

**OPTIMALISASI PENJADWALAN KEGIATAN KANTOR UNTUK  
TRANSPORTASI PEGAWAI  
MENGGUNAKAN ALGORITMA *PRIORITY QUEUE***



**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana  
Komputer pada Program Studi S1 Teknik Informatika Jurusan Teknologi  
Informasi  
Fakultas Teknik  
Universitas Tadulako

Disusun oleh :  
**MUNAWIR MUHSIN**  
**F 551 21 017**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS TADULAKO  
PALU  
2025**

HALAMAN PENGESAHAN

OPTIMALISASI PENJADWALAN KEGIATAN KANTOR UNTUK  
TRANSPORTASI PEGAWAI MENGGUNAKAN ALGORITMA  
PRIORITY QUEUE

Yang dipersiapkan dan disusun oleh:

MUNAWIR MUHSIN

F55121017

SKRIPSI

Telah dipertahankan didepan Majelis Penguji dan dinyatakan diterima sebagai salah satu  
persyaratan untuk memperoleh gelar Strata Satu (S1) Teknik Informatika

Pada tanggal 1 Desember 2025

Mengesahkan,



Dekan Fakultas Teknik  
Universitas Tadulako,

r. Audi Arham Adam, S.T., M.Eng., Ph.D.  
NIP. 19740323 199903 1 002

Ketua Jurusan Teknologi Informasi  
Fakultas Teknik Universitas Tadulako,



Dr. Anita Ahmad Kasim, S.Kom., M.Cs.  
NIP. 19790112 200501 2 002

## HALAMAN PERSETUJUAN

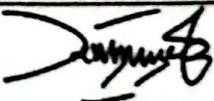
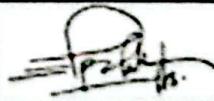
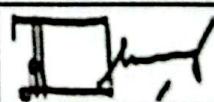
Pada Hari Senin, Tanggal 1 Bulan Desember Tahun 2025, Panitia Ujian Skripsi Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Tadulako, Berdasarkan SK No. 22919/UN28.1.31/DK/2025, Tanggal 1 Desember 2025, Menyatakan Menerima, Menyetujui Skripsi Yang Telah Di Pertanggungjawabkan Dihadapan Panitia Penguji Skripsi Oleh :

**MUNAWIR MUHSIN : F55121017**

Judul:

**"OPTIMALISASI PENJADWALAN KEGIATAN KANTOR UNTUK TRANSPORTASI PEGAWAI MENGGUNAKAN ALGORITMA PRIORITY QUEUE"**

Panitia Ujian Skripsi:

No.	Nama	Jabatan	TTD
1	Dr. Dassy Santi, S.Kom., MT.	Ketua Penguji	
2	Chairunnisa Lamasitudju, S.Kom., M.Pd	Sekretaris Penguji	
3	Ir. Syahrullah, S.Kom., M.Kom.	Anggota Penguji	
4	Dr. Deny Wiria Nugraha, S.T., M.Eng.	Pembimbing	

Palu, 10 Desember 2025



TAS-90CSYGXAYZXOWK1EWDPV

## HALAMAN PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Munawir Muhsin

Stambuk : F55121017

Judul : Optimalisasi Penjadwalan Kegiatan Kantor Untuk Transportasi Pegawai Menggunakan Algoritma Priority Queue.

Dengan ini saya menyatakan bahwa skripsi ini merupakan hasil karya saya sendiri. Seluruh data, informasi, dan temuan yang disajikan dalam penelitian ini diperoleh melalui proses pengamatan, analisis, dan pemikiran saya secara mandiri, yang dilaksanakan berdasarkan metode dan prosedur yang telah dijelaskan dalam laporan ini. Sepanjang pengetahuan saya, tidak terdapat karya atau pendapat orang lain yang saya gunakan tanpa mencantumkan sumbernya. Seluruh kutipan dan referensi dari pihak lain telah saya cantumkan secara lengkap dalam daftar pustaka. Saya sepenuhnya memahami pentingnya kejujuran akademik dan bertanggung jawab penuh atas isi dan hasil dari penelitian ini.

Palu, November 2025

Munawir Muhsin

## ABSTRAK

**Munawir**, Optimalisasi Penjadwalan Kegiatan Kantor Untuk Transportasi Pegawai Menggunakan Algoritma *Priority Queue*. (Studi Kasus: Kantor Perwakilan Bank Indonesia Provinsi Sulawesi Tengah) Dibimbing oleh Dr Deny Wiria Nugraha, S.T., M.Eng.

Penelitian ini bertujuan untuk mengoptimalkan penjadwalan kegiatan kantor dalam rangka meningkatkan efisiensi transportasi pegawai. Proses penjadwalan sebelumnya dilakukan secara manual sehingga sering menimbulkan benturan jadwal dan ketidakseimbangan beban kerja pada sopir. Untuk mengatasi permasalahan tersebut, dikembangkan sebuah sistem penjadwalan berbasis web yang menerapkan algoritma priority queue sebagai mekanisme utama dalam menentukan tingkat prioritas kegiatan. Algoritma ini memungkinkan pendistribusian jadwal dilakukan secara lebih adil, terstruktur, dan sesuai urgensi kegiatan. Metode pengembangan perangkat lunak yang digunakan adalah model prototype, yang terdiri atas enam tahap: *requirements gathering and analysis, quick design, build a prototype, initial user evaluation, refining prototype*, serta *implementation and maintenance*. Pemilihan metode ini memungkinkan adanya umpan balik langsung dari pengguna sehingga pengembangan sistem dapat disesuaikan dengan kebutuhan kantor. Hasil penelitian menunjukkan adanya peningkatan efektivitas penjadwalan, baik dari sisi akurasi maupun kecepatan proses. Pengujian sistem membuktikan bahwa algoritma *priority queue* memiliki performa yang lebih efisien dibandingkan algoritma FCFS, dengan peningkatan waktu eksekusi sebesar 11,02%. Dengan demikian, sistem ini mampu mendukung kelancaran operasional transportasi internal secara lebih optimal.

**Kata Kunci** : Penjadwalan Kegiatan, *Priority Queue*, Metode *Prototype*, Bank Indonesia

## ***ABSTRACT***

**Munawir, Optimization of Office Activity Scheduling for Employee Transportation Using the Priority Queue Algorithm (Case Study: Bank Indonesia Representative Office of Central Sulawesi Province). Supervised by Dr. Deny Wiria Nugraha, S.T., M.Eng.**

*This study aims to optimize office activity scheduling to improve the efficiency of employee transportation. The previous scheduling process was carried out manually, often resulting in schedule conflicts and uneven workload distribution among drivers. To address these issues, a web-based scheduling system was developed by implementing the priority queue algorithm as the main mechanism for determining the priority level of each activity. This algorithm enables a more equitable, structured, and urgency-based distribution of schedules. The software development method used in this study is the prototype model, which consists of six stages: requirements gathering and analysis, quick design, build a prototype, initial user evaluation, refining the prototype, and implementation and maintenance. This method was chosen because it allows continuous user feedback, ensuring that the system aligns with the actual needs of the office. The results show an improvement in scheduling effectiveness, both in terms of accuracy and processing speed. System testing demonstrates that the priority queue algorithm performs more efficiently than the FCFS algorithm, with an execution time improvement of 11.02%. Thus, the system effectively supports the smooth operation of internal transportation processes.*

**Keywords:** *Activity Scheduling, Priority Queue, Prototype Method, Bank Indonesia*

## KATA PENGANTAR

Segala puji syukur penulis panjatkan kehadiran Allah SWT. berkat rahmat dan hidayahnya sehingga penulis mampu menyelesaikan penulisan skripsi dengan judul “Optimalisasi Penjadwalan Kegiatan Kantor Untuk Transportasi Pegawai Menggunakan Algoritma *Priority Queue*”. Penulisan penelitian ini dimaksudkan untuk memenuhi salah satu persyaratan memperoleh gelar Sarjana Teknik pada Program Studi S1 Teknik Informatika, Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Tadulako.

Dalam proses penyusunan Skripsi ini, penulis banyak mendapatkan bimbingan dan petunjuk serta bantuan yang bermanfaat dari berbagai pihak oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Ir. Andi Arham Adam, S.T., M.Sc., Ph.D., ASEAN Eng selaku Dekan Fakultas Teknik Universitas Tadulako.
2. Ibu Dr. Ir. Yuli Asmi Rahman, ST., M.Eng, selaku Wakil Dekan Bidang Akademik Fakultas Teknik, Universitas Tadulako.
3. Ibu Dr. Ir. Anita Ahmad Kasim, S.Kom., M.CS. selaku Ketua Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Tadulako.
4. Bapak Yuri Yudhaswana Joefrie, S.T., M.T., Ph.D. selaku Koordinator Program Studi S1 Teknik Informatika, Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Tadulako.
5. Ibu Dr. Amriana, S.T., M.T. selaku Ketua KDK Rekayasa Sistem Lanjutan Program Studi S1 Teknik Informatika, Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Tadulako.
6. Bapak Dr. Deny Wiria Nugraha, S.T., M.Eng. Selaku Dosen pembimbing tugas akhir, yang telah memberikan arahan dan bimbingan dalam setiap tahap penyusunan proposal ini.
7. Ibu Dr. Dessy Santi, S.Kom., MT. Selaku Ketua Tim Penguji, Ibu Chairunnisa Lamasitudju, S.Kom., M.Pd. Selaku Sekretaris Tim Penguji, dan Bapak Ir. Syahrullah, S.Kom M.Kom. Selaku Anggota Tim Penguji yang telah memberikan masukan, saran, serta penilaian yang sangat berharga untuk kesempurnaan skripsi ini.
8. Seluruh dosen pengajar di Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Tadulako yang telah memberikan pendidikan dan pengetahuan yang sangat berarti kepada penulis.
9. Seluruh staf/laboran/teknisi di laboratorium Jurusan Teknologi Informasi yang telah berpartisipasi dan memberi dukungan dalam pembuatan proposal ini.
10. Seluruh staf akademik dan administrasi Jurusan Teknologi Informasi yang telah memberikan semangat dan bantuannya selama ini.

11. Kepada orang tua kandung tercinta dari penulis bapak Zaenuddin Mustamin dan ibu Inayah, serta orang tua angkat tercinta dari penulis bapak Drs. Muhsin Mustamin dan ibu Masna, yang selalu memberikan dukungan materi, semangat, kasih sayang dan doa yang tak henti-hentinya.
12. Keluarga besar penulis yang selalu memberikan dukungan, masukan dan selalu ada untuk penulis selama menempuh Pendidikan.
13. Seluruh rekan-rekan angkatan 2021 Jurusan Teknologi Informasi, yang telah menjadi keluarga kedua, memberikan semangat, inspirasi, dan kebersamaan yang tak ternilai sepanjang penulis menempuh pendidikan.
14. Teman-teman PMM *Outbound* untad UPNVY yang telah menghibur penulis yaitu Ayub Vigo, S.Kom, Muh. Sigit Rialdi, S.Kom, Amalia Gultom, S.Agr, Mauizzatil Hasanah M.H., S.E, Mutia, S.Kom, Ni'ma Tusyafa'ah, Imam Fauzi, Sahron Angelina Ihalauw, Wahyudi Mansyur, Aisyah Maharani Hatta.
15. Teman-teman nongki yotta! yang telah menemani dan menyemangati penulis dalam mengerjakan skripsi sampai akhir.

Penulis menyadari sepenuhnya bahwa dalam penulisan hasil ini masih terdapat banyak kekurangan. Sehingga penulis mengharapkan kritik dan saran yang membangun dari semua pihak. Akhirnya penulis berharap agar hasil ini dapat dimanfaatkan, baik bagi rekan-rekan mahasiswa maupun bagi masyarakat luas.

Palu, November 2025

Munawir Muhsin

## DAFTAR ISI

<b>HALAMAN PERSETUJUAN</b> .....	i
<b>HALAMAN PERNYATAAN</b> .....	ii
<b>ABSTRAK</b> .....	iii
<b>ABSTRACT</b> .....	iv
<b>KATA PENGANTAR</b> .....	v
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	ix
<b>DAFTAR TABEL</b> .....	x
<b>DAFTAR SIMBOL DAN ISTILAH</b> .....	xi
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian .....	3
1.6 Waktu Pelaksanaan.....	4
1.7 Sistematika Penulisan.....	4
<b>BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI</b> .....	6
2.1 Tinjauan Pustaka .....	6
2.2 Landasan Teori .....	12
2.2.1 Bank Indonesia .....	12
2.2.2 Penjadwalan .....	12
2.2.3 Metode <i>Prototype</i> .....	13
2.2.4 Algortima <i>Priority queue</i> .....	13
2.2.5 Metode <i>Analytic Hierarchy Process</i> (AHP).....	15
2.2.6 <i>Laravel</i> .....	17
2.3 Kerangka Pemikiran.....	17
<b>BAB III METODOLOGI PENELITIAN</b> .....	19
3.1 Tahapan dan <i>Diagram Alir</i> Penelitian .....	19
3.2 Bahan Penelitian .....	23
3.3 Instrumen Penelitian.....	23
3.4 Desain Penelitian.....	24
3.4.1 Jenis Penelitian.....	24

3.4.2 Tipe Penelitian.....	24
3.5 Teknik Pengumpulan Data .....	24
3.6 Metode Analisis Data .....	25
3.7 Objek Penelitian dan Lokasi Penelitian .....	25
3.8 Metode Pengembangan Sistem.....	25
3.9 Pengujian Sistem.....	27
3.10 Perancangan Sistem.....	28
3.10.1 Fitur Sistem .....	28
3.10.2 Desain Sistem.....	28
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>34</b>
4.1 Hasil Penelitian .....	34
4.1.1 Pengembangan Sistem.....	34
4.1.2 Algoritma <i>Priority Queue</i> .....	42
4.1.3 Implementasi Sistem .....	45
4.1.4 Pengujian Sistem.....	54
4.2 Pembahasan.....	56
<b>BAB V PENUTUP .....</b>	<b>60</b>
5.1 Kesimpulan .....	60
5.2 Saran .....	60
<b>DAFTAR PUSTAKA .....</b>	<b>62</b>
<b>DAFTAR RIWAYAT HIDUP .....</b>	<b>64</b>
<b>LAMPIRAN .....</b>	<b>65</b>

## DAFTAR GAMBAR

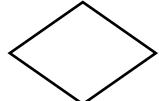
Gambar 2. 1 Flowchart Priority Queue .....	13
Gambar 2. 2 Kerangka Pemikiran .....	18
Gambar 3. 1 Alur Penelitian .....	19
Gambar 3. 2 Flowchart Sistem .....	21
Gambar 3. 3 Metode Prototype .....	25
Gambar 3. 4 Halaman Login .....	28
Gambar 3. 5 Halaman Register .....	28
Gambar 3. 6 Halaman Dashboard .....	29
Gambar 3. 7 Halaman Informasi Jadwal Admin .....	29
Gambar 3. 8 Halaman Informasi Jadwal User .....	29
Gambar 3. 9 Halaman Penjadwalan .....	30
Gambar 3. 10 Halaman Informasi Jadwal .....	30
Gambar 3. 11 Halaman Template Surat .....	31
Gambar 3. 12 Halaman Pembuatan Surat .....	31
Gambar 3. 13 Halaman Pengiriman Surat .....	32
Gambar 3. 14 Halaman Driver Admin .....	32
Gambar 3. 15 Halaman Driver User .....	33
Gambar 4. 1 DFD level 0 .....	34
Gambar 4. 2 DFD level 1 .....	35
Gambar 4. 3 DFD Level 2 Proses 1 .....	35
Gambar 4. 4 DFD Level 2 Proses 2 .....	36
Gambar 4. 5 DFD Level 2 Proses 3 .....	36
Gambar 4. 6 DFD Level 2 Proses 4 .....	36
Gambar 4. 7 DFD Level 2 Proses 5 .....	37
Gambar 4. 8 Use Case Diagram .....	38
Gambar 4. 9 Activity Diagram Admin .....	39
Gambar 4. 10 Activity Diagram Pegawai .....	40
Gambar 4. 11 Class Diagram .....	42
Gambar 4. 12 Halaman Login .....	48
Gambar 4. 13 Halaman Register .....	49
Gambar 4. 14 Halaman Dashboard .....	49
Gambar 4. 15 Halaman Informasi Jadwal Admin .....	50
Gambar 4. 16 Halaman Informasi Jadwal User .....	50
Gambar 4. 17 Halaman Penjadwalan .....	51
Gambar 4. 18 Halaman Informasi Jadwal .....	51
Gambar 4. 19 Halaman Template Surat .....	52
Gambar 4. 20 Halaman Pembuatan Surat .....	52
Gambar 4. 21 Halaman Pengiriman Surat .....	53
Gambar 4. 22 Halaman Driver Admin .....	53
Gambar 4. 23 Halaman Driver User .....	54
Gambar 4. 24 Database Hasil Pengujian Fungsional Bobot .....	58

## DAFTAR TABEL

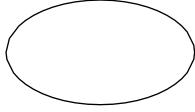
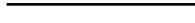
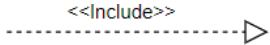
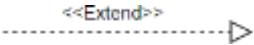
Tabel 1. 1 Waktu Pelaksanaan .....	4
Tabel 2. 1 Perbandingan Penelitian Terdahulu .....	9
Tabel 4. 1 Data Kegiatan BI Sulteng .....	42
Tabel 4. 2 Tabel Matriks Perbandingan Berpasangan .....	43
Tabel 4. 3 Nilai Desimal Matriks Perbandingan Berpasangan.....	43
Tabel 4. 4 Normalisasi Matriks Perbandingan Berpasangan .....	44
Tabel 4. 5 Hasil Bobot Prioritas AHP.....	44
Tabel 4. 6 Hasil perhitungan Kriteria .....	44
Tabel 4. 7 Tabel Skor Jabatan.....	44
Tabel 4. 8 Tabel Skor Tenggat Waktu .....	45
Tabel 4. 9 Tabel Skor Jenis Kegiatan.....	45
Tabel 4. 10 Tabel Reservation .....	46
Tabel 4. 11 Tabel Report .....	46
Tabel 4. 12 Tabel Surat.....	47
Tabel 4. 13 Tabel Driver.....	47
Tabel 4. 14 Tabel Unit.....	47
Tabel 4. 15 Tabel User.....	47
Tabel 4. 16 Tabel Activity .....	47
Tabel 4. 17 Tabel Jenis Activity .....	48
Tabel 4. 18 Tabel Report Activity.....	48
Tabel 4. 19 Pengujian Halaman Admin .....	54
Tabel 4. 20 Pengujian Halaman User .....	55
Tabel 4. 21 Pengujian Perbandingan Algortima .....	56
Tabel 4. 22 Tabel Pengujian Fungsional Bobot .....	57

## DAFTAR SIMBOL DAN ISTILAH

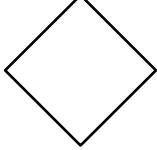
### SIMBOL *FLOWCHART*

Gambar	Nama	Keterangan
	Mulai atau selesai (terminator)	Menunjukkan permulaan (start) atau akhir (stop) dari suatu proses
	Proses	Digunakan untuk menunjukkan kegiatan yang dilakukan oleh komputer
	Input Ouput	Menunjukkan proses <i>input-output</i> yang terjadi tanpa bergantung dari jenis peralatannya
	Seleksi	Simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada. Simbol ini biasanya ditemui pada flowchart program
	Arus Data	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain (connecting line). Simbol ini juga berfungsi untuk menunjukkan garis alir dari proses

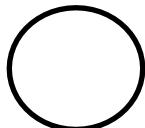
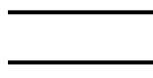
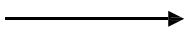
## SIMBOL USE CASE DIAGRAM

Simbol	Nama	Arti Simbol
	Aktor	Menunjukkan pihak yang berinteraksi dengan sistem dan memberikan informasi sekaligus menerima informasi dari sistem.
	Use Case	Menunjukkan proses yang dibangun dalam sistem tersebut.
	Asosiasi	Menunjukkan abstraksi penghubung antara aktor dengan use case.
	<i>Include</i>	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya.
	<i>Extend</i>	Menunjukkan bahwa suatu use case merupakan tambahan fungsionalitas dari use case lainnya jika suatu kondisi terpenuhi.

### **SIMBOL ACTIVITY DIAGRAM**

<b>Simbol</b>	<b>Nama</b>	<b>Arti Simbol</b>
	Status Awal	Menunjukkan bagaimana sistem diawali.
	Status Akhir	Menunjukkan bagaimana sistem diakhiri
	Aktivitas	Menunjukkan eksekusi dari suatu sistem.
	Percabangan	Menunjukkan simbol untuk pengambilan keputusan bagaimana alur dalam <i>activity diagram</i> berjalan selanjutnya berdasarkan kriteria atau pertanyaan tertentu.

### **SIMBOL DATA FLOW DIAGRAM**

<b>Simbol</b>	<b>Nama</b>	<b>Arti Simbol</b>
	<i>External Entity</i>	Menunjukkan permulaan ( <i>start</i> ) atau akhir ( <i>stop</i> ) dari suatu proses
	<i>Proses</i>	Menunjukkan proses yang dilakukan sebuah mesin untuk mengubah input menjadi <i>output</i> menjadi <i>format</i> yang berbeda.
	<i>Data Base</i>	Digunakan sebagai penyimpanan data yang bersifat sementara maupun permanen.
	Alur Data	Menunjukkan arus data yang mengalir di antara suatu proses, terminator, dan data store. Hal ini digambarkan dengan simbol tanda panah

## SIMBOL CLASS DIAGRAM

Gambar	Nama	Keterangan
	Komponen atas	Komponen ini berisikan nama <i>class</i> . Setiap <i>class</i> pasti memiliki nama yang berbeda-beda.
	Komponen tengah	Komponen ini berisikan atribut dari <i>class</i> , komponen ini digunakan untuk menjelaskan kualitas dari suatu kelas. Atribut ini dapat menjelaskan dapat ditulis lebih detail, dengan cara memasukan tipe nilai.
	Komponen bawah	Komponen ini menyertakan operasi yang ditampilkan dalam bentuk daftar. Operasi ini dapat menggambarkan bagaimana suatu <i>class</i> dapat berinteraksi dengan data.
	Asosiasi	Asosiasi dapat diartikan sebagai hubungan antara dua <i>class</i> yang bersifat statis. Biasanya asosiasi menjelaskan <i>class</i> yang memiliki atribut tambahan seperti <i>class</i> lain.
	Pewarisan	Pewarisan atau inheritance dapat disebut juga generalization dalam class diagram adalah suatu kemampuan untuk mewarisi seluruh atribut dan metode dari class asalnya ( <i>superclass</i> ) ke class lain ( <i>subclass</i> ).
	Agregasi	Agregasi adalah hubungan antara dua class di mana salah satu class merupakan bagian dari class lain, tetapi dua class ini dapat berdiri masing-masing.

Singkatan		
LDP	=	Lembar Disposisi Pejabat
UMI	=	Unit Manajemen Intern
DFD	=	<i>Data Flow Diagram</i>

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Perubahan lingkungan dan ekonomi global yang semakin kompleks juga dinamis memaksa semua pihak beradaptasi dengan cepat. Inovasi dan efisiensi operasional menjadi tuntutan utama bagi setiap organisasi, termasuk Bank Indonesia, untuk dapat beradaptasi dengan perubahan lingkungan global yang kompleks. Sebagai bank sentral terdepan, Bank Indonesia memiliki visi menjadi bank sentral digital terdepan dengan tata kelola kuat yang berkontribusi nyata terhadap perekonomian nasional dan terbaik di antara negara *emerging markets* untuk Indonesia Maju. (Bank Indonesia Institute, 2020) Kondisi ini menuntut digitalisasi pada seluruh aspek pekerjaan yang ada di Bank Indonesia, termasuk manajemen logistik dan transportasi, yang masih banyak dilakukan secara manual.

