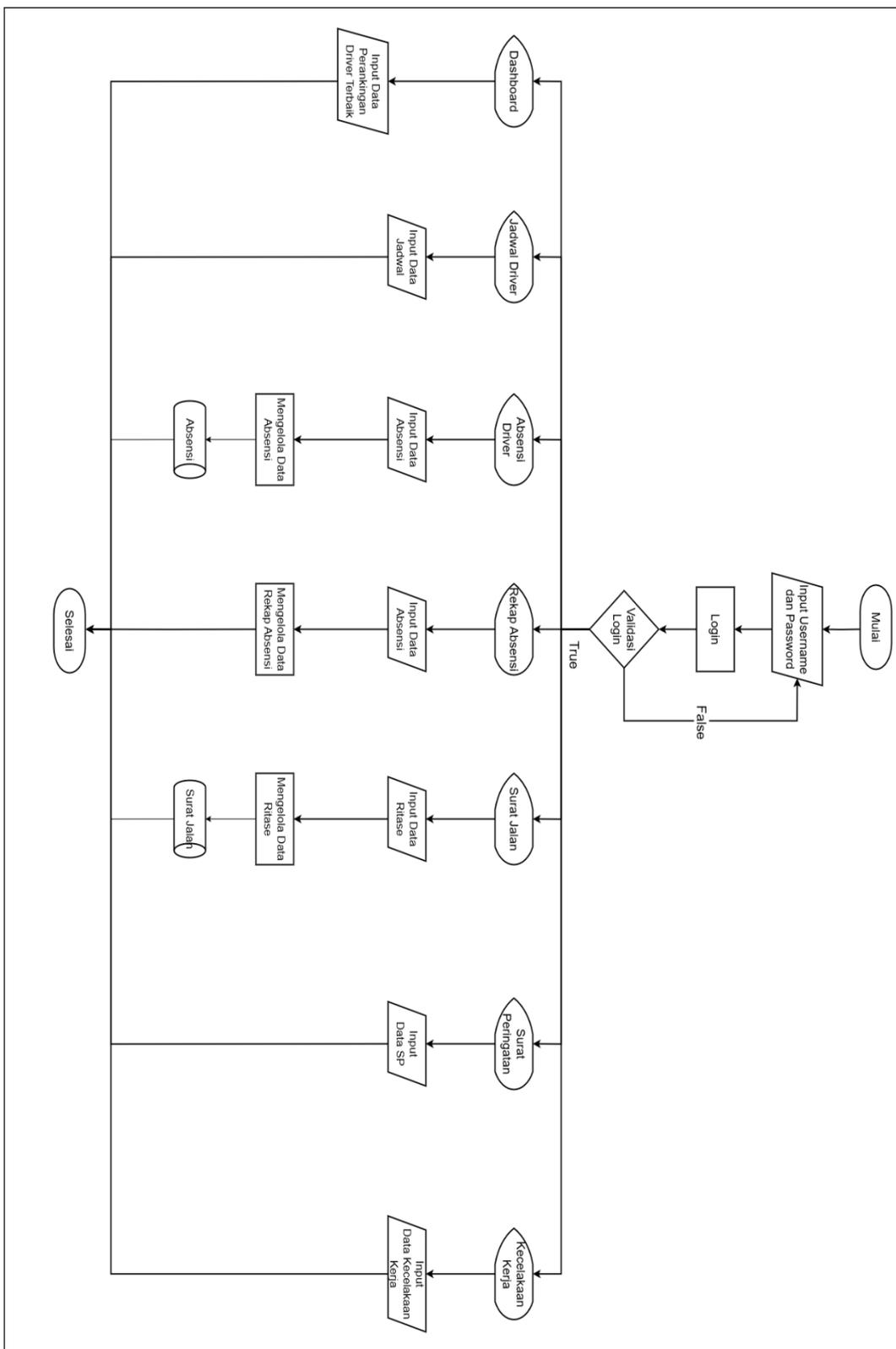
1. Admin melakukan *login* pertama dengan menginput *username*, dan *password*.
2. Aplikasi mengecek apakah data *login* yang di input benar atau tidak, apabila data benar maka admin akan masuk ke menu utama dan apabila data salah maka admin kembali ke halaman *login*.
3. Setelah *login* berhasil maka akan tampil halaman awal aplikasi yaitu halaman *dashboard*. Halaman *dashboard* akan menampilkan sub menu pada navigasi antara lain Kelola Pengguna, Jadwal *Driver*, Rekap Absensi, Surat Jalan, Surat Peringatan, Kecelakaan Kerja, Parameter Penilaian, Penilaian *Driver*, dan Hasil Penilaian.
4. Halaman Kelola Pengguna terdiri dari dua submenu, yaitu Manajemen Akun dan Data *Driver*. Halaman Manajemen Akun, Admin dapat menambahkan, mengedit, dan menghapus data akun pengguna. Halaman Data *Driver*, admin dapat menginput data *driver*, mengedit informasi *driver* dan menghapus data *driver*.
5. Halaman jadwal *driver* Admin dapat menginput dan mengelola data jadwal kerja *driver*, seperti tanggal dan *shift* kerja dan Admin dapat memperbarui jadwal dan menghapus jadwal kerja *driver*.
6. Halaman Rekap Absensi pada admin hanya dapat melihat informasi absensi *driver* berdasarkan periode yang dipilih
7. Halaman Surat Jalan pada admin dapat melihat informasi *ritase driver* yang mencakup jumlah ritase, tanggal, dan keterangan pengantaran.
8. Halaman Surat Peringatan pada admin dapat menambahkan data surat peringatan, mengedit isi SP yang telah dibuat, serta menghapus surat peringatan.
9. Halaman Kecelakaan Kerja pada admin dapat menambahkan, mengedit dan menghapus data kecelakaan kerja.
10. Halaman Parameter Penilaian pada admin dapat menambahkan, mengedit dan menghapus kriteria dan bobot penilaian.