Penjadwalan transportasi yang efisien akan berdampak pada kelancaran mobilitas, penghematan waktu dan peningkatan produktivitas. Penjadwalan dapat diartikan sebagai pengalokasian sejumlah sumber daya untuk melakukan sejumlah tugas atau operasi dalam jangka waktu tertentu dan merupakan proses pengambilan keputusan yang peranannya sangat penting (Zakharia et al., 2021). Dalam praktiknya, suatu penjadwalan transportasi untuk kegiatan seringkali akan menghadapi berbagai kendala, antara lain perubahan mendadak pada prioritas kegiatan, keterbatasan jumlah kendaraan dan sopir, serta kesalahan *administrasi*, yang dapat menjadi penyebab utama kegagalan pelaksanaan jadwal.

Transportasi merupakan salah satu peran penting dalam mendukung kelancaran setiap kegiatan, terutama dalam lingkungan kerja yang menuntut mobilitas tinggi dan ketepatan waktu. Ketersediaan transportasi yang terencana dengan baik bukan hanya memastikan bahwa pegawai dapat tiba di lokasi kegiatan sesuai jadwal, tetapi juga menjaga efektivitas koordinasi serta kualitas layanan yang diberikan. (Rahayu & Martini, 2022) Ketidakteraturan dalam sistem transportasi sering kali menimbulkan keterlambatan, membebani pegawai, dan mengganggu alur pekerjaan secara keseluruhan. Oleh karena itu, pengelolaan transportasi yang baik dapat mendukung kelancaran suatu kegiatan.

Penelitian sebelumnya menunjukkan bahwa pemilihan metode penjadwalan yang tepat dapat meminimalisir keterlambatan dan meningkatkan efisiensi proses. Misalnya, penelitian (Femy Mulya et al., 2020) yang membuktikan bahwa penjadwalan yang terstruktur membantu perusahaan jasa perawatan kendaraan menyelesaikan pekerjaan tepat waktu dan terorganisir. Hasil tersebut menegaskan

bahwa metode penjadwalan yang sesuai dapat menjadi solusi atas permasalahan keterlambatan dan ketidakefisienan di berbagai sektor.

Penjadwalan kegiatan merupakan proses penyusunan waktu pelaksanaan beserta sumber daya yang diperlukan dalam suatu kegiatan. Tujuan utama dari proses ini adalah untuk memastikan setiap kegiatan dapat berlangsung sesuai dengan waktu yang telah ditetapkan. Selain itu, penjadwalan juga berperan penting dalam meminimalkan potensi terjadinya benturan antara satu kegiatan dengan kegiatan lainnya, sehingga pelaksanaan dapat berjalan lebih efektif dan terorganisasi. (Yuslikhatun Ulya & Supriyono, 2025)

Kondisi serupa juga dihadapi oleh Kantor Perwakilan Bank Indonesia Provinsi Sulawesi Tengah sebagai salah satu lembaga pemerintahan yang memiliki peran strategis dalam menjaga stabilitas ekonomi, Kantor Perwakilan Bank Indonesia Provinsi Sulawesi Tengah secara rutin melaksanakan berbagai kegiatan dinas serta operasional yang melibatkan mobilitas pegawai ke lokasi-lokasi tertentu. Penjadwalan transportasi yang efektif menjadi kunci untuk mendukung kelancaran berbagai aktivitas, mulai dari kunjungan kerja, pertemuan dengan *stakeholder*, hingga kegiatan internal lainnya. Namun, proses penjadwalan kegiatan dan alokasi transportasi saat ini masih dilakukan secara manual, yang seringkali menimbulkan permasalahan seperti penyesuaian kembali jadwal yang ada dikarenakan adanya suatu kegiatan dengan prioritas tinggi diutamakan, ketidakefisienan penggunaan kendaraan dinas, serta keterlambatan dalam pelaksanaan kegiatan. dapat mengganggu kinerja pegawai dan mengurangi efektivitas operasional secara keseluruhan. Oleh karena itu, diperlukan sebuah sistem yang mampu mengelola penjadwalan secara otomatis dan cerdas, dengan mempertimbangkan prioritas kegiatan serta tenggat waktu pelaksanaannya.

Berdasarkan pada permasalahan yang dihadapi maka penelitian ini bertujuan untuk melakukan optimalisasi terhadap penjadwalan kegiatan di Kantor Perwakilan Bank Indonesia Provinsi Sulawesi Tengah dengan mengembangkan sistem berbasis *website*. Sistem ini akan menggunakan Algoritma *Priority queue* untuk mengatur prioritas permintaan secara otomatis. Algoritma ini memastikan permintaan dengan prioritas tertinggi diliyani lebih dulu, dan jika terdapat penjadwalan dengan tingkat prioritas yang sama maka sistem akan menerapkan prinsip *First Come First Served* (FCFS). (Setyawatu & Bachtiar Maulachela, 2020) Dengan adanya sistem ini, diharapkan penjadwalan dapat dilakukan secara lebih efisien, transparan, dan mampu meminimalkan benturan jadwal di masa mendatang.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas dapat ditarik rumusan masalah sebagai berikut:

1. Bagaimana melakukan optimasi pada sistem penjadwalan kegiatan di kantor perwakilan Bank Indonesia Provinsi Sulawesi Tengah untuk peningkatan efisiensi transportasi pegawai?
2. Bagaimana efektivitas algoritma *Priority queue* dalam menentukan prioritas dari suatu penjadwalan?

### 1.3 Batasan Masalah

Adapun Batasan masalah dari penelitian ini yaitu:

1. Data yang digunakan pada penelitian ini merupakan data yang diberikan oleh unit kerja *management intern* secara terbatas pada tahun 2024-2025, berupa data kegiatan, sopir, pegawai dan unit kendaraan.
2. Pengujian sistem yang digunakan berfokus pada penggunaan *framework Laravel* tanpa adanya penggunaan *framework* tambahan.
3. Penelitian tidak secara mendalam membahas aspek keamanan data, seperti enkripsi, *backup* otomatis, atau proteksi terhadap serangan *cyber*.
4. Sistem yang dibuat tidak melakukan optimalisasi pada *rute* atau waktu tempuh dari lokasi ke tempat kegiatan.
5. Perhitungan pembobotan kriteria menggunakan metode *Analytic Hierarchy Process* (AHP).

### 1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk melakukan optimalisasi pada sistem penjadwalan kegiatan kantor perwakilan Bank Indonesia Provinsi Sulawesi Tengah menggunakan algoritma *priority queue* untuk peningkatan efisiensi transportasi pegawai.

### 1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini, yaitu:

#### 1. Manfaat akademis

Dapat menambah wawasan pada bidang teknologi informasi dalam mengatur penjadwalan yang disesuaikan dengan kondisi yang diinginkan oleh *stackholder*, serta memberikan pemahaman praktis mengenai penerapan algoritma *priority queue* dalam sistem penjadwalan. Selain itu, hasil penelitian ini dapat menjadi acuan bagi penelitian lain yang ingin mengoptimalkan suatu sistem serupa, baik dalam sektor perbankan maupun industri lainnya, sehingga mendorong pengembangan inovasi dan efisiensi dalam pengelolaan sumber daya.

#### 2. Manfaat praktis

Manfaat praktis dari penelitian ini adalah pengembangan sistem yang dapat menyederhanakan proses penjadwalan kegiatan, sehingga pembagian tugas menjadi lebih adil dan merata berdasarkan tingkat prioritas. Selain itu, sistem ini juga membantu mengurangi waktu yang dibutuhkan dalam proses

koordinasi manual, sekaligus meningkatkan produktivitas pegawai dengan meminimalkan hambatan administratif.

### 1.6 Waktu Pelaksanaan

Pelaksanaan penelitian ini dilakukan selama 4 bulan pada kantor perwakilan Bank Indonesia Provinsi Sulawesi Tengah pada tahun 2024-2025. Berikut Pelaksanaan pembuatan skripsi oleh penulis.

*Tabel 1. 1 Waktu Pelaksanaan*

No.	Kegiatan	Bulan									
		April	Mei	Juni	Juli	Agustus	September	Oktober	November	Desember	
1.	Studi Literatur										
2.	Perancangan Perangkat Lunak										
3.	Pengumpulan Data										
4.	Analisis Data										
5.	Pembuatan Perangkat Lunak										
6.	Pengujian Sistem										
7.	Seminar Proposal										
8.	Perbaikan Proposal										
9.	Seminar Hasil										
10.	Perbaikan Hasil										
11.	Skripsi										

### 1.7 Sistematika Penulisan

Sistematika penulisan pada penelitian ini dikelompokkan menjadi beberapa sub bab dengan penyampaian sebagai berikut:

#### BAB I PENDAHULUAN

Bab ini berisi Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian dan Sistematika Penulisan.

#### BAB II TINJAUAN PUSTAKA & LANDASAN TEORI

Bab ini menjelaskan tentang perbandingan antara penelitian yang dilakukan oleh penulis dengan penelitian lain yang pernah ada sebelumnya dan berisikan teori yang berupa pengertian dan definisi yang diambil dari kutipan buku yang berkaitan dengan penyusunan laporan.

#### BAB III METODE PENELITIAN

Bab ini menjelaskan tentang bahan dan alat yang digunakan dalam penelitian dan tahapan penelitian yang dibuat berdasarkan teori dasar serta rumusan masalah yang ada.

## **BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini menjelaskan tentang penarikan kesimpulan dan juga terdapat saran untuk pengembangan dan penelitian aplikasi yang serupa dengan aplikasi yang telah dibuat penulis.

## **BAB V KESIMPULAN DAN SARAN**

Bab ini menyimpulkan temuan-temuan penting dari penelitian ini dan memberikan saran untuk pengembangan lebih lanjut. Saran dapat mencakup pengembangan fitur tambahan untuk aplikasi, serta rekomendasi bagi pengembang dan penjual dalam memanfaatkan aplikasi untuk meningkatkan pengalaman pengguna.

## **DAFTAR PUSTAKA**

Daftar pustaka berisi sumber informasi atau referensi-referensi yang dijadikan sebagai rujukan penulis dalam melakukan penelitian.

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Dalam penelitian ini terdapat beberapa penelitian-penelitian sebelumnya yang terkait dengan penelitian yang dilakukan penulis antara lain sebagai berikut:

1. Penelitian yang berjudul “Sistem Informasi Penjadwalan Supir Bus PO Logos Berbasis *Website*” yang diteliti oleh (Zakharia et al., 2021). Penelitian ini dilatarbelakangi oleh permasalahan dalam pengaturan jadwal pengemudi bus yang masih dilakukan secara manual, sehingga mengakibatkan operator membutuhkan waktu yang lebih lama dalam melakukan penjadwalan, yang pada akhirnya dapat berdampak pada ketidakefektifan operasional perusahaan. Untuk mengatasi permasalahan tersebut, penelitian ini menggunakan algoritma genetika dalam pengembangan sistem penjadwalan supir bus berbasis *website*. Algoritma genetika dipilih karena mampu mengoptimalkan penjadwalan berdasarkan berbagai faktor yang telah ditentukan. Hasil dari penelitian ini adalah sebuah sistem berbasis *website* yang memungkinkan operator dan supir untuk mengakses jadwal secara *real-time*. Namun dalam penelitian ini masih terdapat beberapa kesenjangan salah satunya adalah belum mempertimbangkan aspek prioritas penugasan berdasarkan pada riwayat kerja dari sopir. Oleh karena itu, penelitian lanjutan disarankan untuk mengembangkan sistem penjadwalan yang tidak hanya mempertimbangkan efisiensi waktu tetapi juga memastikan distribusi tugas yang lebih adil dan optimal bagi seluruh sopir yang terlibat.
2. Penelitian yang berjudul “Membangun Sistem Antrian *Online* Untuk Bimbingan Tugas Akhir” yang diteliti oleh (Dermawan et al., 2023). Penelitian ini dilatarbelakangi oleh permasalahan dalam proses penjadwalan serta bimbingan tugas akhir yang masih dilakukan secara manual. Dalam sistem yang ada, dosen harus mengatur jadwal bimbingan bagi mahasiswa yang terbagi dalam tiga kelas, yaitu dua jenjang untuk S1 dan satu jenjang untuk S2. Proses manual ini dinilai kurang efektif karena memerlukan koordinasi yang kompleks dari segi prosedur, waktu, dan tempat. Untuk mengatasi permasalahan tersebut, penelitian ini mengembangkan sistem antrian online berbasis *website* yang memungkinkan mahasiswa untuk menjadwalkan bimbingan dengan dosen pembimbing secara mandiri. Sistem ini juga menyediakan daftar antrian mahasiswa bimbingan yang dapat diakses oleh pengguna, serta memiliki fitur notifikasi otomatis melalui *WhatsApp* yang akan dikirimkan kepada mahasiswa yang telah melakukan penjadwalan maupun kepada dosen pembimbing. Meskipun sistem ini telah memberikan solusi dalam mengotomatiskan proses

penjadwalan bimbingan, penelitian ini masih memiliki beberapa keterbatasan. Salah satu kesenjangannya adalah belum adanya mekanisme prioritas dalam sistem antrian, misalnya mempertimbangkan tingkat kepentingan bimbingan atau jadwal kerja dosen yang sering berubah. Selain itu, sistem ini belum mengakomodasi kemungkinan penjadwalan ulang secara fleksibel jika terjadi pembatalan atau perubahan jadwal mendadak dari dosen maupun mahasiswa. Oleh karena itu, penelitian lanjutan direkomendasikan untuk mengembangkan sistem yang lebih adaptif guna meningkatkan efisiensi dan efektivitas sistem secara keseluruhan.

3. Penelitian yang berjudul “Sistem Layanan Antrian Klinik Kesehatan Berbasis *Web* dan *Whatsapp* Menggunakan Metode FIFO” yang diteliti oleh (Wardana et al., 2023). Penelitian ini dilatarbelakangi oleh permasalahan dalam sistem antrian di suatu klinik kesehatan yang masih dilakukan secara manual. Sistem manual ini menyebabkan penumpukan pasien di ruang tunggu, sehingga membuat kondisi klinik menjadi penuh dan kurang nyaman. Selain itu, pasien harus tetap berada di klinik hingga nomor antriannya dipanggil, yang menyebabkan ketidaknyamanan karena mereka tidak dapat melakukan aktivitas lain selama menunggu. Untuk mengatasi masalah tersebut, penelitian ini mengembangkan sistem layanan antrian berbasis *website* yang terintegrasi dengan *whatsApp* yang menggunakan metode *First In First Out* (FIFO). Dengan sistem ini, pasien dapat mengambil nomor antrian secara daring melalui *website* atau *whatsApp*, sehingga mereka tidak perlu menunggu di klinik dan dapat memantau sisa antrian dari jarak jauh. Hal ini diharapkan dapat mengurangi kepadatan di ruang tunggu serta meningkatkan kenyamanan bagi pasien dan efisiensi pelayanan di klinik. Namun, penelitian ini masih memiliki beberapa keterbatasan. Salah satu kesenjangan yang belum diatasi adalah belum adanya fitur prioritas bagi pasien dengan kondisi darurat atau lansia, yang memerlukan perhatian lebih cepat dibanding pasien lainnya. Selain itu, sistem ini belum sepenuhnya mempertimbangkan kemungkinan perubahan atau pembatalan antrian secara fleksibel, yang dapat terjadi karena alasan tertentu baik dari pasien maupun tenaga medis. Oleh karena itu, penelitian lanjutan direkomendasikan untuk mengembangkan sistem yang lebih adaptif dengan mempertimbangkan faktor prioritas pasien guna meningkatkan kualitas pelayanan kesehatan secara lebih optimal.
4. Penelitian yang berjudul “Implementasi Algoritma Antrian Prioritas Menggunakan *Array* di *Python* untuk Sistem Antrian Layanan Darurat” yang diteliti oleh (Valent et al., 2025). Penelitian ini dilatarbelakangi oleh permasalahan dalam manajemen layanan darurat di rumah sakit, di mana rumah sakit menghadapi tantangan dalam memastikan pelayanan yang responsif dan efisien, terutama dalam menangani pasien dengan tingkat urgensi yang berbeda-beda. Sistem antrian yang tidak berbasis prioritas sering kali menyebabkan pasien dengan kondisi kritis harus menunggu lebih lama dibandingkan pasien

dengan kondisi yang tidak mendesak, yang dapat berdampak pada keselamatan pasien. Untuk mengatasi permasalahan tersebut, penelitian ini mengembangkan sistem antrian layanan darurat berbasis *Graphical User Interface* (GUI) dengan algoritma antrian prioritas menggunakan *array* di *Python*. Sistem ini dirancang agar pasien dengan tingkat prioritas tertinggi mendapatkan pelayanan terlebih dahulu, sehingga rumah sakit dapat lebih optimal dalam menangani kasus-kasus darurat. Meskipun sistem ini telah memberikan solusi dalam meningkatkan efisiensi layanan darurat, penelitian ini masih memiliki beberapa keterbatasan. Salah satu kesenjangan yang belum diatasi adalah bahwa sistem yang masih berbasis GUI memiliki keterbatasan dalam aksesibilitas, karena hanya dapat digunakan pada perangkat yang telah terpasang oleh aplikasi tersebut. Hal ini dapat menghambat fleksibilitas pengguna, terutama bagi tenaga medis yang membutuhkan akses dari berbagai lokasi atau perangkat yang berbeda. Oleh karena itu, pengembangan sistem berbasis *website* menjadi solusi yang lebih fleksibel dan mudah diakses, baik oleh tenaga medis maupun *administrator* rumah sakit.

5. Penelitian yang berjudul “Penerapan Algoritma *Dynamic Priority Scheduling* Pada Aplikasi Antrian Pencucian Mobil Berbasis *Mobile*” yang diteliti oleh (Setyawatu & Bachtiar Maulachela, 2020). Penelitian ini dilatarbelakangi oleh permasalahan dalam layanan jasa pencucian mobil, di mana panjangnya antrian serta kurangnya informasi yang dapat diakses pelanggan mengenai perkiraan giliran mereka menyebabkan ketidaknyamanan. Hal ini berdampak pada pelanggan yang akhirnya memilih mencari tempat pencucian lain dengan antrian yang lebih sedikit atau yang menyediakan informasi antrian yang lebih transparan dan mudah diakses. Untuk mengatasi permasalahan tersebut, penelitian ini mengembangkan aplikasi berbasis *mobile* yang menggunakan algoritma *Dynamic Priority Scheduling* dalam pengelolaan antrian pencucian mobil. Algoritma ini memungkinkan penjadwalan antrian berdasarkan tingkat prioritas pelanggan, sehingga pelanggan dengan prioritas lebih tinggi mendapatkan giliran lebih dahulu. Dengan sistem ini, pelanggan dapat memperoleh informasi yang lebih akurat mengenai estimasi waktu pencucian, sehingga meningkatkan pengalaman dan kepuasan pelanggan. Meskipun penelitian ini telah memberikan solusi dalam mengoptimalkan pengelolaan antrian pencucian mobil, masih terdapat beberapa keterbatasan. Salah satu kesenjangan yang belum diatasi adalah ketergantungan pada aplikasi *mobile*, yang mengharuskan pelanggan memasang aplikasi terlebih dahulu untuk dapat mengakses layanan. Hal ini dapat menjadi kendala bagi pelanggan yang tidak terbiasa menggunakan aplikasi *mobile* atau yang memiliki keterbatasan dalam penyimpanan memori serta akses ke *internet*. Oleh karena itu, penelitian lanjutan direkomendasikan untuk mengembangkan versi berbasis *website* yang dapat diakses lebih fleksibel tanpa perlu instalasi aplikasi. Dengan demikian, sistem antrian pencucian mobil dapat menjangkau lebih banyak pelanggan.

6. Penelitian yang berjudul “Penjadwalan dan Pelaporan Menggunakan *Dynamic Priority Scheduling* dan *Geolocation* untuk Keamanan Lingkungan” yang diteliti oleh (Hilmawan Masyfa et al., 2023). Penelitian ini dilatar belakangi oleh permasalahan dalam sistem keamanan lingkungan di kawasan perumahan, di mana kegiatan ronda malam dilakukan secara terjadwal sebagai bentuk pengamanan wilayah. Namun, proses penjadwalan dan sistem pelaporan masih menjadi tantangan utama, karena jadwal ronda perlu disusun berdasarkan prioritas warga yang memilih jadwalnya sendiri, serta diperlukan sistem pelaporan yang dapat mencatat dan menunjukkan lokasi kejadian tindak kejahatan secara akurat. Sebagai solusi, penelitian ini menghasilkan sistem informasi keamanan lingkungan berbasis *website* yang memungkinkan warga untuk memilih jadwal ronda sesuai preferensi mereka. Sistem ini juga dilengkapi dengan fitur pengingat otomatis yang akan mengirimkan notifikasi kepada warga yang telah dijadwalkan untuk bertugas melalui *channel telegram*. Selain itu, sistem ini memiliki fitur pelaporan berbasis geolokasi, yang memungkinkan warga melaporkan tindak kejahatan disertai dengan bukti lokasi dan titik koordinat kejadian. Dengan demikian, data yang dikumpulkan dapat dianalisis untuk mengidentifikasi wilayah yang paling rawan terhadap tindak kejahatan, sehingga langkah preventif dapat lebih optimal dilakukan. Meskipun sistem ini telah berhasil mengotomatiskan proses penjadwalan dan pelaporan keamanan lingkungan, masih terdapat beberapa keterbatasan yang dapat dikembangkan lebih lanjut. Salah satu kesenjangan yang belum diatasi adalah belum terintegrasinya sistem dengan *platform* komunikasi yang populer digunakan saat ini, yaitu *WhatsApp*. Hal ini menyebabkan pemberitahuan jadwal ronda dan laporan keamanan masih terbatas pada sistem berbasis *website* serta *channel telegram*. Dengan integrasi *WhatsApp*, notifikasi dapat dikirimkan langsung kepada warga tanpa perlu memasang aplikasi tambahan seperti *telegram*, sehingga meningkatkan kenyamanan dan kemudahan dalam penggunaan sistem. Dengan pengembangan lebih lanjut, sistem ini dapat semakin meningkatkan efektivitas bagi keamanan lingkungan yang berbasis teknologi.

*Tabel 2. 1 Perbandingan Penelitian Terdahulu*

No.	Nama, Tahun, dan Judul	Metode	Hasil Penelitian	Keterkaitan Dengan Penelitian
1.	(Zakharia, Widiatry, dan Nova Noor Kamala Sari, 2021) “Sistem Informasi Penjadwalan Supir Bus PO Logos Berbasis <i>Website</i> ”	Penelitian ini menggunakan metode pengembangan perangkat lunak yaitu <i>waterfall</i> dimana tahapan yang digunakan	Penelitian ini menghasilkan sistem penjadwalan supir bus PO Logos dengan menggunakan algoritma genetika dimana pengujian dengan	Persamaan dengan penelitian ini yaitu menghasilkan sistem yang sama yaitu sistem penjadwalan untuk supir sedangkan perbedaan yaitu

		<p>hanya ada 4 dari 5 tahapan <i>waterfall</i> yaitu <i>Requirement Analysis and Definition</i> dengan melakukan wawancara serta studi literatur, Design dengan menggunakan <i>Unified Modelling Language</i> (UML), Pengkodean, dan Pengujian.</p>	<p>menggunakan black box testing dengan pengujian terhadap 10, 100, 250, 500, dan 1000 populasi yang menghasilkan pengujian dengan iterasi terbaik serta waktu eksekusi lebih cepat ketika populasi 1000 sehingga sistem telah berhasil untuk mengurangi tugas dari operator</p>	<p>algoritma yang digunakan adalah genetika serta metode pengembangan perangkat lunak yang digunakan adalah <i>waterfall</i></p>
2.	(Gumgum Dermawan, Zen Munawar, Cecep Kurnia Sastradipraja, Novianti Indah Putri, Sri Sutjiningtyas, 2023) “Membangun Sistem Antrian <i>Online</i> Untuk Bimbingan Tugas Akhir”	<p>Penelitian ini mengumpulkan data dengan melakukan wawancara, observasi serta studi literatur, penggunaan metode pengembangan perangkat lunak yang digunakan adalah <i>waterfall</i> serta melakukan desain pengembangan perangkat lunak dengan <i>Unified Modelling Language</i> (UML),</p>	<p>Pada penelitian ini menghasilkan suatu sistem antrian yang dapat diakses darimana pun serta terkoneksi dengan <i>whatsapp</i> sehingga dapat mengingatkan kepada dosen maupun mahasiswa terkait jadwal bimbingan</p>	<p>Persamaan penelitian yaitu menghasilkan sistem penjadwalan yang dapat melakukan <i>reminder</i> kepada pihak yang dituju melalui pesan <i>whatsapp</i> sedangkan perbedaannya yaitu menggunakan algoritma <i>First Come, First Served</i> (FCFS)</p>
3.	(Richo Wardana, Sucipto, Lina Firliana, 2023) “Sistem Layanan Antrian Klinik Kesehatan Berbasis <i>Web</i> ”	<p>Metode pengembangan perangkat lunak pada penelitian ini yaitu <i>waterfall</i> dengan pembuatan desain menggunakan <i>Data Flow Diagram</i> serta <i>Entity Relationship Diagram</i></p>	<p>Penelitian ini manghasilkan suatu sistem antrian yang dapat memudahkan pasien dalam mengambil nomor antrian dan melihat sisa antrian yang sedang dilayani, sistem yang dibuat juga terhubung dengan API <i>whatsapp</i> yang</p>	<p>Persamaan penelitian ini terletak pada pemanfaatan API <i>whatsapp</i> yang dapat terhubung pada sistem antrian berbasis <i>website</i> serta perbedaannya terletak pada metode pengembangan perangkat lunak yang</p>

			dapat memudahkan pasien mengambil nomor antrian serta melihat sisa antrian untuk pasien yang tidak dapat mengakses sistem antrian berbasis website	menggunakan <i>waterfall</i>
4.	(Hansel Valent, Rizal Muslim Sinaga, Sovantri Putra, Paskah Halawa, Selfi Audy Priscillia, Fanny Ramadhani, 2025) “Implementasi Algoritma Antrian Prioritas Menggunakan <i>Array</i> Di <i>Python</i> Untuk Sistem Antrian Layanan Darurat”	Penelitian ini menggunakan metode secara sistematis yang dimulai dengan identifikasi masalah, membuat <i>flowchart</i> sistem, lalu masuk ke dalam tahapan implementasi, dan diakhiri dengan pengujian pada sistem	Penelitian ini manghasilkan sebuah GUI untuk sistem antrian yang ada disuatu klinik yang menggunakan <i>array</i> pada bahasa <i>python</i> untuk mengurutkan antrian berdasarkan prioritas yang telah dimasukkan oleh pasien	Persamaan dari penelitian ini terdapat pada penggunaan algoritma prioritas untuk menghasilkan antrian yang sesuai prioritas yang telah ditentukan sedangkan perbedaannya adalah penerapan sistem antrian yang berbasis <i>array</i> di <i>python</i>
5.	(Rizki Setyawatu, Adam Bachtiar Maulachela, 2020) “Penerapan Algoritma <i>Dynamic Priority Scheduling</i> Pada Aplikasi Antrian Pencucian Mobil Berbasis <i>Mobile</i> ”	Metode pengembangan perangkat lunak yang digunakan adalah <i>Rapid Application Development</i> (RAD)	Penelitian ini menghasilkan sistem penjadwalan berbasis website yang dapat mengatasi waktu tunggu pelanggan dengan menggunakan algoritma <i>dynamic priority scheduling</i>	Persamaan penelitian ini yaitu menghasilkan suatu sistem penjadwalan berbasis website yang menggunakan algoritma prioritas sedangkan perbedaannya adalah metode pengembangan perangkat lunak yaitu menggunakan <i>Rapid Application Development</i> (RAD)
6.	(Faiz Hilmawan Masyfa, Dany Primanita Kartikasari, Tibyani, 2023) “Penjadwalan dan Pelaporan Menggunakan <i>Dynamic Priority Scheduling</i> dan <i>Geolocation</i> untuk	Metode yang digunakan bertipe implementatif dengan tahapan yang diawali identifikasi masalah yang	Pada penelitian ini menghasilkan suatu sistem informasi keamanan lingkungan yang dapat mengatur jadwal untuk para warga yang akan	Persamaan dari penelitian ini adalah menghasilkan suatu sistem informasi penjadwalan dengan memanfaatkan

	Keamanan Lingkungan”	dilanjutkan dengan studi literatur, kemudian pengumpulan data dilakukan dengan wawancara sebagai daa primer dan studi literatur sebagai data sekunder sedangkan untuk metode pengembangan perangkat lunak menggunakan <i>waterfall</i>	bertugas menjaga lingkungan, dimana penjadwalannya diurutkan berdasarkan prioritas dari warga serta akan ada <i>reminder</i> yang diberikan ke warga melalui <i>chanel telegram</i> , kemudian sistem tersebut dapat menjadi tempat pelaporan kejadian yang terjadi di lingkungan tersebut dengan memasukan lokasi kejadian serta gambar dari lokasinya	algoritma prioritas serta sistem dapat memberikan <i>reminder</i> kepada orang yang dituju sedangkan untuk perbedaannya terletak pada metode pengembangan perangkat lunak yang menggunakan <i>waterfall</i> serta pemanfaatan API yang berbeda yaitu <i>chanel telegram</i>
--	----------------------	--	---	---

Pada Tabel 2.1, dapat disimpulkan bahwa perbedaan dengan penelitian terdahulu terletak pada metode pengembangan perangkat lunak. Peneliti menggunakan metode pengembangan perangkat lunak yaitu *prototyping*.

## 2.2 Landasan Teori

### 2.2.1 Bank Indonesia

Bank Indonesia (BI) merupakan lembaga yang sangat vital dalam kehidupan perekonomian nasional karena kebijakan-kebijakan yang ditempuh oleh BI akan memiliki dampak langsung yang dirasakan oleh masyarakat. Bank Indonesia merupakan lembaga bank sentral yang memiliki peranan krusial dalam perekonomian, khususnya dalam sektor moneter, keuangan, dan perbankan. Peranan tersebut tercermin dalam tugas-tugas utama bank sentral, yang melibatkan penetapan dan pelaksanaan kebijakan moneter, pengaturan dan pengawasan terhadap sektor perbankan, serta menjaga kelancaran sistem pembayaran. (Simanullang et al., 2024)

### 2.2.2 Penjadwalan

Penjadwalan adalah bagian yang Strategis dari proses perencanaan dan juga merupakan rencana pengaturan aturan kerja serta mengalokasikan sumber daya baik waktu maupun fasilitas untuk setiap pekerjaan yang harus diselesaikan. Penjadwalan merupakan sebuah fungsi pengambilan keputusan untuk rencana pengaturan dan mengalokasikan sumber daya yang menyangkut waktu dan fasilitas sarana. Masalah penjadwalan muncul dikarenakan adanya beberapa pekerjaan yang

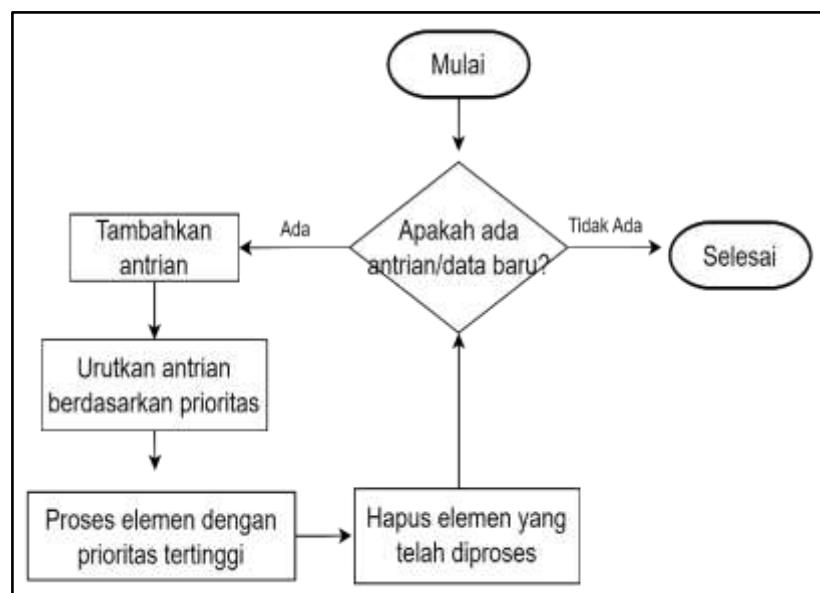
dikerjakan secara bersamaan, sedangkan sumber daya yang dimiliki terbatas, masalah penjadwalan dapat terjadi dimana saja dan dalam kasus apapaja (Amir S et al., 2021)

### 2.2.3 Metode *Prototype*

*Prototype* adalah model yang digunakan untuk mensimulasikan program kepada pengguna untuk memastikan kesesuaian dengan kebutuhan mereka. Dalam konteks pengembangan sistem informasi, *prototype* sering digunakan ketika pengguna telah menetapkan tujuan umum perangkat lunak, tetapi tidak dapat merinci spesifikasi kebutuhan untuk fitur dan fungsi yang akan dimiliki oleh perangkat lunak tersebut. *Prototype* adalah alat yang efektif untuk memperoleh umpan balik pada sistem yang diusulkan dan untuk menjelaskan bagaimana sistem dapat memenuhi kebutuhan informasi pengguna. (Syarif & Risdiansyah, 2024)

### 2.2.4 Algoritma *Priority queue*

*Priority queue* merupakan suatu algoritma yang menitikberatkan proses penentuan antrian berdasarkan pada aturan prioritas yang ditentukan. Pada algoritma ini, setiap proses yang masuk kedalam antrian diidentifikasi berdasarkan nomor prioritas. Kemudian, algoritma ini akan mengatur urutan antrian, mengacu pada nomor prioritas setiap proses, sehingga proses dengan nomor prioritas yang paling tinggi akan mendapatkan urutan antrian pertama. Namun apabila terdapat proses dengan tingkat prioritas yang sama, maka algoritma akan melihat mana proses yang masuk lebih dahulu masuk kedalam antrian. Artinya algoritma ini, akan mengadopsi algoritma *First Come First Served* (FCFS) sebagai solusinya (Setyawatu & Bachtiar Maulachela, 2020). Berikut *flowchart* dari algoritma priority queue yang dapat dilihat pada gambar 2.1



Gambar 2. 1 Flowchart Priority Queue

Adapun penjelasan dari diagram alir algoritma *priority queue* sebagai berikut:

1. Mulai  
Merupakan proses awal dari algoritma *priority queue*.
2. Kondisi antrian/data  
Pemeriksaan kondisi apakah ada antrian/data baru yang ditambahkan.
3. Tambahkan antrian  
Sistem akan menambahkan antrian/data yang baru saja ditambahkan.
4. Urutkan antrian berdasarkan prioritas  
Setelah antrian/data baru yang ditambahkan maka akan dinilai prioritas dari antrian tersebut apakah tinggi atau rendah yang kemudian akan diurutkan.
5. Proses elemen dengan prioritas tertinggi  
Ketika antrian yang telah diproses nilainya baik yang nilai prioritasnya tinggi atau rendah maka akan dilanjutkan dengan proses mengurutkan antrian dari yang prioritas tertinggi terlebih dahulu.
6. Hapus elemen yang telah diproses  
Ketika antrian dengan prioritas tertinggi telah didapatkan maka antrian tersebut akan diproses terlebih dahulu dan akan dihapuskan dalam daftar antrian untuk dilanjutkan ke antrian selanjutnya.
7. Selesai  
Merupakan proses akhir yang akan dijalankan jika tidak terdapat kondisi antrian/data baru yang ditambahkan.

Berikut *pseudocode* untuk algoritma *priority queue*:

```
CLASS PriorityQueue:  
// Properti (Variabel Kelas)  
PRIVATE QueueList AS ARRAY  
// Konstruktor (Inisialisasi)  
METHOD __CONSTRUCT:  
QueueList = ARRAY KOSONG  
END METHOD  
// 1. Metode ENQUEUE (Menambahkan Elemen ke Antrean)  
METHOD ENQUEUE(Data, Priority):  
// Buat objek/item baru  
Item = { 'data': Data, 'priority': Priority }  
Added = FALSE  
// Cari posisi penyisipan (insert) berdasarkan prioritas  
// Loop melalui setiap elemen dalam QueueList  
FOR i DARI 0 SAMPAI PANJANG(QueueList) - 1:  
// Cek: Apakah prioritas elemen saat ini LEBIH BESAR  
// dari prioritas item yang baru? (Prioritas lebih kecil = lebih tinggi)  
IF QueueList[i]['priority'] > Priority THEN  
// Sisipkan Item baru sebelum elemen saat ini (di indeks i)  
SISIPKAN Item KE QueueList PADA INDEKS i  
Added = TRUE  
BREAK LOOP // Keluar dari perulangan karena sudah disisipkan
```

```

END IF
END FOR
// Jika perulangan selesai dan Item belum ditambahkan (karena prioritasnya
terendah)
IF Added IS FALSE THEN
// Tambahkan Item di akhir antrean
TAMBAHKAN Item KE AKHIR QueueList
END IF
END METHOD
// 2. Metode DEQUEUE (Mengambil Elemen Prioritas Tertinggi)
METHOD DEQUEUE:
// Cek apakah antrean kosong
IF NOT IS_EMPTY() THEN
// Elemen prioritas tertinggi selalu berada di awal (indeks 0)
HASIL = HAPUS DAN AMBIL ELEMEN PERTAMA DARI QueueList
RETURN HASIL
END IF
// Jika antrean kosong
RETURN NULL
END METHOD
// 3. Metode IS_EMPTY (Cek Kekosongan Antrean)
METHOD IS_EMPTY:
// Cek apakah panjang QueueList sama dengan 0
RETURN PANJANG(QueueList) == 0
END METHOD
// 4. Metode DISPLAY (Menampilkan Isi Antrean)
METHOD DISPLAY:
// Ulangi setiap Item dalam QueueList
FOR EACH Item IN QueueList:
TAMPILKAN "Data: " + Item['data'] + " | Prioritas: " + Item['priority'] +
BARIS_BARU
END FOR
END METHOD
END CLASS

```

### 2.2.5 Metode *Analytic Hierarchy Process* (AHP)

*Analytical Hierarchy Process* merupakan suatu metode pendukung keputusan yang dikembangkan oleh Thomas L. Saaty. Model pendukung keputusan ini akan menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hierarki. Menurut Saaty (1993), hierarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur multilevel dimana level pertama adalah tujuan, yang diikuti level faktor, kriteria, sub kriteria, dan seterusnya hingga level terakhir dari alternatif. (Galuh & Siregar, 2025)

Prinsip kerja AHP adalah penyederhanaan suatu persoalan kompleks yang tidak terstruktur, strategis, dan dinamik menjadi bagian-bagiannya, serta menata dalam suatu hierarki. Kemudian tingkat kepentingan setiap variabel diberi nilai numerik secara subjektif tentang arti penting variabel tersebut secara relatif dibandingkan dengan variabel lain. (Munthafa & Mubarok, 2017)

Berikut tahapan dalam metode AHP:

- a. Membuat matrik perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap tujuan atau kriteria yang setingkat di atasnya.

*Tabel 2. 2 Tabel Matrik Perbandingan Berpasangan*

	Kriteria-1	Kriteria-2	Kriteria-3	Kriteria-n
Kriteria-1	K11	K12	K13	K1n
Kriteria-2	K21	K22	K23	K2n
Kriteria-3	K31	K32	K33	K3n
Kriteria-n	Kn1	Kn2	Kn3	Knn

- b. Menghitung perbandingan prioritas dalam bentuk matriks. Skala dalam mengisi matriks menggunakan skala Saaty pada tabel berikut:

*Tabel 2. 3 Skala Penilaian AHP*

Intensitas Kepentingan	Keterangan
1	Sama Penting
3	Cukup Penting
5	Lebih Penting
7	Sangat Lebih Penting
9	Mutlak Lebih Penting
2,4,6,8	Nilai antara dua skala nilai yang berdekatan

- c. Persamaan dari bobot prioritas pada AHP

$$W_i = \frac{\sum_{j=1}^n (A_{ij} \div \sum_{k=1}^n A_{kj})}{n} \quad (1)$$

Dimana:

- $W_i$  : Bobot prioritas untuk kriteria ke-i.  
 $A_{ij}$  : Elemen matriks perbandingan berpasangan di baris i dan kolom j.  
 $\sum_{k=1}^n A_{kj}$  : Jumlah seluruh elemen di kolom j pada matriks.  
 $\sum_{j=1}^n$  : Penjumlahan dari indeks j mulai dari 1 hingga n.  
 $n$  : Jumlah kriteria.

- d. Nilai Eigen Maksimum

$$\lambda_{max} = \frac{\sum \lambda_i}{n} \quad (2)$$

Dimana:

- $\sum \lambda_i$  : Nilai rata-rata dari hasil semua kriteria.  
 $n$  : Jumlah kriteria.

- e. Indeks Konsistensi

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (3)$$

Dimana:

$\lambda_{max}$  : Nilai Maksimum

$n$  : Jumlah kriteria.

f. Rasio Konsistensi

$$CR = \frac{CI}{RI} \quad (4)$$

$CI$  : *Consistency Index*.

$CR$  : *Consistency Ratio*.

*Tabel 2. 4 Random Consistency Index*

$n$	<i>Random Consistency Index</i>
1	0,00
2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49

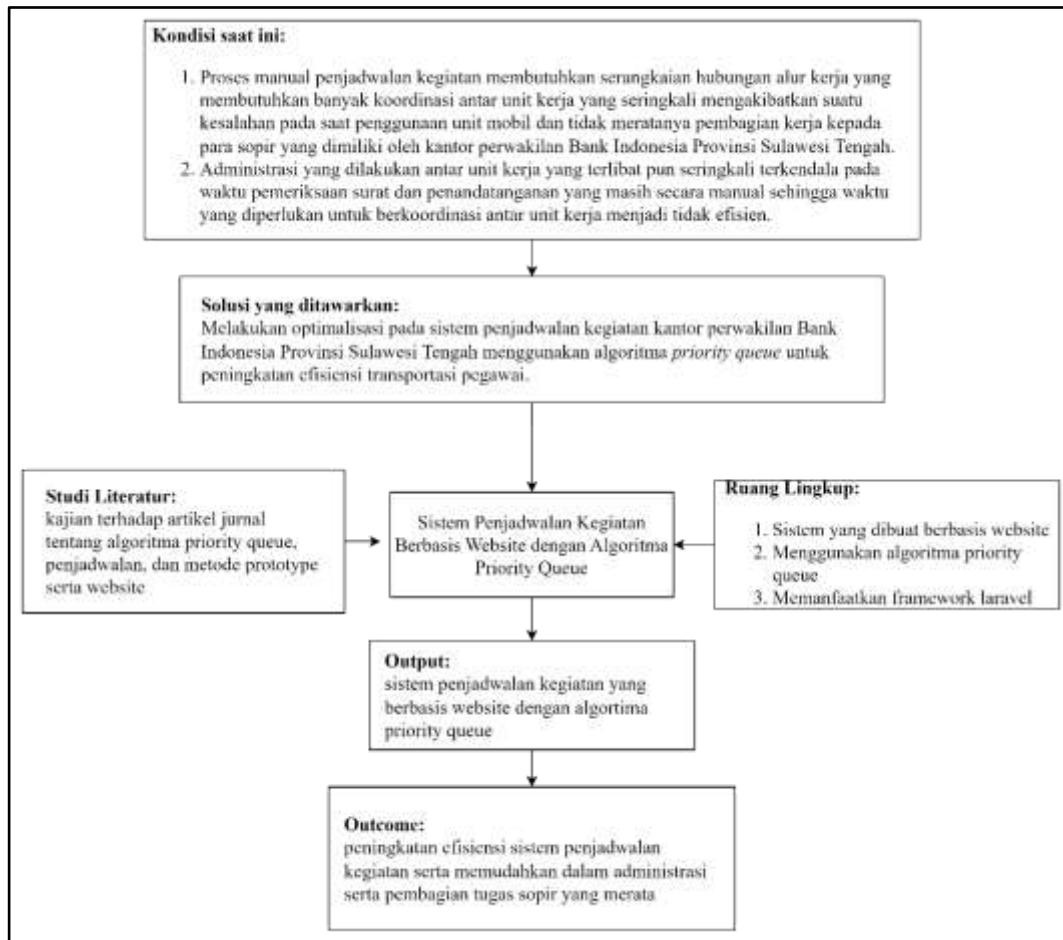
Jika nilai  $CR < 0,1$ , maka matriks dianggap konsisten.

## 2.2.6 Laravel

*Laravel* merupakan sebuah kerangka kerja pemrograman yang berbasis *open source* yang dipakai oleh banyak *developer* dari seluruh dunia. Kemudahan penggunaan dan dokumentasi yang lengkap menjadi salah satu faktor mengapa *Laravel* menjadi primadona dalam beberapa tahun terakhir. *Laravel* juga menjadi salah satu *framework* yang dapat membantu *developer* untuk memaksimalkan penggunaan PHP didalam proses pengembangan *website*. Selain itu, *Laravel* juga memiliki beberapa fitur unggulan, seperti *template engine*, *routing*, dan *modularity* (Aipina & Witriyono, 2022).