11. Halaman Penilaian *Driver* pada admin dapat melakukan cetak rekap kinerja driver, menghitung dan megupdate penilaian driver berdasarkan kriteria dan bobot penilaian yang sudah di tentuhkan.
12. Halaman Hasil Penilaian pada admin dapat melakukan cetak hasil penilian driver terbaik.
13. Selesai.

b. *Flowchart Driver*

*Flowchart* ini menggambarkan alur aktivitas *driver* dalam sistem, mulai dari *login* hingga mengakses menu seperti *dashboard*, jadwal kerja, absensi, rekap absensi, surat jalan, surat peringatan (SP), dan kecelakaan kerja. Diagram ini membantu memahami peran dan interaksi driver dalam sistem secara sederhana dan terstruktur yang dapat dilihat pada gambar 3.5.



Gambar 3.5 Flowchart Driver



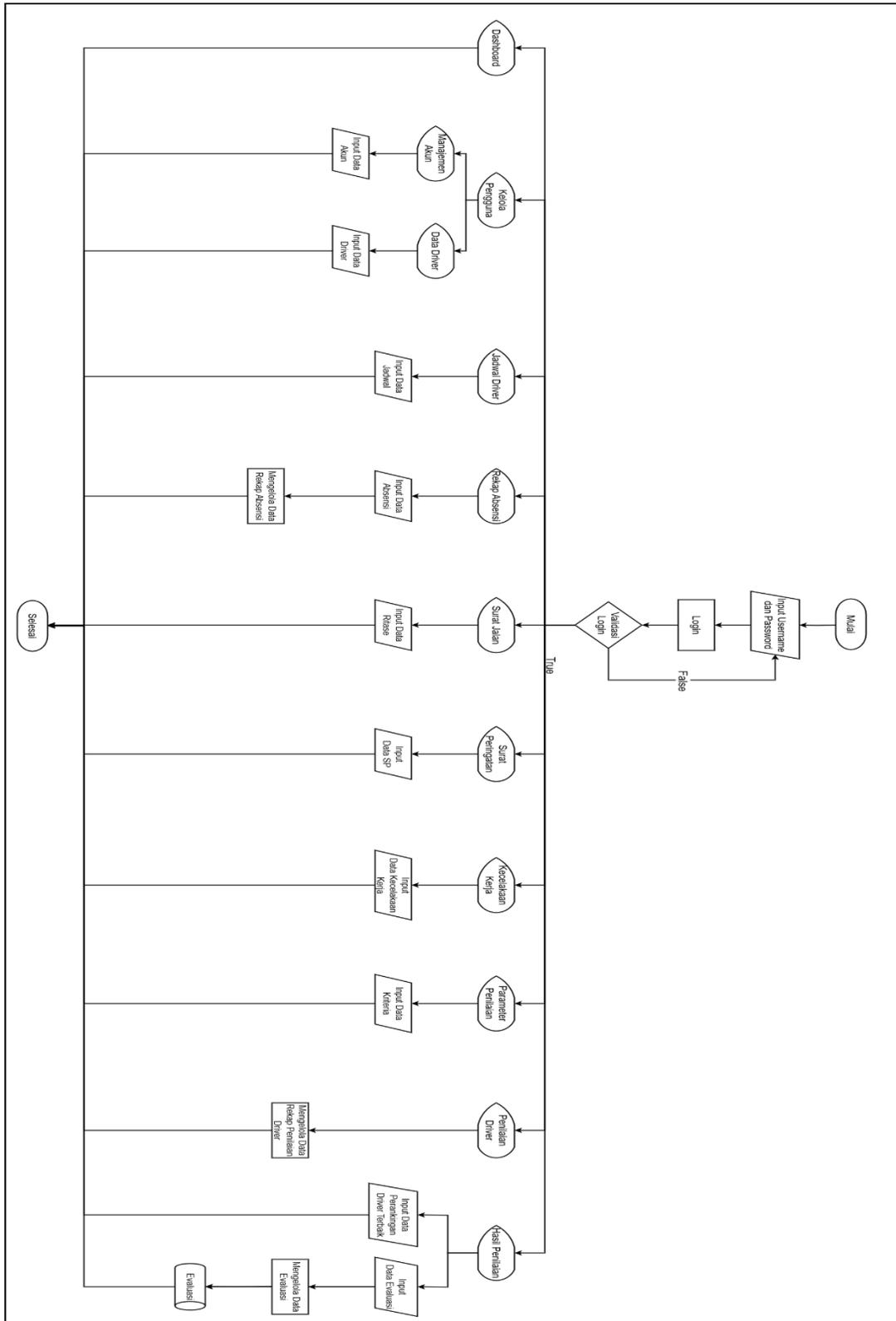
---

**Event List :**

1. *Driver* melakukan *login* pertama dengan menginput *username*, dan *password*.
2. Aplikasi mengecek apakah data *login* yang di input benar atau tidak, apabila data benar maka *driver* akan masuk ke menu utama dan apabila data salah maka *driver* kembali ke halaman *login*.
3. Setelah login berhasil maka akan tampil halaman awal aplikasi yaitu halaman *dashboard*.
4. Halaman *dashboard* akan menampilkan sub menu navigasi antara lain jadwal *driver*, absensi *driver*, rekap absensi, surat jalan, surat peringatan, kecelakaan kerja dan *logout*.
5. Halaman jadwal *driver* merupakan pembagian waktu kerja atau *shift driver*.
6. Halaman absensi merupakan halaman yang digunakan untuk merekap kehadiran *driver* Pada halaman ini, sistem menampilkan informasi jadwal kerja dan data kehadiran *driver* berdasarkan tanggal.
7. Halaman rekap absensi merupakan halaman yang menyajikan rangkuman kehadiran seluruh *driver* dalam periode tertentu.
8. Halaman surat jalan merupakan halaman yang berisi data surat jalan pengantaran batubara.
9. Halaman surat peringatan merupakan halaman daftar surat peringatan yang telah dikeluarkan kepada *driver* berdasarkan tingkat pelanggaran (SP1, SP2, SP3).
10. Halaman kecelakaan kerja merupakan halaman mencatat dan memantau seluruh insiden kecelakaan kerja yang terjadi di lingkungan operasional perusahaan.
11. Selesai

c. *Flowchart* Pimpinan

*Flowchart* pimpinan ini disusun untuk menggambarkan alur kerja yang dilakukan pimpinan dalam pemilihan *driver* terbaik yang dapat dilihat pada gambar 3.6.