## 2.3 Kerangka Pemikiran

Kerangka pemikiran dalam penelitian ini disusun sebagai dasar konseptual yang menggambarkan alur logis dari identifikasi permasalahan hingga tujuan yang ingin dicapai melalui pengembangan sistem. Berikut merupakan kerangka pemikiran untuk penelitian ini:



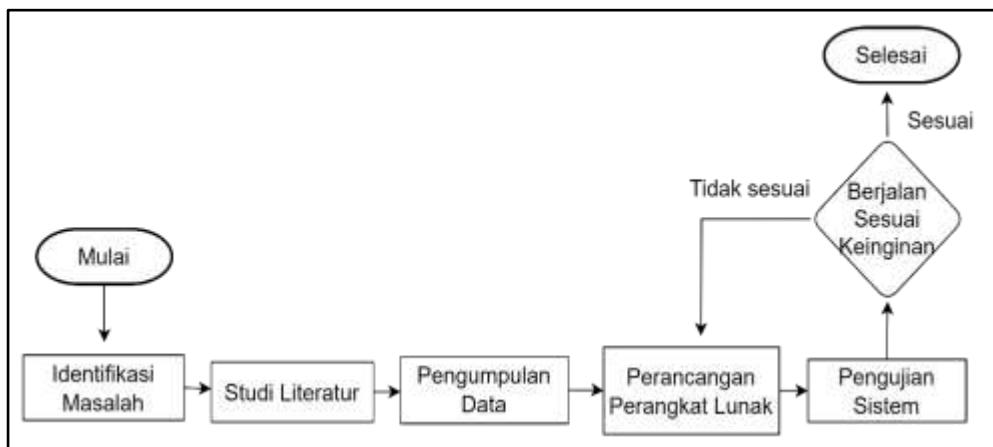
*Gambar 2. 2 Kerangka Pemikiran*

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Tahapan dan *Diagram Alir* Penelitian

Terdapat beberapa tahapan yang dilakukan oleh penulis secara sistematis dan terstruktur baik untuk dapat mencapai tujuan dari penelitian ini yaitu melakukan optimalisasi terhadap penjadwalan kegiatan bank indonesia provinsi Sulawesi Tengah. Berikut *diagram* alir dari penelitian yang dapat dilihat pada gambar 3.1 berikut:



*Gambar 3. 1 Alur Penelitian*

Adapun penjelasan dari *diagram* alir penelitian adalah sebagai berikut :

##### 1. Identifikasi Masalah

Tahapan pada penelitian ini diawali dengan mengidentifikasi permasalahan yang terjadi di Kantor Perwakilan Bank Indonesia Provinsi Sulawesi Tengah. Permasalahan utama yang dihadapi adalah sistem penjadwalan kegiatan yang masih dilakukan secara manual. Dengan jumlah sopir dan unit kendaraan yang terbatas, sering kali terjadi bentrok penugasan. Selain itu, pegawai yang membutuhkan penjadwalan kegiatan dengan sopir dan kendaraan tertentu harus melalui proses pengecekan manual oleh kepala sopir, ditambah lagi dengan pembuatan surat permohonan penggunaan kendaraan yang memerlukan persetujuan dari manajer unit. Proses ini tidak hanya memakan waktu, tetapi juga berpotensi menimbulkan kesalahan koordinasi.

Berdasarkan permasalahan tersebut, diperlukan optimalisasi sistem penjadwalan yang lebih terstruktur dan efisien. Dengan pengembangan sistem berbasis teknologi, diharapkan proses penjadwalan dapat berjalan lebih cepat, transparan, dan merata.

## 2. Studi Literatur

Pada tahapan ini, penulis melakukan studi literatur sebagai langkah awal untuk memperoleh pemahaman yang mendalam terkait permasalahan yang diangkat dalam penelitian. Studi literatur dilakukan dengan menelusuri berbagai sumber informasi yang relevan, baik dari jurnal ilmiah, artikel, laporan penelitian sebelumnya, maupun referensi teknologi yang tersedia secara daring. Tujuan dari studi ini adalah untuk mengidentifikasi teknologi dan pendekatan yang sesuai dalam mengatasi permasalahan sistem penjadwalan kegiatan yang masih dilakukan secara manual.

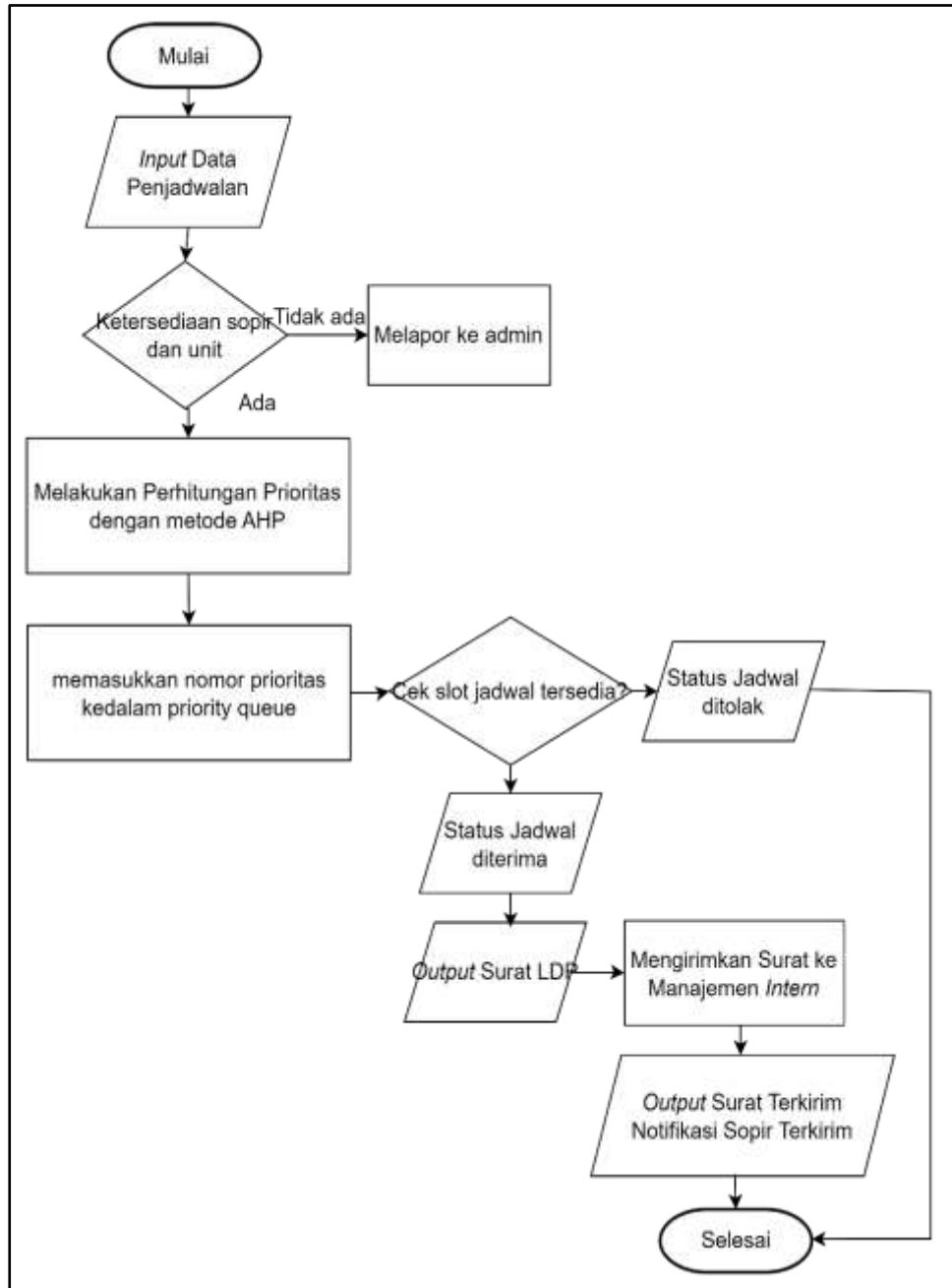
Penulis mencari rujukan untuk fitur-fitur yang umum diterapkan pada sistem penjadwalan modern, selain itu, penulis menelusuri website-website yang menjadi referensi utama dalam perancangan sistem informasi berbasis web, termasuk struktur antarmuka pengguna, alur sistem, serta penerapan prinsip efisiensi dan transparansi dalam penjadwalan tugas. Hasil dari studi literatur ini menjadi dasar dalam merancang solusi yang tepat guna, relevan, dan mampu menjawab kebutuhan dari Kantor Perwakilan Bank Indonesia Provinsi Sulawesi Tengah secara optimal.

## 3. Pengumpulan Data

Data yang dikumpulkan merupakan data primer yang mendukung pembuatan sistem seperti data pegawai dan sopir sebagai *user* yang akan menggunakan sistem, data unit kendaraan, serta data pendukung pembuatan surat lembar disposisi pejabat. Adapun data yang diberikan merupakan data pada tahun 2024. Sedangkan untuk data sekunder merupakan data yang mendukung penelitian seperti informasi dari sumber jurnal dan *website* yang serupa dengan penelitian.

## 4. Perancangan Perangkat Lunak

Pada tahapan ini penulis melakukan perancangan terhadap sistem yang akan dibuat berdasarkan pada permasalahan yang terjadi serta studi literatur yang telah dilakukan. Sistem ini akan menerapkan algoritma *priority queue* dengan penggunaan *framework Laravel* dalam pembangunan *website* penjadwalan kegiatan, bahasa pemrograman yang digunakan adalah bahasa *php* dan *javascript*.



*Gambar 3. 2 Flowchart Sistem*

Adapun penjelasan terkait *flowchart* sistem adalah sebagai berikut:

1. Mulai

Langkah awal dalam proses untuk memulai penggunaan sistem

2. Input data penjadwalan

Pegawai dan *admin* dapat melakukan *input* data untuk melakukan penjadwalan.

3. Kondisi ketersediaan sopir dan unit

Proses pemeriksaan kondisi terhadap ketersediaan sopir dan unit yang akan digunakan oleh pengguna.

4. Melapor ke *admin*

Jika proses pemeriksaan kondisi ketersediaan sopir dan unit tidak ada maka pengguna perlu untuk melapor ke *admin* untuk dapat memberikan konfirmasi terkait jumlah sopir dan unit yang telah selesai bertugas.

5. Melakukan Perhitungan Prioritas dengan metode AHP

Setelah input data penjadwalan dilakukan selanjutnya perlu untuk melakukan perhitungan prioritas untuk mendapatkan nomor prioritas disetiap penjadwalan, perhitungan sendiri menggunakan metode AHP yang kemudian akan mendapatkan bobot dari kriteria yang telah ditentukan. kemudian bobot tersebut digunakan untuk menghitung nomor prioritas

6. Memasukkan nomor prioritas kedalam *priority queue*

Setelah mendapatkan bobot dari metode AHP kemudian dilanjutkan dengan menjumlahkan kriteria dan mendapatkan nomor prioritas, nomor tersebut digunakan untuk melihat seberapa tinggi tingkat prioritas dari suatu jadwal yang kemudian akan dibandingkan dengan jadwal yang memiliki tanggal yang sama, selanjutnya dimasukkan kedalam algoritma untuk diurutkan berdasarkan prioritas yang ada untuk kemudian diproses.

7. Kondisi cek slot jadwal tersedia?

Pemeriksaan yang dilakukan untuk melihat apakah pada tanggal yang dipilih sudah terdapat jadwal lainnya dan telah memenuhi kapasitas slot atau belum.

8. Status jadwal ditolak

Jika pemeriksaan yang dilakukan terpenuhi maka jadwal yang baru akan memiliki status ditolak dan perlu untuk memilih ditanggal yang lain.

9. Status jadwal diterima

Jika pemeriksaan yang dilakukan memiliki slot yang masih kosong maka jadwal yang baru akan memiliki status diterima.

10. *Output* Surat LDP

Proses ini akan memberikan keluaran surat LDP yang dapat diunduh oleh pengguna untuk selanjutnya mendapatkan persetujuan dari menejer unit pegawai

## 11. Mengirimkan Surat ke Manajemen *Intern*

Pada proses ini pegawai yang telah mendapatkan persetujuan dari menejer unit maka perlu untuk mengirimkan surat tersebut melalui sistem ke unit manajemen *intern*.

## 12. *Output* Surat Terkirim dan Notifikasi Sopir Terkirim

Proses ini akan memberikan keluaran berupa notifikasi ke pengguna bahwa surat yang telah dikirim melalui sistem telah berhasil diteruskan ke unit manajemen *intern* dan juga berhasil memberikan notifikasi ke sopir terkait penugasan.

## 13. Selesai

Proses telah selesai

## 5. Pengujian Sistem

Pada tahapan ini, penulis akan melakukan pengujian terhadap suatu sistem yang telah berhasil dikembangkan. Pengujian ini bertujuan untuk memastikan bahwa seluruh fitur yang dibuat berfungsi sesuai dengan yang diharapkan, serta untuk melakukan evaluasi dini terhadap kesesuaian alur sistem mulai dari proses *input* hingga implementasi algoritma yang digunakan.

### 3.2 Bahan Penelitian

Penelitian ini menggunakan jenis data primer sebagai sumber data utamanya, yang merupakan data yang dikumpulkan langsung pada lokasi penelitian. Data yang diberikan oleh unit manajemen *intern* berupa data kegiatan, pegawai, sopir, dan unit kendaraan. Sedangkan untuk data sekunder merupakan data yang mendukung penelitian seperti informasi dari sumber jurnal dan *website* yang serupa dengan sistem yang akan dibangun dalam penelitian.

### 3.3 Instrumen Penelitian

Alat penelitian yang digunakan adalah perangkat lunak dan perangkat keras sebagai berikut.

1. Perangkat lunak yang digunakan sebagai berikut :
  - a. *OS Windows 11*
  - b. *XAMPP*
  - c. *Visual Studio Code*
  - d. *Web Browser*
2. Perangkat keras yang digunakan sebagai berikut :
  - a. *Laptop ThinkBook 14 G2 ARE*
  - b. *Windows 11*
  - c. *AMD Ryzen 5 4500U*

- d. *RAM 8GB*
- e. *SSD 500 GB*

### 3.4 Desain Penelitian

#### 3.4.1 Jenis Penelitian

Penelitian ini menggunakan pendekatan kuantitatif, Metode kuantitatif melibatkan pengumpulan dan analisis data berdasarkan angka, yang memungkinkan peneliti untuk menyelidiki fenomena dan hubungan antara variabel dengan pendekatan ilmiah yang terstruktur. (A. Siroj et al., 2024) Pendekatan ini dipilih dengan tujuan untuk menghasilkan analisis yang terukur, objektif, dan dapat mengidentifikasi pola atau hubungan antara variabel yang diteliti secara akurat.

#### 3.4.2 Tipe Penelitian

Penelitian ini menggunakan tipe deskriptif, di mana penelitian ini dimaksudkan untuk mengeksplorasi atau mengklarifikasi suatu gejala, fenomena atau kenyataan sosial yang ada, penelitian deskriptif berusaha untuk mendeskripsikan sejumlah variabel yang berkenaan dengan masalah dan unit yang diteliti. (Hasan & M.Syahran, 2023) Penelitian ini bertujuan untuk memberikan gambaran serta menjelaskan bagaimana sistem penjadwalan kegiatan yang dibangun mampu meningkatkan efisiensi kerja. Melalui pendekatan ini, data yang dikumpulkan akan diolah untuk mengukur efektivitas sistem dalam mengoptimalkan proses penjadwalan, mengurangi kesalahan, dan mempercepat alur kerja secara keseluruhan.

### 3.5 Teknik Pengumpulan Data

Adapun teknik pengumpulan data yang digunakan dalam penelitian ini meliputi observasi langsung dan studi literatur. Dalam proses pengumpulan data, penulis perlu memahami berbagai kebutuhan yang diperlukan dalam membangun sistem penjadwalan kegiatan di kantor, sehingga komunikasi dengan pihak-pihak yang bertanggung jawab terhadap sistem menjadi hal yang penting untuk dilakukan. Melalui komunikasi ini, penulis dapat mengidentifikasi informasi yang relevan serta memperoleh masukan dalam merancang fitur-fitur yang dibutuhkan dalam sistem. Oleh karena itu, survei lapangan secara langsung perlu dilakukan penulis guna memperoleh data yang lebih akurat dan memperoleh gambaran menyeluruh terkait alur kerja serta permasalahan yang ada, sehingga sistem yang dibangun benar-benar sesuai dengan kebutuhan pengguna di lapangan.

Setelah tahapan observasi dilakukan, proses dilanjutkan dengan tahap studi literatur. Pada tahap ini, penulis mencari dan mengkaji berbagai sumber yang relevan, baik dari buku bacaan maupun sumber daring, guna menunjang

keberhasilan dalam membangun sistem penjadwalan kegiatan yang efektif dan sesuai kebutuhan.

### 3.6 Metode Analisis Data

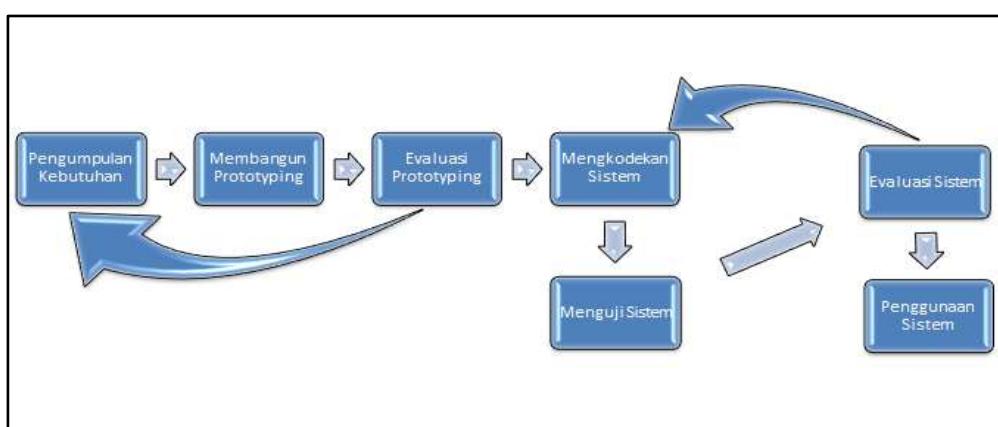
Metode analisis data merupakan suatu proses sistematis yang digunakan untuk mencari, mengolah, dan menganalisis data di lapangan serta memberikan kesimpulan akhir yang dapat mendukung penelitian. Data yang dikumpulkan akan diolah dan dianalisa menggunakan metode kuantitatif untuk kemudian dapat dilakukan uji performa serta menguji validitas dari algoritma yang digunakan, penggunaan metode perhitungan *throughput* untuk mengukur berapa banyak kegiatan yang berhasil dijadwalkan dan diselesaikan tanpa konflik & sesuai prioritas. Kemudian hasil dari pengujian tersebut akan memberikan gambaran terkait optimalisasi sistem penjadwalan.

### 3.7 Objek Penelitian dan Lokasi Penelitian

Objek dalam penelitian ini adalah optimalisasi penjadwalan kegiatan yang sebelumnya dilakukan secara manual, kemudian disederhanakan melalui pengembangan sistem penjadwalan berbasis *website*. Penelitian ini dilaksanakan selama empat bulan, terhitung sejak bulan Maret 2024 hingga Juni 2024, dan berlokasi di Kantor Perwakilan Bank Indonesia Provinsi Sulawesi Tengah.

### 3.8 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah model *prototype*. Model *prototype* adalah suatu proses yang memungkinkan developer membuat sebuah model *software*, metode ini baik digunakan apabila *client* tidak bisa memberikan informasi yang maksimal mengenai kebutuhan yang diinginkannya.(Wulandari et al., 2024) Melalui pendekatan ini, *client* dapat melihat gambaran nyata dari sistem yang akan dibangun dan memberikan masukan secara langsung, sehingga pengembangan dapat disesuaikan dengan kebutuhan secara bertahap.



Gambar 3. 3 Metode Prototype

Berikut penjelasan dari gambar diatas sebagai berikut:

1. Pengumpulan kebutuhan

Tahapan awal dalam model *prototype* ini bertujuan untuk mengumpulkan serta menganalisis data dan informasi yang perlu untuk diberikan oleh pengguna. Kemudian hasil dari data yang telah diberikan akan dikembangkan menjadi suatu sistem untuk menyelesaikan masalah yang dihadapi oleh pengguna. Adapun Tahapan pengumpulan data yang dilakukan dengan wawancara, observasi dan permintaan data kepada pengguna serta *admin* dari kantor tersebut.

2. Membangun *prototype*

Tahapan dilanjutkan dengan membuat *prototype* sistem yang akan dibuat, dalam hal ini *prototype* dimaksudkan untuk dapat memberikan suatu gambaran awal terhadap pengguna terkait sistem yang akan digunakan. pembuatan sistem ini memanfaatkan *framework* figma untuk membuat suatu tampilan yang dapat dipahami oleh pengguna, selain melakukan desain antarmuka pada tahapan ini juga melakukan desain *database* beserta alur kerja dari sistem yang akan dibuat.

3. Evaluasi *prototype*

Setelah tahapan pembuatan *prototype* dilakukan selanjutnya hasilnya dilakukan evaluasi untuk menentukan fitur maupun desain yang disesuaikan dengan permintaan dari pengguna, pengguna juga perlu untuk melihat apakah alur dari pengguna sistem sudah sesuai dengan kebutuhan. Jika pengguna masih belum menerima dan memberikan beberapa masukkan maka tahapan pengembangan sistem akan diulang kembali ke proses membuat *prototype* namun jika pengguna telah menyetujui maka tahapan akan dilanjutkan.

4. Mengkodekan sistem

Tahapan selanjutnya adalah proses penulisan kode program. Pada tahapan ini, seluruh rancangan yang telah disusun pada tahap desain baik itu rancangan fitur, alur sistem, maupun struktur antarmuka akan mulai diterjemahkan ke dalam bentuk implementasi kode. *Prototype* yang telah mendapatkan persetujuan dari pengguna akan menjadi acuan utama dalam proses pengkodean ini. Dengan demikian, setiap komponen yang telah dirancang pada tahap sebelumnya diimplementasikan menggunakan bahasa pemrograman dan teknologi yang sesuai.

5. Pengujian sistem

Setelah sistem telah dibuatkan kode program, kemudian sistem dilakukan pengujian untuk melihat apakah fitur yang telah dibuat berjalan

sesuai dengan permintaan dari pengguna serta alur dari sistem juga sudah sesuai maka sistem dilanjutkan ke tahap selanjutnya.

#### 6. Evaluasi sistem

Tahapan dilakukan dengan melakukan evaluasi terhadap sistem yang telah dikodekan, evaluasi ini dilakukan agar pengguna dapat memastikan bahwa sistem yang akan digunakan sudah sesuai ekspektasi dari pengguna baik secara desain, alur maupun fitur yang ada didalam sistem. Apabila dari pengguna masih terdapat permintaan terkait sistem maka masukkan dari pengguna akan ditampung yang kemudian tahapan akan diulang ke pengkodean sistem, namun jika tidak ada masukkan dari pengguna maka tahapan dilanjutkan.

#### 7. Penggunaan sistem

Tahapan terakhir dari model *prototype* adalah menggunakan sistem dimana sistem yang telah melewati beberapa tahapan selanjutnya sistem siap untuk digunakan. Sistem yang jadi diserahkan kepada pengguna untuk selanjutnya digunakan sebagai mestinya, dengan tetap dilakukan pemeliharaan terhadap sistem sesuai dengan persetujuan dengan pengguna.

### 3.9 Pengujian Sistem

Jenis pengujian yang dilakukan pada penelitian ini adalah pengujian *black box* dan pengujian algoritma.

#### 1. Pengujian *black box*

*Black-Box Testing* merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Blackbox Testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. *Blackbox Testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. (Wintana et al., 2022)

#### 2. Pengujian algoritma

Algoritma merupakan urutan langkah-langkah penyelesaian suatu masalah, langkah-langkah tersebut berupa kumpulan perintah atau instruksi yang dibuat secara jelas dalam penyelesaian suatu masalah. Pengujian algoritma merupakan proses evaluasi untuk memastikan bahwa sebuah algoritma bekerja sesuai dengan tujuan yang diharapkan, baik dari segi akurasi, efisiensi waktu, maupun keandalan dalam berbagai kondisi *input*.

### 3.10 Perancangan Sistem

#### 3.10.1 Fitur Sistem

Sistem yang akan dikembangkan dilengkapi dengan berbagai fitur, antara lain *dashboard* yang menampilkan tanggal dan informasi setiap kegiatan, fitur penjadwalan kegiatan, pembuatan surat LDP, pengiriman surat melalui *email* kepada unit manajemen internal, notifikasi *WhatsApp* untuk sopir, serta pengaturan status sopir.

#### 3.10.2 Desain Sistem

Desain sistem ini dirancang agar setiap fitur pada sistem dapat tergambar dengan jelas dan mudah dipahami. Berikut merupakan *prototype* dari *website* penjadwalan kegiatan di Kantor Perwakilan Bank Indonesia Provinsi Sulawesi Tengah.

##### 1. Halaman *Login*



*Gambar 3. 4 Halaman Login*

Pada gambar 3.4 merupakan halaman *website* untuk *login* ke sistem taman mover bima yang terdiri dari *form* alamat *email* dan *password* dengan tombol *login* dan *link* menuju ke halaman *register*.

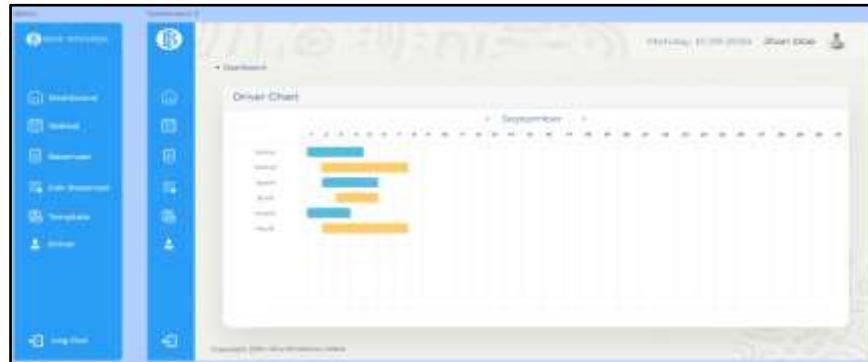
##### 2. Halaman *Register*



*Gambar 3. 5 Halaman Register*

Pada gambar 3.5 merupakan halaman *website* untuk *register* terdiri dari *form* *username*, *email address* dan *password*, dengan tombol *register* dan *link* menuju halaman *login*.

### 3. Halaman *Dashboard*



*Gambar 3. 6 Halaman Dashboard*

Pada gambar 3.6 merupakan halaman website *dashboard* sistem yang menampilkan beberapa menu seperti *dashboard*, jadwal, reservasi, cek reservasi, template, *driver* dan log out. Halaman ini menampilkan jadwal kegiatan yang telah dijadwalkan dengan memberikan visual dari kalender.

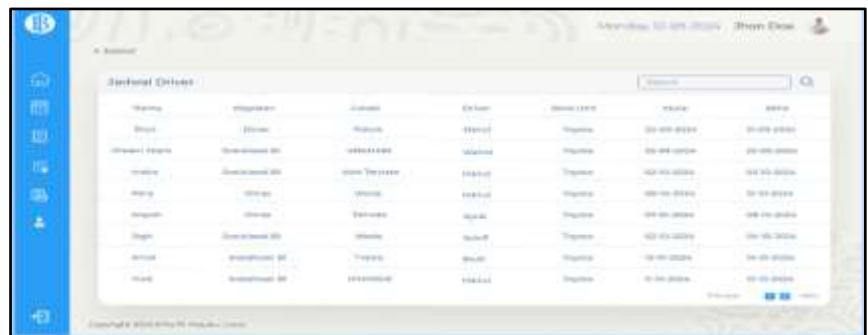
### 4. Halaman Informasi Jadwal Admin



*Gambar 3. 7 Halaman Informasi Jadwal Admin*

Pada gambar 3.7 merupakan halaman dari menu jadwal untuk *role admin* yang merupakan halaman yang menampilkan jadwal kegiatan dengan detail dari kegiatan yang dijadwalkan oleh seluruh *user* dan *admin* dapat melakukan manipulasi pada jadwal tersebut untuk dapat diubah ataupun dihapus sesuai kebutuhan.

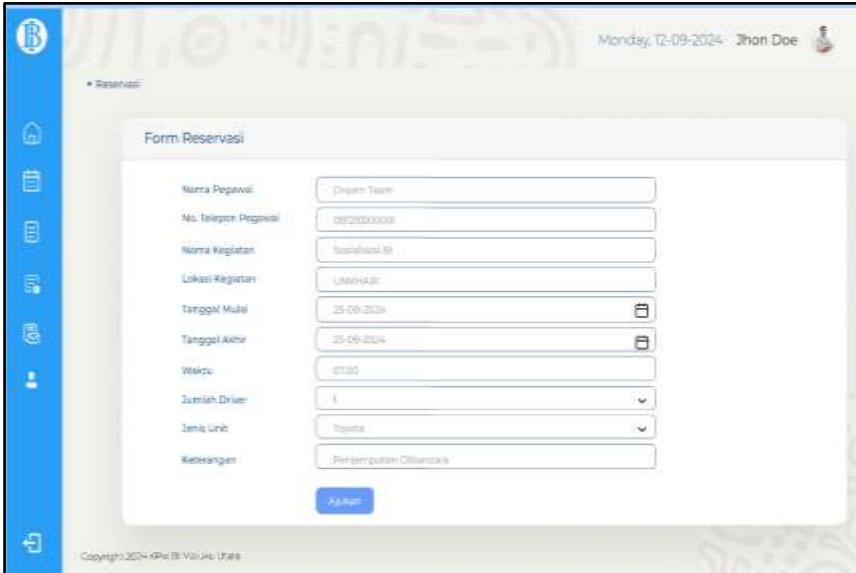
### 5. Halaman Informasi Jadwal User



*Gambar 3. 8 Halaman Informasi Jadwal User*

Pada gambar 3.8 merupakan halaman dari menu jadwal untuk *role user* yang merupakan halaman yang menampilkan jadwal kegiatan dengan detail dari kegiatan yang dijadwalkan oleh seluruh *user* berbeda dengan *role admin*, pada *user* tidak dapat melakukan manipulasi apapun.

#### 6. Halaman Penjadwalan



Monday, 12-09-2024 Jhon Doe

Form Reservasi

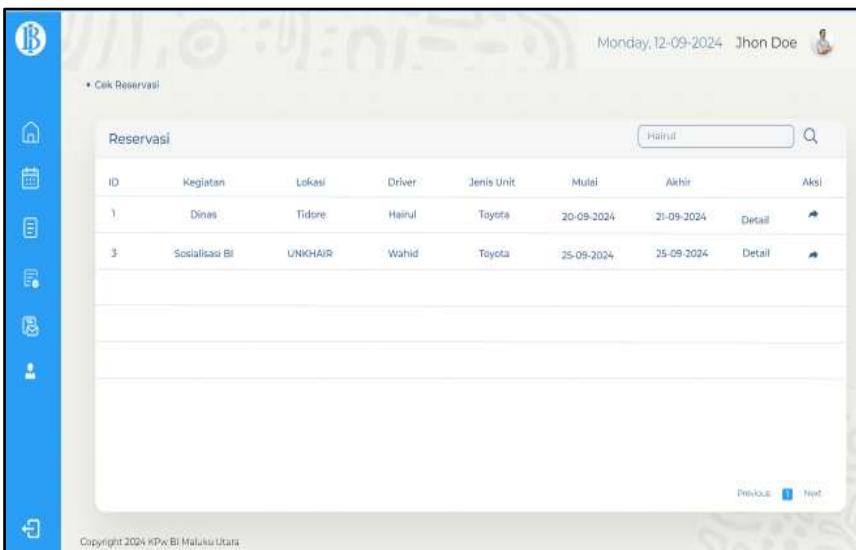
Nama Pegawai:	Dewi Tami
No. Telepon Pegawai:	08123456789
Nama Kegiatan:	Socialisasi BI
Lokasi Kegiatan:	UNIKHAIR
Tanggal Mulai:	25-09-2024
Tanggal Akhir:	25-09-2024
Waktu:	07.00
Jumlah Driver:	1
Jenis Unit:	Toyota
Keterangan:	Perjalanan Dikendara

Simpan

Gambar 3. 9 Halaman Penjadwalan

Pada gambar 3.9 merupakan halaman dari menu reservasi dimana pada halaman ini penjadwalan dilakukan dengan mengisi *form* yang ada dan memilih jumlah sopir dan unit yang akan digunakan untuk mengantarkan ke lokasi tujuan

#### 7. Halaman Informasi Jadwal



Monday, 12-09-2024 Jhon Doe

Cek Reservasi

Reservasi

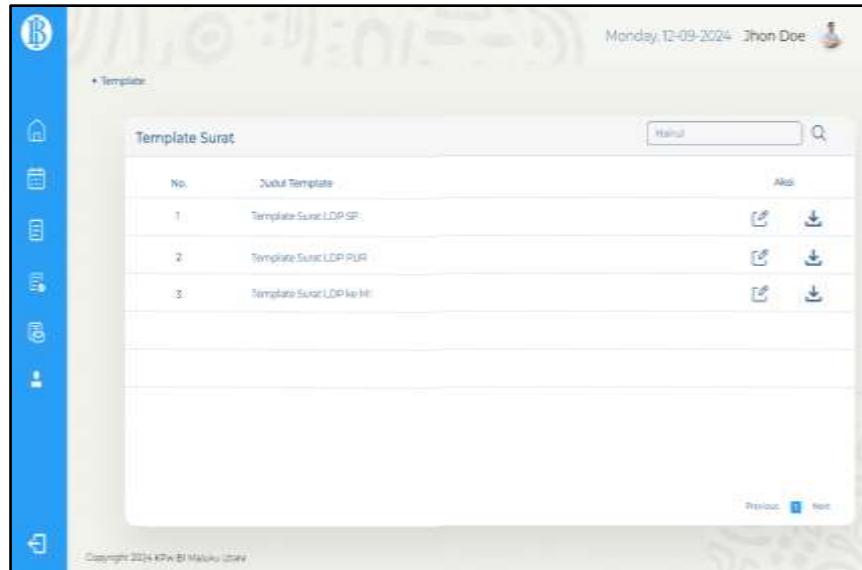
ID	Kegiatan	Lokasi	Driver	Jenis Unit	Mulai	Akhir	Aksi
1	Dinas	Tidore	Hairul	Toyota	20-09-2024	21-09-2024	Detail
3	Socialisasi BI	UNIKHAIR	Wahid	Toyota	25-09-2024	25-09-2024	Detail

Previous 1 Next

Gambar 3. 10 Halaman Informasi Jadwal

Pada gambar 3.10 merupakan halaman dari menu cek reservasi dimana pada halaman ini akan menampilkan informasi penjadwalan kegiatan yang dilakukan oleh *user* yang *login* tersebut.

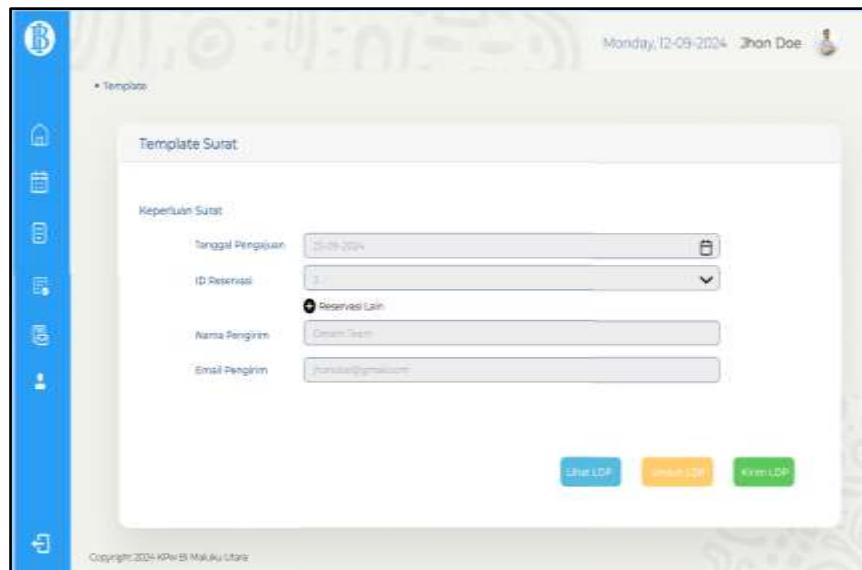
#### 8. Halaman Template Surat



*Gambar 3. 11 Halaman Template Surat*

Pada gambar 3.11 merupakan halaman dari menu template dimana pada halaman ini akan menampilkan template surat yang akan dibuat sesuai dengan keperluan surat.

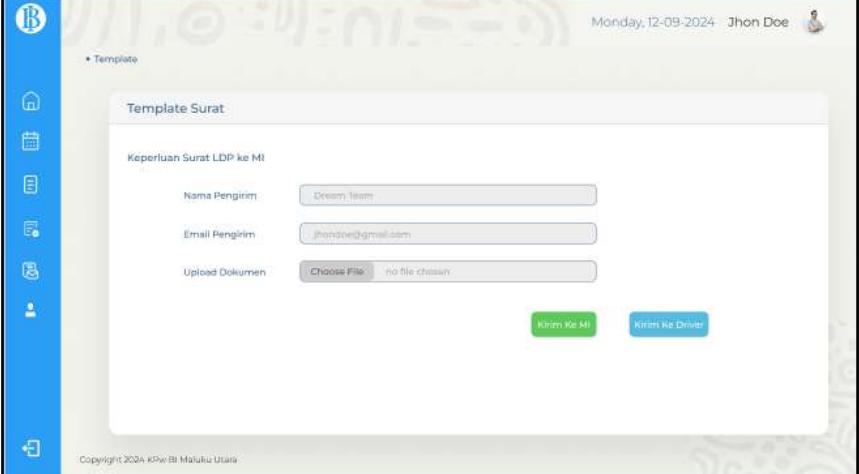
#### 9. Halaman Pembuatan Surat



*Gambar 3. 12 Halaman Pembuatan Surat*

Pada gambar 3.12 merupakan halaman pembuatan surat, halaman ini akan menampilkan *form* keperluan pembuatan surat LDP. Dengan tombol untuk melakukan unduh LDP dengan link untuk mengirim LDP.

#### 10. Halaman Pengiriman Surat



Monday, 12-09-2024 Jhon Doe

Template

Template Surat

Keperluan Surat LDP ke MI

Nama Pengirim: Dream Team

Email Pengirim: jhondoe@gmail.com

Upload Dokumen: Choose File - no file chosen

Kirim ke MI | Kirim ke Driver

Gambar 3. 13 Halaman Pengiriman Surat

Pada gambar 3.13 merupakan halaman untuk pengiriman surat LDP, dimana surat akan dikirimkan ke email unit manajemen intern dan kirim notifikasi melalui wa ke sopir.

#### 11. Halaman Driver Admin



Monday, 12-09-2024 Jhon Doe

Driver

Daftar Driver KPW BI Maluku Utara

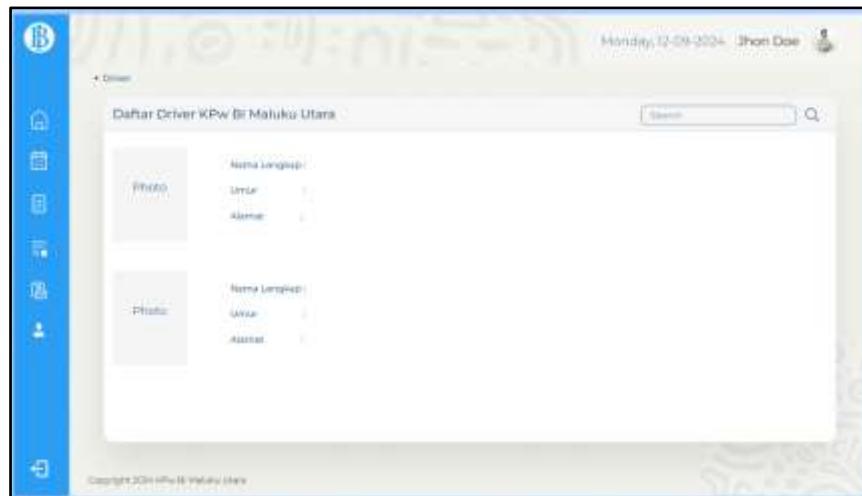
Photo	Nama Lengkap	Umur	Alamat	Action
	John Doe	25	Jl. Pemuda 123	Lihat
	Jhon Doe	25	Jl. Pemuda 123	Lihat

Copyright 2024 KPW BI Maluku Utara

Gambar 3. 14 Halaman Driver Admin

Pada gambar 3.14 merupakan halaman untuk melihat informasi terkait *driver* mulai dari foto, nama, alamat dan umur, pada halaman *driver admin* terdapat tombol yang dapat melihat apakah *driver* sedang tersedia atau tidak, dan jika *driver* sedang tidak tersedia di sistem namun *driver* tersebut telah selesai melakukan tugas maka *admin* dapat merubah status tersebut untuk dapat mengubahnya menjadi tersedia agar *user* dapat kembali menggunakan jasa *driver* tersebut.

## 12. Halaman *Driver User*



*Gambar 3. 15 Halaman Driver User*

Pada gambar 3.15 merupakan halaman untuk melihat informasi terkait *driver* mulai dari foto, nama, alamat dan umur, pada halaman *driver user* tidak terdapat aksi apapun hanya dapat melihat informasi saja.

## BAB IV

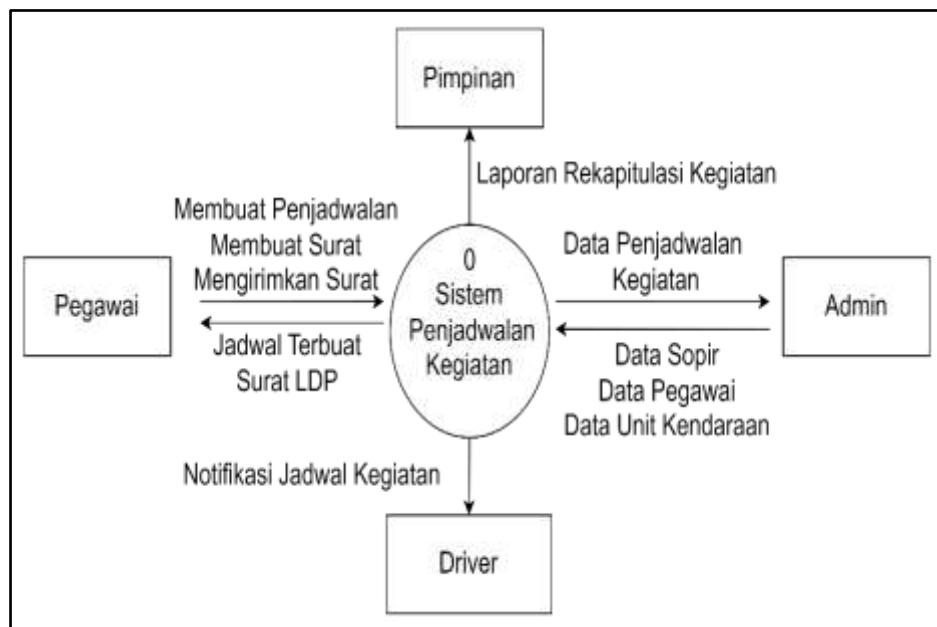
### HASIL DAN PEMBAHASAN

#### 4.1 Hasil Penelitian

##### 4.1.1 Pengembangan Sistem

###### 1. *Data Flow Diagram*

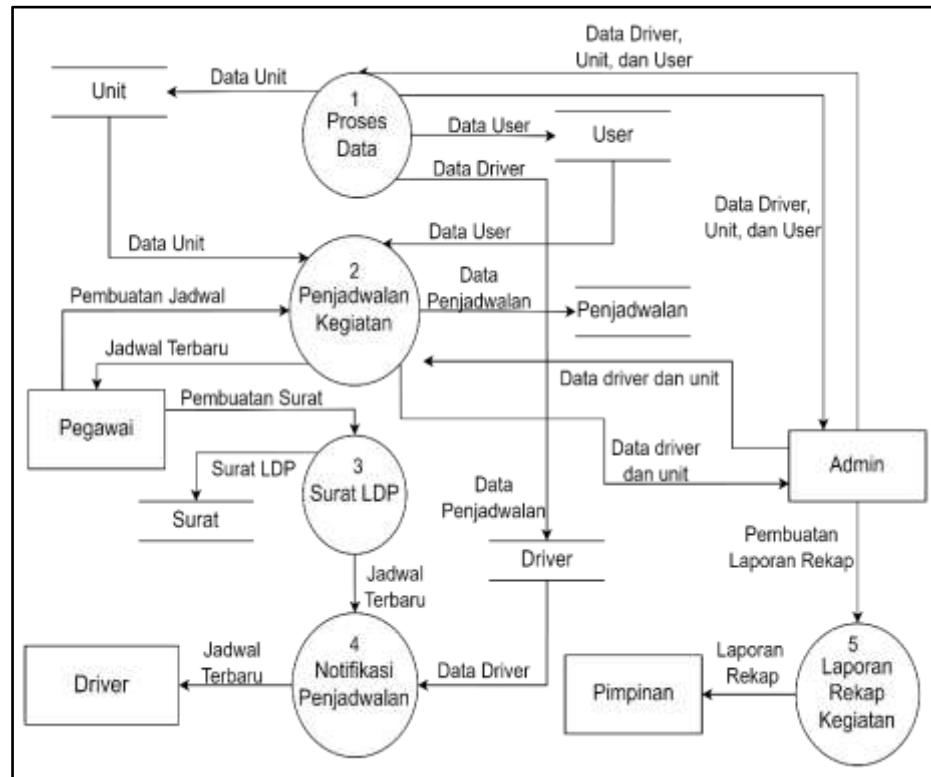
*Data Flow Diagram* (DFD) adalah *diagram* yang menggambarkan aliran data dari sebuah proses atau sistem informasi. Pada DFD, terdapat informasi terkait *input* dan *output* dari setiap proses tersebut. Pada Gambar 4.1 ditunjukkan DFD Level 0 untuk sistem penjadwalan kegiatan dengan 4 entitas eksternal, yaitu *admin*, *pegawai*, *driver*, dan *pimpinan*. *Admin* berperan dalam mengelola penjadwalan kegiatan dengan memasukkan data berupa sopir, pegawai, dan unit kendaraan. Sistem kemudian merespons dengan menghasilkan *output* berupa data penjadwalan kegiatan. Kemudian *pegawai* berperan dalam membuat penjadwalan, menyusun surat, serta mengirimkan surat melalui sistem. Selanjutnya, sistem memberikan respons berupa daftar jadwal yang berhasil dibuat serta surat LDP yang dapat diunduh oleh *pegawai*. Untuk *driver* akan menerima suatu notifikasi yang dikirimkan oleh sistem melalui *whatsapp driver* sedangkan untuk *pimpinan* akan menerima laporan rekapitulasi dari kegiatan yang telah dilakukan yang diberikan oleh *admin*.



Gambar 4. 1 DFD level 0

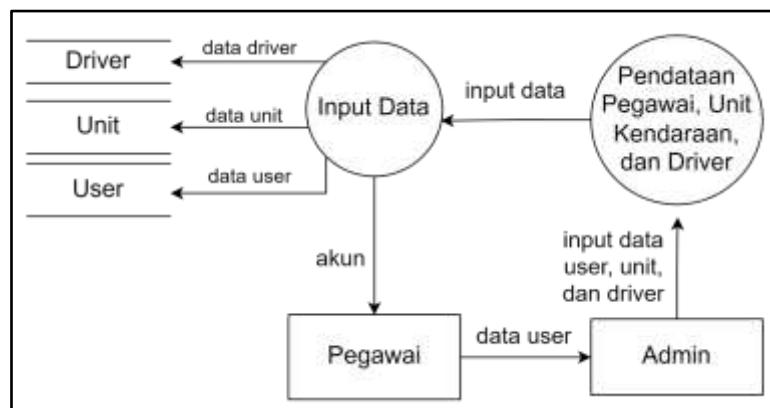
Pada Gambar 4.2 ditunjukkan DFD level 1 yang memiliki terdapat 5 proses utama, yaitu proses data, penjadwalan kegiatan, surat LDP,

notifikasi penjadwalan, dan laporan rekap kegiatan serta 4 entitas yang terlibat, yaitu *admin*, *pegawai*, *driver* dan *pimpinan*.