Gambar 3. 6 Flowchart Pimpinan



---

**Event List :**

1. Pimpinan melakukan *login* pertama dengan menginput *username*, dan *password*.
2. Aplikasi mengecek apakah data *login* yang di input benar atau tidak, apabila data benar maka pimpinan akan masuk ke menu utama dan apabila data salah maka *driver* kembali ke halaman *login*.
3. Setelah login berhasil maka akan tampil halaman awal aplikasi yaitu halaman *dashboard*.
4. Setelah *login* berhasil maka akan tampil halaman awal aplikasi yaitu halaman *dashboard*. Halaman *dashboard* akan menampilkan sub menu pada navigasi antara lain Kelola Pengguna, Jadwal *Driver*, Rekap Absensi, Surat Jalan, Surat Peringatan, Kecelakaan Kerja, Parameter Penilaian, Penilaian *Driver*, dan Hasil Penilaian.
5. Halaman Kelola Pengguna terdiri dari dua submenu, yaitu *Manajemen Akun* dan *Data Driver*. Pada halaman ini, pimpinan hanya memiliki hak akses untuk melihat data pengguna yang telah terdaftar, tanpa dapat mengubah atau menambahkan informasi.
6. Halaman jadwal *driver* merupakan halaman yang berisi jadwal *shift* kerja *driver*.
7. Halaman Rekap Absensi merupakan halaman yang berisi rekap absensi *driver* berdasarkan periode yang dipilih.
8. Halaman surat jalan merupakan halaman yang berisi jumlah *ritase driver*.
9. Halaman surat peringatan merupakan halaman yang berisi data *driver* yang mengalami SP.
10. Halaman kecelakaan kerja merupakan halaman yang berisi data *driver* yang mengalami kecelakaan kerja.
11. Halaman parameter merupakan halaman yang berisi daftar kriteria atau indikator yang digunakan dalam proses evaluasi kinerja untuk *driver*.
12. Halaman Penilaian *Driver* merupakan halaman yang berisi evaluasi kinerja *driver* berdasarkan parameter yang telah ditetapkan.

13. Halaman Hasil Penilaian merupakan halaman yang berisi penilaian *driver* dan perangkian *driver*. Pada halaman ini pimpinan dapat memberikan evaluasi ke *driver* terbaik dan terburuk.

14. Selesai.

### 3.5.4 Spesifikasi Kebutuhan *Software/Hardware*

Untuk menghasilkan sebuah aplikasi yang dapat menghasilkan informasi yang cepat, tepat dan akurat dibutuhkan elemen-elemen dari sistem komputer sebagai alat bantu. Adapun alat tersebut terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*), yaitu:

#### 1. Perangkat Keras (*Hardware*)

Perangkat keras adalah komponen komputer yang dapat dilihat dan digunakan secara fisik. Perangkat keras yang digunakan meliputi: Alat untuk Pengembangan (*Development*) Perangkat yang digunakan selama proses pengembangan aplikasi:

##### a. Minimum Spesifikasi

- 1) CPU: Intel Core i3 (Gen 6) atau setara
- 2) RAM: 8 GB DDR4
- 3) Penyimpanan: HDD 256 GB

##### b. Perangkat yang Digunakan Saat Ini

- 1) Laptop *ASUS Vivobook Go E1404GAB\_E1404GA* x64-based PC  
*Processor Intel(R) Core(TM) i3-N305, 1800 Mhz, 8 Core(s), 8 Logical Processor(s)*
- 2) RAM DDR4 8 GB
- 3) GPU Nvidia GTX 1650
- 4) Penyimpanan SSD 512 GB

##### c. Alat untuk Penerapan (Running Aplikasi) Perangkat minimum yang diperlukan untuk menjalankan aplikasi setelah dikembangkan:

- 1) Minimum Spesifikasi
- 2) CPU: Intel Core i3 (Gen 6) atau setara
- 3) RAM: 4 GB DDR4



- 4) Penyimpanan: HDD 128 GB
- 5) Sistem Operasi: Windows 11

## 2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam pembuatan aplikasi:

- 1) *Browser (Edge)*
- 2) *Text editor (Visual Studio Code v1.97.0)*
- 3) XAMPP digunakan sebagai *server* penerjemah bahasa *pemrograman* yang digunakan untuk Pembangunan sistem.
- 4) *Version Control & Repository Hosting (GitHub)*
- 5) *Draw.io* untuk pembuatan diagram dan perancangan sistem.