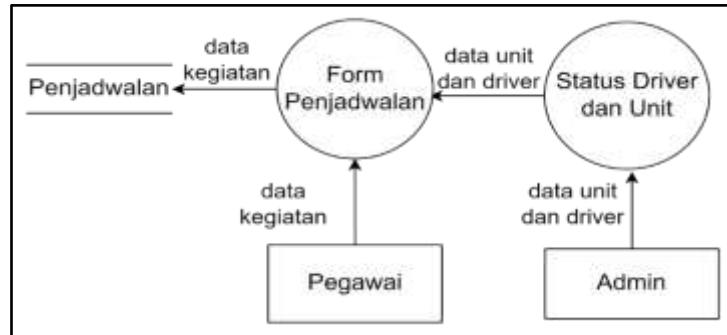
Gambar 4. 2 DFD level 1

Pada gambar 4.3 merupakan DFD level 2 untuk proses 1 yaitu proses data. Pada proses pertama merupakan proses untuk melakukan pendataan terkait data yang akan dimasukkan kedalam sistem, seperti data user yang akan diminta dan diberikan oleh pegawai yang kemudian data tersebut diterima oleh admin kemudian akan dimasukkan kedalam sistem. Setelah sistem telah menyimpan data yang dibutuhkan, maka akun akan diberikan kepada pegawai.



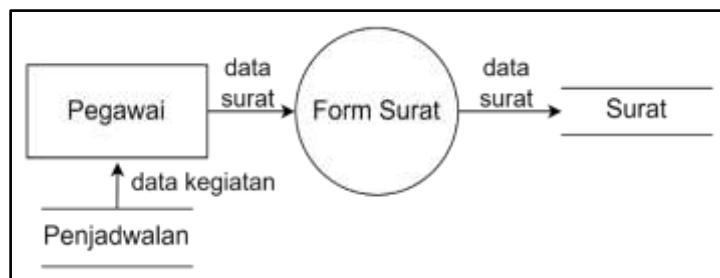
Gambar 4. 3 DFD Level 2 Proses 1

Pada gambar 4.4 merupakan DFD level 2 untuk proses 2 yaitu penjadwalan kegiatan. Pada proses kedua sebelum pegawai melengkapi *form* penjadwalan, admin perlu untuk memasukkan data dari status *driver* dan unit untuk dapat digunakan kembali oleh pegawai. Setelah itu barulah pegawai dapat mengisi *form* penjadwalan.



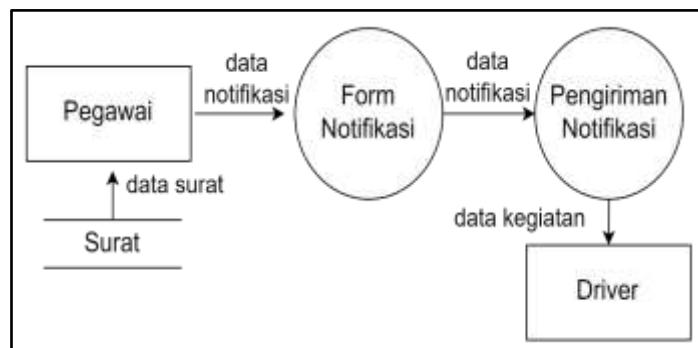
Gambar 4. 4 DFD Level 2 Proses 2

Pada gambar 4.5 merupakan DFD level 2 untuk proses 3 yaitu surat LDP. Pada proses ini pegawai mengambil data kegiatan untuk dapat melengkapi *form* surat LDP.



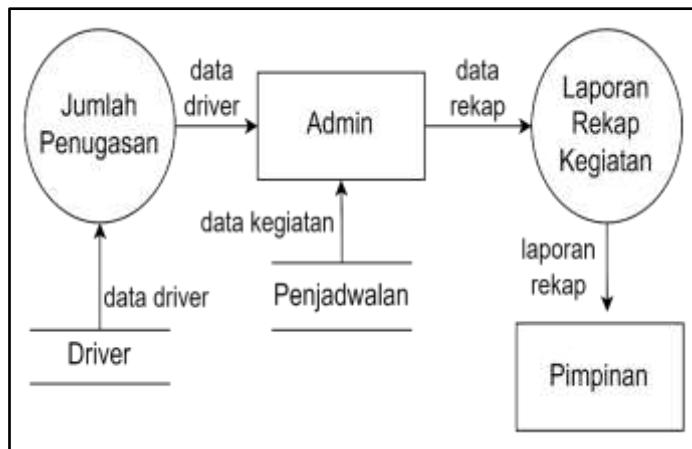
Gambar 4. 5 DFD Level 2 Proses 3

Pada gambar 4.6 merupakan DFD level 2 untuk proses 4 yaitu notifikasi penjadwalan. Setelah berhasil membuat surat LDP kemudian pegawai melengkapi *form* notifikasi dan memasukkan surat LDP sebelumnya untuk kemudian akan mengirimkan notifikasi pemberitahuan kepada *driver*



Gambar 4. 6 DFD Level 2 Proses 4

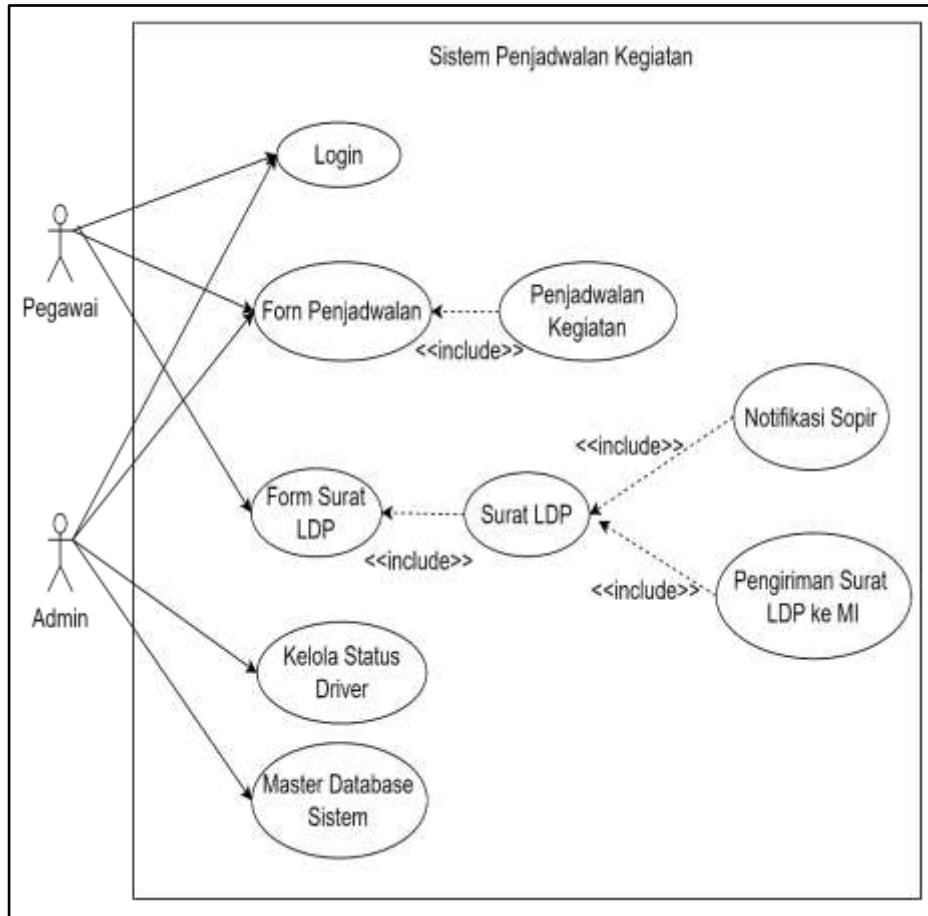
Pada gambar 4.7 merupakan DFD level 2 untuk proses 5 yaitu laporan rekap kegiatan. Pada proses ini perlu untuk mengambil data dari sistem untuk dapat membuat laporan rekapitulasi kegiatan yang kemudian akan diserahkan kepada pimpinan.



Gambar 4. 7 DFD Level 2 Proses 5

## 2. Use case diagram

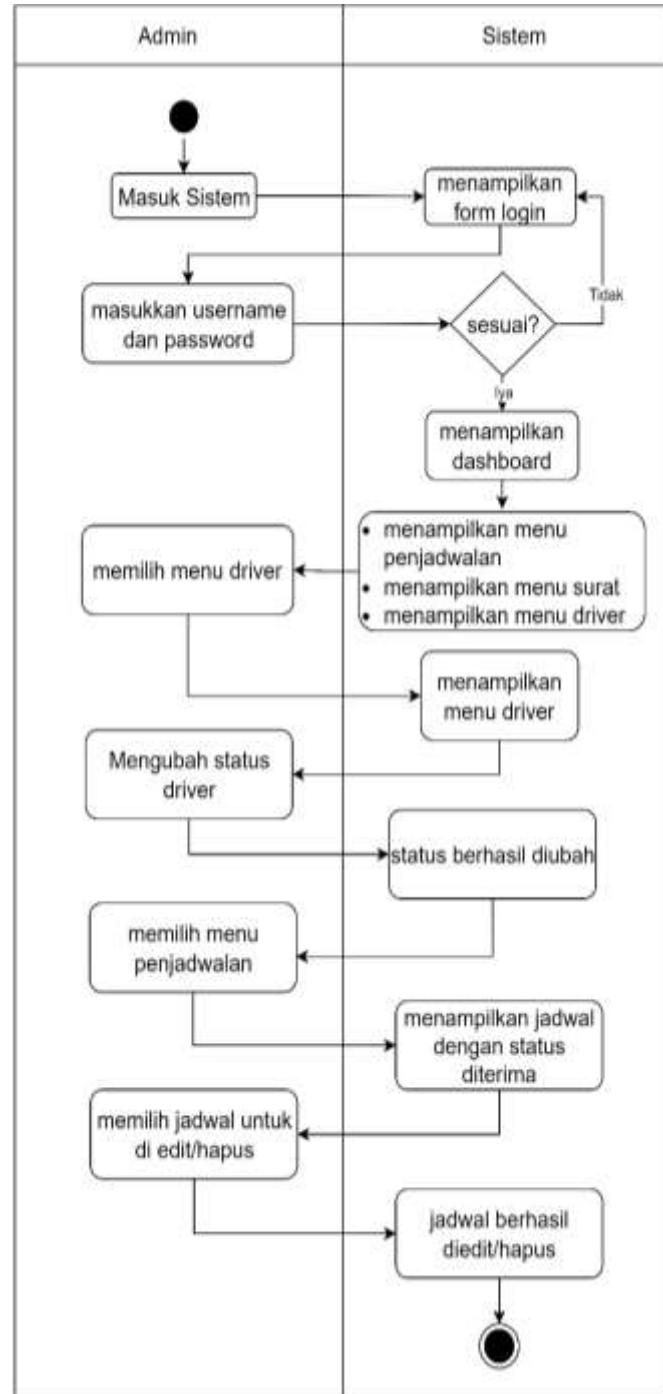
Pada gambar 4.8 di bawah ini menunjukkan terdapat dua aktor utama yang terlibat dalam sistem, yaitu *admin* dan *pegawai*. *Admin* memiliki akses untuk mengelola penjadwalan kegiatan, status sopir dan *database* sistem. Pada fitur penjadwalan kegiatan, *admin* dapat melakukan proses *edit* dan *delete*, pada fitur status sopir *admin* dapat mengubah status sopir menjadi tersedia atau tidak tersedia sedangkan pada *database* sistem *admin* dapat *edit*, *delete* dan *input* data baru. Sementara itu, *pegawai* memiliki akses untuk melakukan *login*, penjadwalan kegiatan, serta pembuatan surat LDP. Fitur *login* digunakan untuk membedakan peran pengguna, baik sebagai *pegawai* maupun *admin* yang akan masuk ke dalam sistem. Fitur penjadwalan merupakan fitur utama yang digunakan *pegawai* untuk mengajukan kegiatan dengan mengisi kebutuhan yang diperlukan dalam pembuatan surat LDP. Selanjutnya, pada fitur pembuatan surat LDP, *pegawai* dapat memilih jadwal yang telah berhasil dibuat dan berstatus diterima untuk kemudian dikirimkan kepada UMI serta sopir yang bertugas.



Gambar 4.8 Use Case Diagram

### 3. Activity diagram

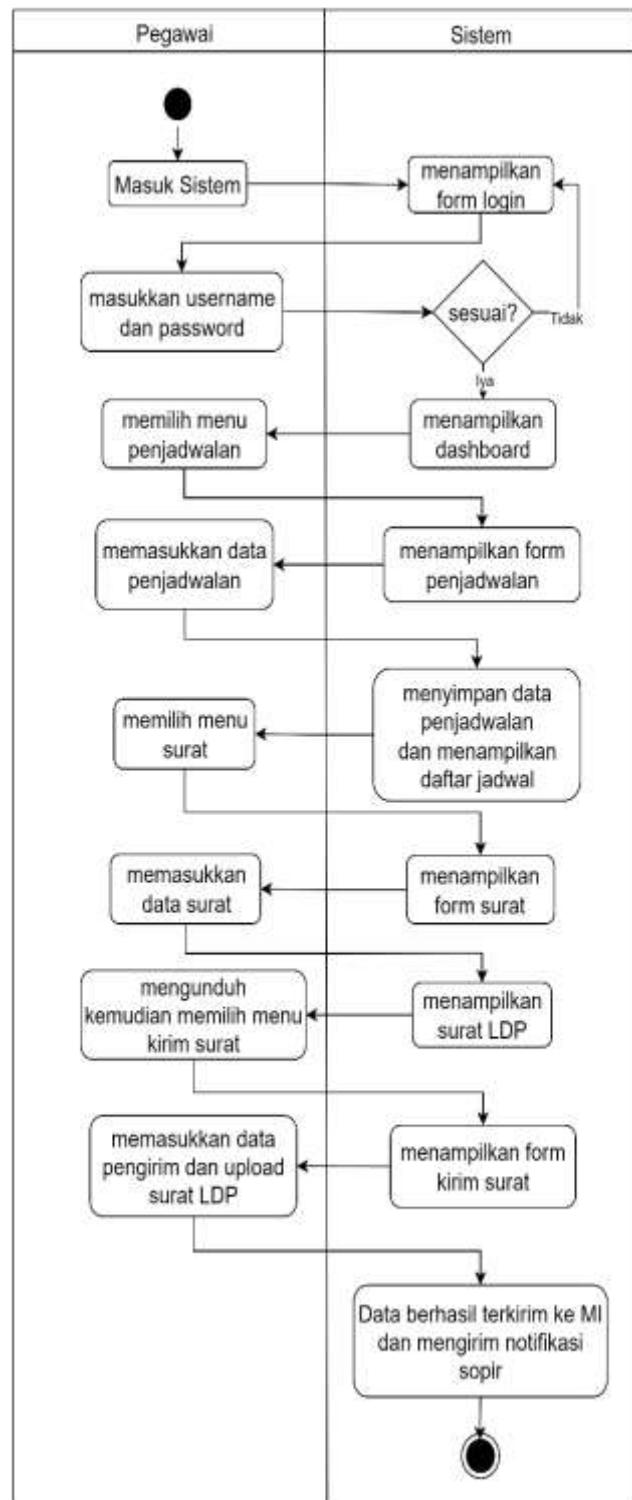
Pada Gambar 4.9 merupakan *activity diagram* untuk *admin*, alur penggunaan sistem oleh *admin* diawali dengan proses *login*, di mana sistem akan memverifikasi peran pengguna yang masuk. Setelah berhasil *login*, sistem menampilkan *dashboard* yang berisi informasi jadwal kegiatan serta menu pilihan yang tersedia. Selanjutnya, *admin* memilih menu *driver* untuk mengubah status *driver* dari tidak tersedia menjadi tersedia. Setelah status berhasil diperbarui, kemudian *admin* memilih menu penjadwalan untuk melihat daftar jadwal yang akan diubah sesuai permintaan. Kemudian, *admin* memilih jadwal yang akan dilakukan perubahan, lalu sistem akan menjalankan perintah sesuai pilihan *admin*, baik itu meng-*edit* maupun menghapus jadwal yang dipilih.



Gambar 4. 9 Activity Diagram Admin

Pada Gambar 4.10 ditampilkan *activity diagram* untuk pegawai. Proses dimulai ketika pegawai melakukan *login* dan berhasil dikenali oleh sistem. Setelah itu, sistem akan menampilkan *dashboard* dengan beberapa pilihan menu. Pegawai kemudian memilih menu penjadwalan untuk mengakses *form* penjadwalan, lalu mengisi data yang diperlukan, dan akan disimpan ke dalam sistem. Selanjutnya, pegawai memilih menu surat untuk mengakses template surat, lalu mengisi *form* surat LDP.

Setelah *form* dilengkapi, surat dapat diunduh untuk keperluan persetujuan. Apabila surat telah mendapatkan persetujuan, pegawai kembali masuk ke sistem lalu memilih menu kirim surat, melengkapi *form* pengiriman, dan sistem secara otomatis akan mengirimkan surat tersebut ke unit manajemen intern serta notifikasi kepada *driver*.

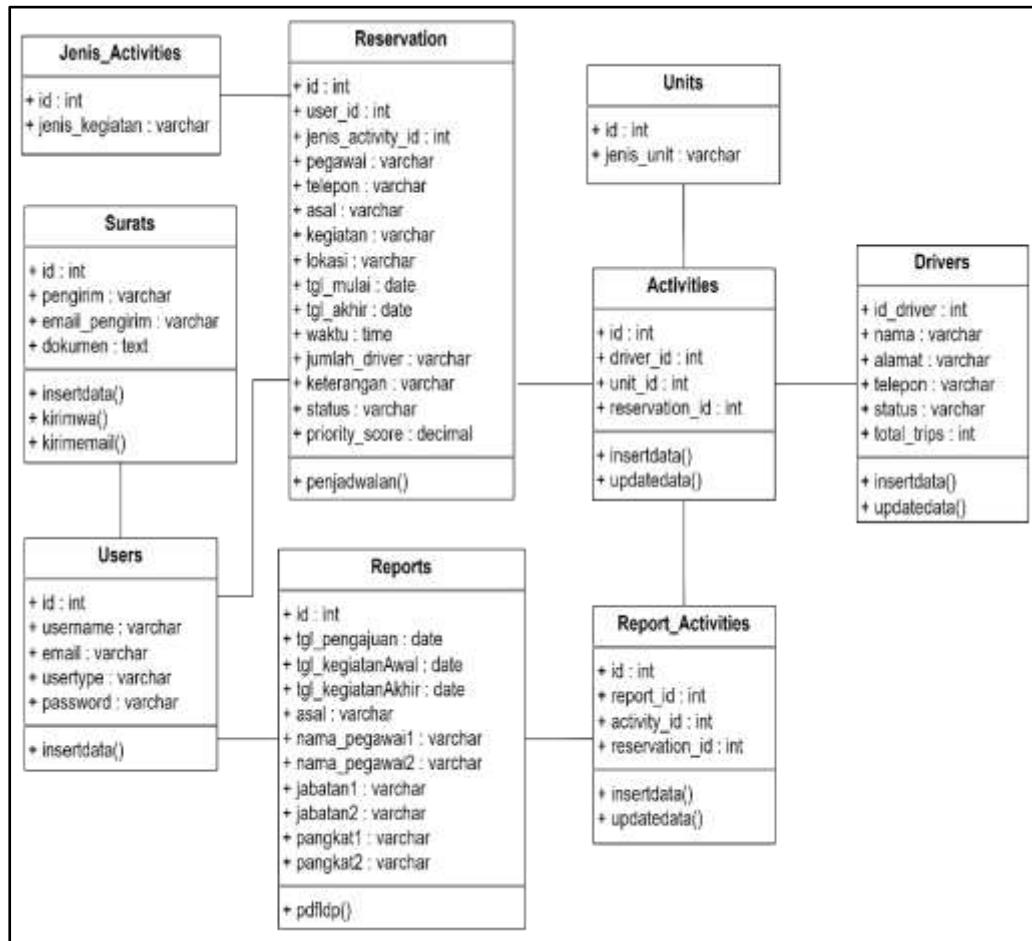


Gambar 4. 10 Activity Diagram Pegawai

#### 4. Class Diagram

Pada gambar 4.11 merupakan *class diagram* dari sistem penjadwalan kegiatan. Terdapat 9 tabel utama yang digunakan didalam sistem yang terdiri atas tabel *jenis\_activity*, *surat*, *user*, *reservation*, *report*, *unit*, *activity*, *report\_activity*, dan *driver*. Tabel *user* berfungsi untuk menyimpan informasi pengguna yang mengakses sistem, seperti *id*, *username*, *email*, *password*, dan *usertype*. Melalui tabel ini, sistem dapat mengenali identitas pengguna serta menentukan peran mereka, apakah sebagai *user* biasa atau *admin*. Selanjutnya, tabel *jenis\_activity* menyimpan data terkait jenis kegiatan yang dijadwalkan, misalnya kegiatan yang dilakukan di dalam kota atau di luar kota. Tabel ini memiliki relasi dengan tabel *reservation* untuk menentukan jenis kegiatan pada setiap penjadwalan. Tabel *unit* berisi data mengenai unit kendaraan yang akan digunakan dalam kegiatan, dengan atribut seperti *id* dan *jenis\_unit*. Kemudian tabel *driver* menyimpan *id*, nama, alamat, telepon, status dan *total\_trips* yang merupakan jumlah tugas yang telah dilakukan oleh *driver*, tabel ini juga memiliki hubungan dengan tabel *reservation* untuk menentukan sopir yang bertugas pada suatu kegiatan. Tabel *reservation* yang menjadi inti dari sistem penjadwalan. Tabel ini menyimpan berbagai atribut penting seperti *id*, *user\_id*, *jenis\_activity\_id*, pegawai, telepon, asal, kegiatan, lokasi, *tgl\_mulai*, *tgl\_akhir*, waktu, jumlah\_driver, keterangan, status, dan *priority\_score*. Melalui tabel ini, sistem melakukan proses penjadwalan dengan memanggil relasi dari tabel *user* dan *jenis\_activity* untuk melengkapi data kegiatan. Serta menyimpan skor dari bobot suatu penjadwalan untuk dapat dibandingkan. Selanjutnya, tabel *activity* berfungsi sebagai tabel jembatan (*pivot table*) yang menghubungkan tiga tabel utama, yaitu *reservation*, *unit*, dan *driver*. Dalam *framework* Laravel, konsep ini dikenal sebagai *eloquent relationship*, di mana hubungan antar tabel yang bersifat *many-to-many* memerlukan tabel perantara agar data antar entitas dapat saling terhubung dengan baik. Tabel *report* digunakan untuk menyimpan data laporan pembuatan surat LDP, atribut yang disimpan meliputi *id*, *tgl\_pengajuan*, *tgl\_kegiatanAwal*, *tgl\_kegiatanAkhir*, asal, *nama\_pegawai1*, *nama\_pegawai2*, *jabatan1*, *jabatan2*, *pangkat1*, dan *pangkat2*. Data pegawai yang dicatat dalam tabel ini merupakan pihak yang akan menandatangani surat LDP. Tabel *report\_activity* memiliki fungsi serupa dengan tabel *activity*, yakni sebagai tabel penghubung (*pivot table*) yang berisi relasi *id\_activity*, *id\_report*, dan *id\_reservation* yang digunakan untuk mengaitkan laporan dengan kegiatan tertentu. Terakhir, tabel *surat* menyimpan data mengenai proses penyuratan yang dilakukan oleh sistem. Atributnya meliputi *id*, *pengirim*, *email\_pengirim*, dan *dokumen*. Tabel ini berperan dalam proses pengiriman surat secara

otomatis melalui fungsi kirimwa dan kirimemail, sehingga mempermudah distribusi dokumen secara digital.



Gambar 4. 11 Class Diagram

#### 4.1.2 Algoritma Priority Queue

##### 1. Data Kegiatan

Berikut adalah data kegiatan yang akan diolah untuk dapat memberikan urutan jadwal sesuai dengan prioritas yang telah ditentukan. Proses pengurutan jadwal ini akan membutuhkan suatu kriteria yang menjadi prioritas untuk menjadi pembanding antar jadwal yang dibuat.

Tabel 4. 1 Data Kegiatan BI Sulteng

Nama Kegiatan	Jabatan	Tanggal Mulai	Tanggal Akhir	Jenis Kegiatan
Kegiatan Sosialisasi CBP Rupiah	Unit PUR	31/08/2025	31/08/2025	Luar Kota
Dialog dan Lokakarya	Unit FPKP	05/08/2025	05/08/2025	Dalam Kota

BI Goes To Campus	Unit PUR	30/08/2025	30/08/2025	Dalam Kota
Pengantar Narasumber	Unit Humas	31/08/2025	31/08/2025	Dalam Kota
Kuliah Umum	Kepala Perwakilan BI Sulteng	16/08/2025	16/08/2025	Dalam Kota
Kunjungan Situs Kakao	Kepala Perwakilan BI Sulteng	02/08/2025	02/08/2025	Luar Kota
BI-GTC Perluasan Digitalisasi	Kepala Perwakilan BI Sulteng	30/08/2025	30/08/2025	Luar Kota
Pekan QRIS Nasional	Kepala Perwakilan BI Sulteng	15/08/2025	15/08/2025	Dalam Kota
QRIS Jelajah	Manager SP	19/08/2025	20/08/2025	Luar Kota
Sosialisasi QRIS MMS	Unit SP	16/08/2025	16/08/2025	Dalam Kota
Liaison Bank	Kepala Perwakilan BI Sulteng	06/08/2025	07/08/2025	Dalam Kota
Survey Lokasi BI Go To Campus	Unit Humas	28/08/2025	28/08/2025	Dalam Kota

## 2. Perhitungan metode AHP

Berdasarkan data kegiatan untuk menentukan prioritas suatu penjadwalan maka diperlukan kriteria, adapun kriteria yang dibutuhkan adalah jabatan, tenggat waktu, dan jenis kegiatan, dari ketiga kriteria tersebut perlu diberikan bobot. Proses pembobotan menggunakan perhitungan AHP untuk memperoleh bobot tiap kriteria berdasarkan tingkat kepentingan yang kemudian hasil dari pembobotan kriteria akan menghasilkan skor prioritas. Skor prioritas inilah yang kemudian dijadikan kriteria prioritas untuk dapat mengurutkan jadwal berdasarkan prioritasnya.

### a. Menghitung perbandingan prioritas

*Tabel 4. 2 Tabel Matriks Perbandingan Berpasangan*

Kriteria	C1	C2	C3
Jabatan	1	3	5
Tenggat Waktu	1/3	1	2
Jenis Kegiatan	1/5	1/2	1

*Tabel 4. 3 Nilai Desimal Matriks Perbandingan Berpasangan*

Kriteria	C1	C2	C3
Jabatan	1	3	5

Tenggat Waktu	0,33	1	2
Jenis Kegiatan	0,2	0,5	1
Total	1,53	4,5	8

Tabel 4. 4 Normalisasi Matriks Perbandingan Berpasangan

Kriteria	C1	C2	C3	Total
Jabatan	0,6535	0,6666	0,625	0,648
Tenggat Waktu	0,2156	0,2222	0,25	0,230
Jenis Kegiatan	0,1307	0,1111	0,125	0,122

b. Hasil Bobot Prioritas

Tabel 4. 5 Hasil Bobot Prioritas AHP

Kriteria	Bobot
Jabatan	0,648
Tenggat Waktu	0,230
Jenis Kegiatan	0,122

c. Hitung Nilai  $\lambda_{max}$

Tabel 4. 6 Hasil perhitungan Kriteria

Kriteria	Jumlah Kolom Matriks Awal	Bobot Prioritas	Hasil
Jabatan	1,533	0,648	0,993384
Tenggat Waktu	4,500	0,230	1,035000
Jenis Kegiatan	8.000	0,122	0,976000

$$\lambda_{max} = 0,993384 + 1,035000 + 0,976000 = 3,004384$$

d. Hitung Consistency Index (CI)

$$CI = \frac{\lambda_{max} - n}{n - 1} = \frac{3,004384 - 3}{3 - 1} = \frac{0,004384}{2} = 0,002192$$

e. Hitung Consistency Ratio (CR)

$$CR = \frac{CI}{IR} = \frac{0,002192}{0,58} = 0,00378$$

Berdasarkan hasil perhitungan Consistency Ratio, diperoleh nilai CR < 0,10 yang menunjukkan bahwa perbandingan antar kriteria dalam matriks AHP bersifat konsisten. Dengan demikian, bobot prioritas yang diperoleh dari perhitungan dapat diterima dan digunakan dalam proses perhitungan selanjutnya.

f. Skor Jabatan

Tabel 4. 7 Tabel Skor Jabatan

Nama Jabatan	Skor
Kepala Perwakilan BI Sulteng	4

Deputi	3
Manager	2
Fungsi Perumusan KEKDA Provinsi	1
Fungsi Data dan Statistik Ekonomi dan Keuangan	1
Fungsi Pelaksanaan Pengembangan UMKM, KI Dan Syaria	1
Fungsi Logistik, SDM, Sekretariat, Pengamanan dan Protokol	1
Unit Manajemen Intern	1
Unit Implementasi PUR	1
Unit Implementasi SP	1
Unit Kehumasan	1
Unit Layanan dan Pengelolaan Uang	1
Unit Keuangan	1

g. Skor Tenggat Waktu

*Tabel 4. 8 Tabel Skor Tenggat Waktu*

Rentang Waktu	Skor
<=1 hari	3
<=3 hari	2
>3 hari	1

h. Skor Jenis Kegiatan

*Tabel 4. 9 Tabel Skor Jenis Kegiatan*

Jenis Kegiatan	Skor
Dalam Kota	2
Luar Kota	1

#### 4.1.3 Implementasi Sistem

##### 1. Implementasi Perangkat Keras dan Perangkat Lunak

Perangkat keras yang digunakan dalam pengembangan sistem ini terdiri dari sebuah laptop Lenovo ThinkBook 14 G2 ARE dengan prosesor AMD Ryzen 5 4500U with Radeon Graphics 2.38 GHz. Laptop ini didukung oleh RAM sebesar 8 GB serta SSD berkapasitas 477 GB untuk penyimpanan data.

Perangkat lunak yang digunakan mencakup sistem operasi Windows 11, code editor Visual Studio Code, serta XAMPP sebagai server lokal.

## 2. Implementasi Basis Data

### a. Tabel *Reservation*

Tabel *reservation* berfungsi untuk menyimpan data-data dari pegawai yang melakukan penjadwalan terhadap kegiatan. Berikut adalah struktur dari tabel *reservation*:

*Tabel 4. 10 Tabel Reservation*

Field	Type	Length/Values	Index
Id	bigint	20	Primary Key
User_id	bigint	20	Foreign Key
Jenis_activity_id	bigint	20	Foreign Key
Pegawai	varchar	255	
Telepon	varchar	13	
Kegiatan	varchar	255	
Lokasi	varchar	255	
Tgl_mulai	date		
Tgl_akhir	date		
Waktu	time		
Jumlah_driver	int	20	
Keterangan	varchar	255	

### b. Tabel *Report*

Tabel *report* berfungsi untuk menyimpan data-data dari pegawai yang melakukan pembuatan surat LDP. Berikut adalah struktur dari tabel *report*:

*Tabel 4. 11 Tabel Report*

Field	Type	Length/Values	Index
Id	bigint	20	Primary Key
Tgl_pengajuan	date		
Tgl_kegiatanAwal	date		
Tgl_KegiatanAkhir	date		
Asal	varchar	255	
Nama_pegawai1	varchar	255	
Nama_pegawai2	varchar	255	
Jabatan1	varchar	255	
Jabatan2	varchar	255	
Pangkat1	varchar	255	
Pangkat2	varchar	255	

### c. Tabel Surat

Tabel surat berfungsi untuk menyimpan data-data surat ldp yang dikirimkan ke UMI. Berikut adalah struktur dari tabel surat:

*Tabel 4. 12 Tabel Surat*

Field	Type	Length/Values	Index
Id	bigint	20	Primary Key
Pengirim	varchar	255	
Email pengirim	varchar	255	
dokumen	text		

d. Tabel *driver*

Tabel *driver* berfungsi untuk menyimpan data-data sopir yang ada di kantor. Berikut adalah struktur dari tabel *driver*:

*Tabel 4. 13 Tabel Driver*

Field	Type	Length/Values	Index
Id	bigint	20	Primary Key
Nama	varchar	255	
Alamat	varchar	255	
Telepon	varchar	12	
Status	varchar	255	

e. Tabel *unit*

Tabel *unit* berfungsi untuk menyimpan data-data unit kendaraan yang ada di kantor. Berikut adalah struktur dari tabel *unit*:

*Tabel 4. 14 Tabel Unit*

Field	Type	Length/Values	Index
Id	bigint	20	Primary Key
Jenis	varchar	255	

f. Tabel *user*

Tabel *user* berfungsi untuk menyimpan data-data pegawai yang ada di kantor. Berikut adalah struktur dari tabel *user*:

*Tabel 4. 15 Tabel User*

Field	Type	Length/Values	Index
Id	bigint	20	Primary Key
Username	varchar	255	
Email	varchar	255	
Usertype	varchar	255	
Password	varchar	255	

g. Tabel *activity*

Tabel *activity* berfungsi untuk menjadi tabel penghubung data antara tabel *unit*, *driver* dan *reservation*. Berikut adalah struktur dari tabel *activity*:

*Tabel 4. 16 Tabel Activity*

Field	Type	Length/Values	Index
Id	bigint	20	Primary Key
Driver_id	bigint	20	Foreign Key

Unit id	bigint	20	Foreign Key
Reservation id	bigint	20	Foreign Key

h. Tabel jenis *activity*

Tabel jenis *activity* berfungsi untuk menyimpan data dari jenis aktifitas yang akan ditugaskan kepada sopir. Berikut adalah struktur dari tabel jenis *activity*:

*Tabel 4. 17 Tabel Jenis Activity*

Field	Type	Length/Values	Index
Id	bigint	20	Primary Key
Jenis kegiatan	varchar	255	

i. Tabel *report activity*

Tabel *report activity* berfungsi untuk menyimpan data-data dari penjadwalan yang berhubungan dengan surat LDP yang dibuat. Berikut adalah struktur dari tabel *report activity*:

*Tabel 4. 18 Tabel Report Activity*

Field	Type	Length/Values	Index
Id	bigint	20	Primary Key
Report id	bigint	20	Foreign Key
Activity id	bigint	20	Foreign Key
Reservation id	varchar	255	

3. Implementasi Desain Sistem

a. Halaman *Login*

Pada gambar 4.12 merupakan tampilan awal dari sistem penjadwalan, dimana pengguna perlu untuk memasukkan email dan *password* yang sesuai untuk dapat masuk kedalam sistem sekaligus untuk melakukan verifikasi *role* apakah pengguna merupakan *admin* atau *user* biasa.



*Gambar 4. 12 Halaman Login*

b. Halaman *Register*

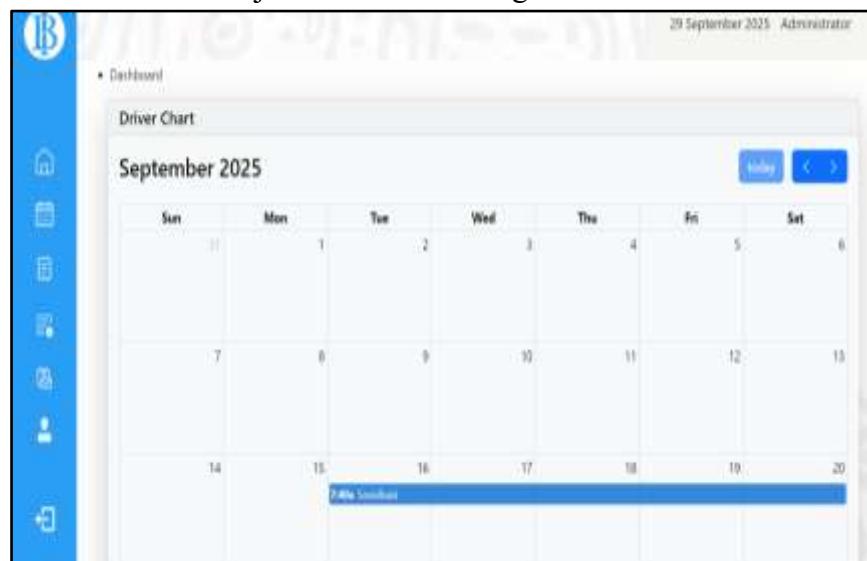
Pada gambar 4.13 merupakan halaman untuk pengguna yang belum memiliki akses kedalam sistem untuk dapat membuat akun terlebih dahulu untuk kemudian dapat menggunakan sistem dengan *role* default yaitu *user*.



*Gambar 4. 13 Halaman Register*

c. Halaman *Dashboard*

Pada gambar 4.12 merupakan halaman untuk melihat jadwal dari kegiatan yang telah atau sedang dilakukan. Tampilan yang diberikan merupakan tampilan kalender sehingga dapat memudahkan pengguna melihat bulan dan hari jadwal dari suatu kegiatan.



*Gambar 4. 14 Halaman Dashboard*

d. Halaman Informasi Jadwal *Admin*

Pada gambar 4.15 merupakan halaman informasi jadwal untuk *role* *admin*, dimana *admin* dapat melihat seluruh jadwal yang telah

dimasukkan oleh seluruh *user* dan bisa melakukan manipulasi terhadap jadwal tersebut baik itu akan dihapus maupun diubah isinya.

Jadwal Driver									
Nama	Kegiatan	Lokasi	Driver	Jenis Unit	Jenis Kegiatan	Waktu	Mulai	Akhir	Aksi
Indra	Sosialisasi	Tidore	Rival Taher	Toyota Innova	Didalam Kota	01:00 AM	2025-09-26	2025-09-30	
Indra	Sosialisasi	Tidore	Iwan Senen	Toyota Hilux	Didalam Kota	01:00 AM	2025-09-26	2025-09-30	
Indra	Sosialisasi	Tidore	Iwan Senen	Ramise	Didalam Kota	07:40 AM	2025-09-16	2025-09-20	

Gambar 4. 15 Halaman Informasi Jadwal Admin

#### e. Halaman Informasi Jadwal User

Pada gambar 4.16 merupakan halaman informasi jadwal *user*, dimana *user* akan melihat seluruh jadwal yang telah dimasukkan oleh seluruh pengguna sistem, namun berbeda dengan *admin* pada halaman *user* hanya bisa melihat tanpa bisa memanipulasi jadwal.

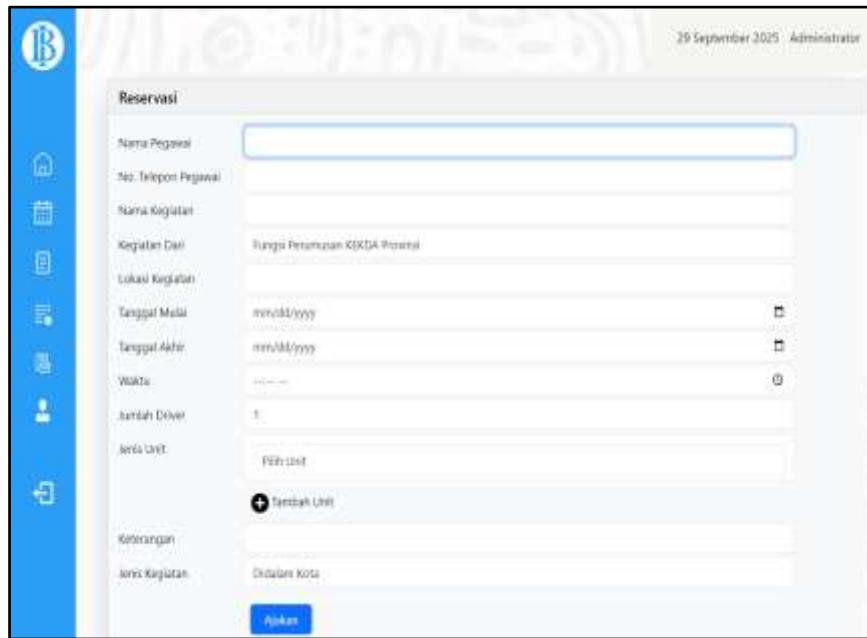
Jadwal Driver									
Nama	Telepon	Kegiatan	Lokasi	Driver	Jenis Unit	Jenis Kegiatan	Waktu	Mulai	Akhir
Indra	082134897467	Sosialisasi	Tidore	Rival Taher	Toyota Innova	Didalam Kota	01:00 AM	2025-09-26	2025-09-30
Indra	082134897467	Sosialisasi	Tidore	Iwan Senen	Toyota Hilux	Didalam Kota	01:00 AM	2025-09-26	2025-09-30
Indra	082134897467	Sosialisasi	Tidore	Iwan Senen	Ramise	Didalam Kota	07:40 AM	2025-09-16	2025-09-20

Gambar 4. 16 Halaman Informasi Jadwal User

#### f. Halaman Penjadwalan

Pada gambar 4.17 merupakan halaman penjadwalan kegiatan, dimana pengguna akan memasukkan hal-hal yang dibutuhkan dalam penjadwalan kemudian setelah memastikan bahwa semua terisi dan

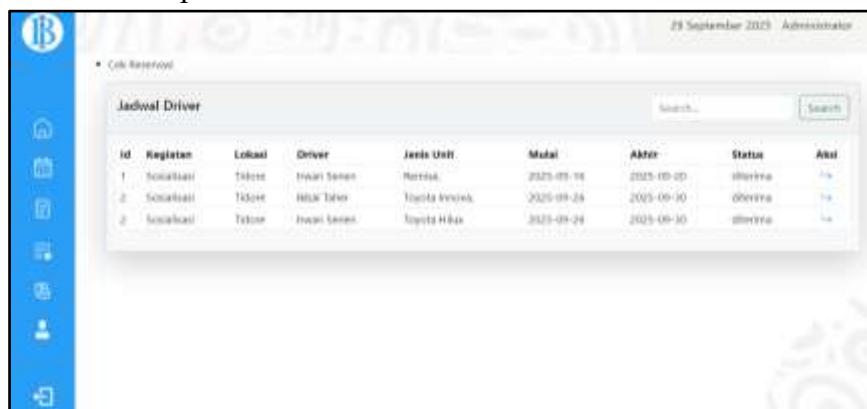
melihat tidak ada kesalahan dalam penjadwalan maka bisa langsung diajukan.



Gambar 4. 17 Halaman Penjadwalan

g. Halaman Informasi Jadwal

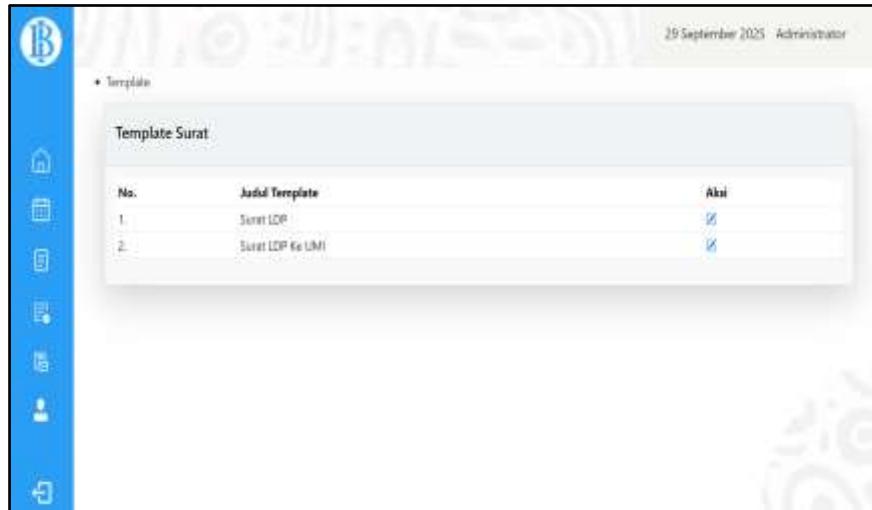
Pada gambar 4.18 merupakan halaman khusus untuk melihat jadwal yang telah diajukan oleh *user* yang masuk tersebut, pada halaman ini jadwal dari *user* lain tidak akan muncul melainkan hanya jadwal yang telah dibuat oleh pemilik akun.



Gambar 4. 18 Halaman Informasi Jadwal

h. Halaman *Template* Surat

Pada gambar 4.19 merupakan halaman untuk melihat template surat yang akan dibuat dan melihat menu untuk mengirim surat ke UMI dan *driver*.



Gambar 4. 19 Halaman Template Surat

i. Halaman Pembuatan Surat

Pada gambar 4.20 merupakan halaman pembuatan surat, dimana pengguna akan memasukkan hal-hal yang dibutuhkan dalam membuat surat LDP kemudian setelah memastikan bahwa semua terisi dan melihat tidak ada kesalahan dalam pembuatan surat maka bisa langsung dilihat dan diunduh.

Gambar 4. 20 Halaman Pembuatan Surat

j. Halaman Pengiriman Surat

Pada gambar 4.21 merupakan halaman untuk mengirimkan surat ke email UMI dan memberikan notifikasi melalui whatsapp ke *driver* yang bertugas. Sebelum mengirimkan pastikan untuk surat LDP telah di setujui oleh manajer dan deputi sebelumnya.



*Gambar 4. 21 Halaman Pengiriman Surat*

k. Halaman *Driver Admin*

Pada gambar 4.22 merupakan halaman untuk melihat infromasi dari *driver*, dimana pada halaman *admin* terdapat tombol untuk melihat apakah status *driver* sedang tidak bertugas atau dalam penugasan apabila tombol masih dalam keadaan tidak tersedia berarti *driver* sedang bertugas namun jika *driver* sudah kembali maka *admin* perlu untuk menekan tombol untuk mengembalikan status *driver* menjadi tersedia atau sedang tidak bertugas



Gambar 4. 22 Halaman Driver Admin

## 1. Halaman *Driver User*

Pada gambar 4.23 merupakan halaman untuk melihat informasi dari *driver*, namun pada halaman *user* tidak ada tombol apapun dan hanya bisa melihat informasi *driver* saja.

Gambar 4. 23 Halaman Driver User

#### 4.1.4 Pengujian Sistem

##### 1. Pengujian Blackbox

Pengujian *Blackbox* dilakukan untuk memastikan aplikasi dapat berjalan dengan baik dan menghasilkan *output* sesuai yang diharapkan dengan model pengujian yang dilakukan pada bagian-bagian fungsi dari sistem.

##### a. Pengujian Halaman Admin

Tabel 4. 19 Pengujian Halaman Admin

Fitur	Output	Hasil
<i>Login</i> menggunakan <i>username</i> dan <i>password</i> yang terdaftar	<i>Login</i> berhasil dan masuk kedalam <i>dashboard</i>	Fitur bekerja dengan baik
<i>Login</i> menggunakan <i>username</i> dan <i>password</i> yang tidak sesuai	<i>Login</i> gagal dan akan dikembalikan ke halaman <i>login</i>	Fitur bekerja dengan baik
Logout	Berhasil keluar dari halaman utama sistem	Fitur bekerja dengan baik
<i>Admin</i> dapat melakukan update data penjadwalan	Sistem menampilkan halaman update dan dapat mengubah data penjadwalan	Fitur bekerja dengan baik
<i>Admin</i> dapat menghapus data penjadwalan yang telah ada	Sistem menghapus data penjadwalan yang telah ada	Fitur bekerja dengan baik
<i>Admin</i> dapat mengubah status <i>driver</i> dari tidak tersedia menjadi tersedia menjadi tersedia	Sistem mengubah status <i>driver</i> dari tidak tersedia menjadi tersedia dan mengubah warna tombol dari merah ke hijau	Fitur bekerja dengan baik

b. Pengujian Halaman *User*

*Tabel 4. 20 Pengujian Halaman User*

Fitur	Output	Hasil
<i>Login</i> menggunakan <i>username</i> dan <i>password</i> yang terdaftar	<i>Login</i> berhasil dan masuk kedalam <i>dashboard</i>	Fitur bekerja dengan baik
<i>Login</i> menggunakan <i>username</i> dan <i>password</i> yang tidak sesuai	<i>Login</i> gagal dan akan dikembalikan ke halaman <i>login</i>	Fitur bekerja dengan baik
Logout	Berhasil keluar dari halaman utama sistem	Fitur bekerja dengan baik
<i>User</i> melakukan penjadwalan sopir	Penjadwalan berhasil dibuat dan akan pindah ke halaman daftar pribadi	Fitur bekerja dengan baik
<i>User</i> melakukan pengecekan jumlah sopir dan unit	Sistem menampilkan jumlah sopir dan unit yang tersedia saat ini	Fitur bekerja dengan baik
<i>User</i> tidak melengkapi data penjadwalan	Sistem memberikan pemberitahuan terkait data yang perlu dilengkapi	Fitur bekerja dengan baik
<i>User</i> mengisi data surat LDP	Surat berhasil terbuat dan menampilkan surat LDP yang dapat diunduh	Fitur bekerja dengan baik
<i>User</i> tidak melengkapi data surat LDP	Sistem memberikan pemberitahuan terkait data yang perlu dilengkapi	Fitur bekerja dengan baik
<i>User</i> melengkapi data pengiriman surat dan mengirimkan surat	Surat berhasil terkirim	Fitur bekerja dengan baik
<i>User</i> tidak melengkapi data pengiriman surat	Sistem memberikan pemberitahuan terkait data yang perlu dilengkapi	Fitur bekerja dengan baik
Surat terkirim melalui email ke unit MI dan notifikasi WA ke sopir	Surat berhasil terkirim melalui email dan whatsapp	Fitur bekerja dengan baik

2. Pengujian Algoritma

Simulasi dilakukan untuk mengetahui bagaimana waktu eksekusi dari algoritma priority queue jika dibandingkan dengan algoritma first come

first serve yang merupakan algoritma sederhana dalam suatu penjadwalan.

*Tabel 4. 21 Pengujian Perbandingan Algoritma*

Pengujian ke-	Data ke-	Waktu Eksekusi FCFS	Waktu Eksekusi PQ
1	50	2.5139 detik	2,52 detik
2	50	2.5162 detik	2,843 detik
3	50	2.5443 detik	2,7619 detik
4	50	2.4064 detik	2,9555 detik
Rata-rata		2,4952 detik	2,7701 detik

### 3. Perhitungan *Throughput*

Berdasarkan pengujian yang dilakukan dengan keberhasilan algoritma priority queue dalam menyelesaikan tugas rata-rata waktunya di 2,7701 detik maka selanjutnya dilakukan perhitungan *throughput* untuk mengukur peningkatan efisiensi dari algoritma FCFS.

$$\text{Throughput} \equiv \frac{\text{mean waktu PQ} - \text{mean waktu FCFS}}{\text{mean waktu FCFS}} \times 100\%$$

$$\text{Throughput} \equiv \frac{2,7701 - 2,4952}{2,4952} \times 100\%$$

$$\text{Throughput} = 11,02\%$$

## 4.2 Pembahasan

Hasil penelitian ini menunjukkan bahwa penerapan algoritma *priority queue* mampu memberikan solusi efektif dalam proses penjadwalan kegiatan yang memiliki tingkat prioritas berbeda. Melalui algoritma ini, sistem dapat menentukan urutan kegiatan berdasarkan tingkat kepentingannya, sehingga kegiatan yang lebih mendesak dapat diprioritaskan terlebih dahulu.

Berdasarkan pengujian yang telah dilakukan, sistem penjadwalan kegiatan yang dirancang telah berjalan dengan baik dan sesuai dengan tujuan awal pembuatan. Sistem dirancang untuk dapat memudahkan pegawai dalam melakukan penjadwalan suatu kegiatan kantor yang membutuhkan *driver* dan unit yang telah disediakan oleh kantor, sistem juga membantu kepala sopir untuk dapat memantau jadwal dari setiap sopir dan bagaimana status dari setiap sopir yang telah dijadwalkan. Setiap fitur utama dari sistem telah berhasil di uji dengan baik menggunakan metode *Blackbox testing*, metode pengujian ini berfokus pada *input* dan *output* tanpa melihat kode program secara langsung. Pada pengujian *Blackbox* diberikan skenario untuk melihat bagaimana sistem menjalankan skenario tersebut. Sebelum skenario dijalankan perlu untuk menentukan kriteria utama yang akan menjadi penomoran prioritas, penomoran ini melibatkan metode AHP. Metode ini akan memberikan penilaian terhadap

setiap kriteria untuk kemudian melihat mana yang lebih penting dibandingkan kriteria lainnya. adapun kriteria yang dibandingkan yaitu jabatan, tenggat waktu serta jenis kegiatan. Dari perhitungan dengan memanfaatkan metode AHP melalui analisis perbandingan berpasangan dan dihitung menggunakan matriks diperoleh nilai bobot masing-masing kriteria sebagai berikut, jabatan bernilai 0,648%, tenggat waktu bernilai 0,230% sedangkan jenis kegiatan bernilai 0,122%. Masing-masing kriteria telah memiliki skor untuk setiap kondisi seperti jabatan memiliki skor 4-1 yang dilihat seberapa tinggi posisi jabatan dikantor BI Sulawesi Tengah, kemudian tenggat waktu memiliki skor 3-1 yang dilihat dari seberapa mendesak kegiatan akan dilakukan, terakhir jenis kegiatan memiliki skor 1-2 dilihat dari lokasi kegiatan yang akan dilakukan. Skor dari kriteria tersebut akan dikalikan dengan bobot dari kriteria yang telah diperoleh sebelumnya menggunakan metode AHP, dari perhitungan tersebut kemudian didapatkan nomor prioritas penjadwalan.

*Tabel 4. 22 Tabel Pengujian Fungsional Bobot*

Jabatan	Tanggal Mulai	Tanggal Akhir	Jenis Kegiatan	Skor Harapan
Kepala Perwakilan BI	01/12/2025	01/12/2025	Dalam Kota	3,296
Manager	06/12/2025	06/12/2025	Luar Kota	1,648
Deputi	01/12/2025	02/12/2025	Luar Kota	2,526
Manager	29/11/2025	29/11/2025	Dalam Kota	2,23

User diminta untuk memasukkan jadwal dengan tingkatan prioritas yang berbeda dengan kombinasi yang sesuai dengan tabel 4.22 diatas, adapun keterangan dari tabel diatas sebagai berikut, kriteria jabatan akan mendapatkan skor sesuai tingkatannya, seperti kepala perwakilan BI mendapat skor 4, manager 2, dan deputi 3. Kriteria tenggat waktu mendapatkan skor sesuai dengan seberapa mendesak kegiatan, seperti tenggat 1 hari mendapat skor 3, tenggat 3 hari mendapat skor 2 dan tenggat lebih dari 3 hari mendapat skor 1. Kriteria jenis kegiatan mendapat skor sesuai dengan lokasi kegiatan, seperti didalam kota mendapat skor 2 dan diluar kota mendapat skor 1.

Berdasarkan tabel 4.22 pengujian dilakukan pada tanggal 29 November 2025 sehingga untuk skor yang didapatkan pada baris pertama yaitu jabatan 4, tenggat waktu 2 dan jenis kegiatan 2, setelah itu masing-masing skor dikalikan dengan bobot kriteria yaitu jabatan di 0,648, tenggat waktu 0,230 dan jenis kegiatan 0,122. Hasil perkalian dari masing-masing bobot kemudian ditambahkan untuk mendapatkan nomor prioritas atau skor prioritas penjadwalan. Maka untuk baris pertama mendapatkan skor prioritas di 3,296, begitu selanjutnya ditarik kedua dengan skor jabatan 2, tenggat waktu 1, dan jenis kegiatan 1 maka skor prioritas di 1,648, kemudian baris ketiga skor jabatan 3, tenggat waktu 2, dan jenis kegiatan 1 maka skor prioritas di 2,526, lalu baris keempat skor jabatan 2, tenggat waktu 3, dan jenis kegiatan 2 maka skor prioritas

di 2,23. Dari perhitungan tersebut diuji coba ke sistem apakah nilai yang dihasilkan oleh sistem akan sama seperti skor prioritas harapan ataupun tidak.

asal	jenis_activity_id	tgl_mulai	tgl_akhir	priority_score
1 Kepala Perwakilan	1	2025-12-01	2025-12-01	3,30
2 Manager	2	2025-12-06	2025-12-06	1,65
3 Deputi	2	2025-12-01	2025-12-02	2,53
4 Manager	1	2025-11-29	2025-11-29	2,23

Gambar 4. 24 Database Hasil Pengujian Fungsional Bobot

Berdasarkan gambar 4.24 dapat dilihat bahwa skenario untuk menguji fungsional dari bobot memiliki skor yang sesuai dengan harapan, hal ini membuktikan bahwa pengujian *Blackbox* sudah berjalan dengan baik, seluruh fitur dimulai dari *login*, melakukan penjadwalan, pembuatan surat LDP, sampai pengiriman surat LDP berjalan sebagaimana yang diharapkan tanpa adanya kesalahan sistem yang tidak dapat ditangani oleh *user*. Adapun fitur-fitur tambahan juga dilakukan pengujian untuk melihat apakah fitur tersebut dapat membantu dalam sistem ataupun tidak, fitur seperti *register* untuk dapat membantu *user* yang tidak memiliki akun untuk dapat masuk kedalam sistem, fitur *dashboard* yang dapat menampilkan jadwal yang telah dibuat oleh *user* lainnya sehingga semua *user* dapat melihat jadwal yang telah berhasil dibuat dan dijadwalkan dari tanggal berapa dan berakhir di tanggal berapa, fitur mengubah status *driver* yang dapat membantu *user* untuk segera mendapatkan *driver* yang sudah selesai dalam penugasan untuk segera dapat dijadwalkan kembali. Fitur-fitur tambahan tersebut telah berhasil diuji dan mendapatkan hasil yang sesuai dengan kebutuhan dari *user* akan fitur tersebut.

Pada pengujian algoritma dilakukan dengan memanfaatkan *framework Laravel* untuk melakukan simulasi penjadwalan secara otomatis untuk dapat mengukur seberapa cepat algoritma dapat menghasilkan suatu penjadwalan dengan beberapa skenario yang telah ditentukan. Dengan penggunaan *framework Laravel* memudahkan simulasi untuk melakukan penjadwalan secara otomatis dengan proses penjadwalan yang sesuai dengan penjadwalan manual, namun dengan isi jadwal yang secara acak diisi sampai memenuhi jumlah data jadwal yang diinginkan untuk pengujian kecepatan algoritma dalam memproses sejumlah skenario tersebut.

Sama halnya seperti proses penjadwalan manual, sistem ini menggunakan algoritma *priority queue* dalam menentukan urutan prioritas setiap kegiatan. Algoritma ini bekerja dengan cara menyusun jadwal berdasarkan bobot prioritas yang telah dihitung sebelumnya. Terdapat tiga kriteria utama yang menjadi dasar dalam perhitungan bobot prioritas, yaitu jabatan dengan nilai bobot 0,648, tenggat waktu dengan nilai bobot 0,230, dan jenis kegiatan dengan nilai bobot 0,122 sehingga jika dijumlahkan bernilai 1. Dengan mempertimbangkan ketiga

kriteria tersebut, sistem dapat menghasilkan urutan jadwal yang menempatkan kegiatan dengan prioritas tertinggi untuk diproses terlebih dahulu, sehingga penjadwalan menjadi lebih efisien dan terstruktur.

Berdasarkan hasil pengujian algoritma yang dilakukan melalui simulasi pada *framework Laravel* dimana pada pengujian tersebut telah membandingkan algoritma *priority queue* dengan *first come first serve* yang dikenal sebagai algoritma sederhana dalam suatu penjadwalan, perbandingan ini kemudian dilakukan untuk dapat melihat bagaimana tingkat eksekusi waktu yang didapatkan kedua algoritma dalam melakukan penjadwalan. Adapun data yang digunakan berjumlah 50 untuk setiap pengujian sehingga jika ditotal terdapat 200 data dalam melakukan perbandingan tersebut. Dari pengujian tersebut didapatkan bahwa untuk algoritma *priority queue* mendapatkan rata-rata waktu eksekusi di 2,7701 detik sedangkan pada FCFS berada di 2,4952 detik, dari kedua hasil tersebut kemudian dimasukkan kedalam perhitungan throughput untuk mendapatkan hasil dari peningkatan dari algoritma *priority queue*, dimana untuk hasil yang didapatkan berada di 11,02% yang dapat disimpulkan bahwa algoritma *priority queue* berhasil meningkatkan Throughput sistem sebesar 11,02% dibandingkan dengan algoritma FCFS. Peningkatan ini membuktikan bahwa mekanisme penentuan prioritas yang diterapkan mampu mengoptimalkan alokasi sumber daya (transportasi) sehingga sistem dapat menyelesaikan lebih banyak kegiatan dalam durasi waktu yang sama.

Adapun keterbatasan dalam penelitian ini terletak pada ketergantungan terhadap *framework Laravel*, baik dalam proses pembuatan sistem maupun tahap pengujian sistem dan algoritma. Selain itu, penelitian ini belum membahas secara mendalam mengenai pengalaman pengguna, khususnya terkait mekanisme notifikasi yang diterima oleh pengguna setelah penjadwalan dilakukan.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Setelah melakukan serangkaian penelitian yang dilakukan terkait optimalisasi penjadwalan kegiatan dengan menggunakan algoritma *priority queue*, dapat disimpulkan sebagai berikut:

1. Optimalisasi sistem penjadwalan kegiatan yang dilakukan di Kantor Perwakilan Bank Indonesia Provinsi Sulawesi Tengah diwujudkan melalui pembangunan sistem berbasis *website* yang terstruktur dan efisien. Dengan adanya sistem ini, proses penjadwalan yang sebelumnya dilakukan secara manual menjadi lebih mudah yang dibuktikan dengan hasil *Blackbox testing* yang menunjukkan bahwa segala fitur yang ada sudah sesuai dengan kebutuhan, pada pengujian tersebut diberikan skenario dengan 4 macam kondisi berurut yaitu jabatan dengan skor 4, 2, 3, 2, kemudian tengat waktu 2, 1, 2, 3, dan jenis kegiatan 2, 1, 1, 2, dari skor tersebut dikalikan dengan bobot masing-masing kriteria. Dalam memperoleh bobot kriteria peneliti memanfaatkan metode AHP dengan hasil perhitungan untuk bobot kriteria jabatan bernilai 0,648, tengat waktu bernilai 0,230 dan jenis kegiatan bernilai 0,122. Setelah berhasil memperoleh bobot kriteria maka skenario sebelumnya menghasilkan skor prioritas berurut di 3,296, 1,648, 2,526, 2,23 dan hasil tersebut telah sesuai dengan pengujian sistem, maka dari itu segala fitur berjalan sesuai kebutuhan yang memudahkan dalam proses pengajuan kegiatan serta meningkatkan keteraturan dalam distribusi jadwal kegiatan.
2. Penggunaan algoritma *priority queue* terbukti efektif dalam membantu proses distribusi dan pengelolaan jadwal kegiatan. Berdasarkan hasil pengujian waktu eksekusi algoritma *priority queue* yang dibandingkan dengan FCFS mampu memberikan peningkatan sebesar 11,02 %. Prinsip prioritas yang diterapkan memungkinkan sistem mengatur kegiatan secara lebih adil sehingga mampu meminimalkan bentrokan jadwal. Hal ini dibuktikan dengan perbedaan status penjadwalan pada setiap data sesuai dengan tingkat prioritasnya masing-masing. Dengan demikian sistem ini dapat meningkatkan efisiensi karena sistem secara otomatis mempertimbangkan faktor prioritas dalam penjadwalan tanpa intervensi manual yang berpotensi subjektif.

#### **5.2 Saran**

Berdasarkan hasil dari penelitian ini, terdapat beberapa saran untuk pengembangan lebih lanjut, yaitu sebagai berikut:

1. Melakukan penelitian lebih lanjut untuk dapat memasukkan fitur absensi sopir didalam sistem serta laporan kinerja. Sehingga data kehadiran dan performa kerja dapat dicatat secara otomatis.

2. Pengembangan lebih lanjut dengan pembuatan *dashboard* yang lebih dinamis dalam melihat jadwal tugas dari sopir dengan warna yang berbeda untuk setiap sopir dan setiap tugas yang sementara berjalan atau yang sudah selesai dilakukan.
3. Sistem dapat ditingkatkan dengan fitur pelacakan posisi kendaraan secara *real-time* melalui teknologi geolokasi.
4. Melakukan perbandingan antara algoritma priority queue dengan algoritma lainnya seperti genetika, Round Robin, dan Shortest Job First.

## DAFTAR PUSTAKA

- A. Siroj, R., Afgani, W., Fatimah, Septaria, D., & Zahira Salsabila, G. (2024). METODE PENELITIAN KUANTITATIF PENDEKATAN ILMIAH UNTUK ANALISIS DATA. *Jurnal Review Pendidikan Dan Pengajaran*, 7 nomor 3. <https://journal.universitaspahlawan.ac.id/index.php/jrpp>
- Aipina, D., & Witriyono, H. (2022). PEMANFAATAN FRAMEWORK LARAVEL DAN FRAMEWORK BOOTSTRAP PADA PEMBANGUNAN APLIKASI PENJUALAN HIJAB BERBASIS WEB. *Jurnal Media Infotama*, 18(1). <https://jurnal.unived.ac.id/index.php>
- Amir S, A., Assidiq, M., & Qashlim, A. A. (2021). SISTEM INFORMASI PENJADWALAN MATA PELAJARA BERBASIS WEB MENGGUNAKAN METODE ALGORITMA GENETIKA PADA SMKN 6 MAJENE. *Journal Peqguruang: Conference Series*, 3(2), 861. <https://doi.org/10.35329/jp.v3i2.2235>
- Bank Indonesia Institute. (2020). *Profil Bank Indonesia*. Bank Indonesia. <https://www.bi.go.id/id/tentang-bi/profil/Default.aspx>
- Dermawan, G., Munawar, Z., Kurnia Sastradipraja, C., Indah Putri, N., & Sutjiningtyas, S. (2023). *Membangun Sistem Antrian Online Untuk Bimbingan Tugas Akhir*. 10. <https://jurnal.plb.ac.id/index.php/tematik/index>
- Femy Mulya, M., Trisanto, D., & Rismawati, N. (2020). Analisis Dan Implementasi Metode Earliest Due Date (EDD) Untuk Meminimalisir Keterlambatan Dalam Proses Penjadwalan Perbaikan Kendaraan. *Faktor Exacta*, 13(3). <https://doi.org/10.30998/faktorexacta.v13i3.7254>
- Galuh, Y., & Siregar, M. Y. (2025). *Analisis Implementasi Metode AHP dan SAW pada Sistem Pendukung Keputusan Pemilihan Siswa Berprestasi* (Vol. 2, Issue 1).
- Hasan, S., & M.Syahran, J. (2023). Jenis-Jenis Penelitian Dalam Penelitian Kuantitatif dan Kualitatif. *Jurnal Pendidikan, Sosial & Humaniora*, Vol. 1 Nomor 1. <http://ejournal.yayasanpendidikanzurriyatulquran.id/index.php/qosim>
- Hilmawan Masyfa, F., Primanita Kartikasari, D., & Tibyani. (2023). Penjadwalan dan Pelaporan Menggunakan Dynamic Priority Scheduling dan Geolocation untuk Keamanan Lingkungan Scheduling and Reporting using Dynamic Priority Scheduling and Geolocation for Environmental Security. In *Februari* (Vol. 22, Issue 1).
- Munthafa, A. E., & Mubarok, H. (2017). *PENERAPAN METODE ANALYTICAL HIERARCHY PROCESS DALAM SISTEM PENDUKUNG KEPUTUSAN PENENTUAN MAHASISWA BERPRESTASI*. 3(2).
- Rahayu, D. K., & Martini, S. (2022). PERAN MODA TRANSPORTASI DARAT DALAM MOBILITAS TENAGA KERJA. In *Agustus* (Vol. 22, Issue 2). <https://doi.org/10.26593/JTRANS.V22I2.6061.109-116>
- Setyawatu, R., & Bachtiar Maulachela, A. (2020). *Penerapan Algoritma Dynamic Priority Scheduling Pada Aplikasi Antrian Pencucian Mobil Berbasis Mobile*

- (*Implementation of Dynamic Priority Scheduling Algorithm in Mobile Car Queue Washing Applications*). 2(1), 29–35.
- Simanullang, Y. Y., Munthe, R. P., Aritonang, O. A., Sitompul, B. C., Sitorus, P. A., Hasugian, C., & Siallagan, E. H. (2024). PERAN BANK INDONESIA TERHADAP PELAKSANAAN KLIRING ANTAR BANK. *Jurnal PenKoMi : Kajian Pendidikan Dan Ekonomi*, 7(1). <http://jurnal.stkipbima.ac.id/index.php/PK/index>
- Syarif, M., & Risdiansyah, D. (2024). PEMANFAATAN METODE PROTOTYPE DALAM PERANCANGAN SISTEM INFORMASI PENJUALAN BERBASIS WEBSITE. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 4).
- Valent, H., Muslim Sinaga, R., Putra, S., Halawa, P., Audy Priscillia, S., & Ramadhani, F. (2025). IMPLEMENTASI ALGORITMA ANTRIAN PRIORITAS MENGGUNAKAN ARRAY DI PYTHON UNTUK SISTEM ANTRIAN LAYANAN DARURAT. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 9, Issue 1).
- Wardana, R., Sucipto, & Firliana, L. (2023). SISTEM LAYANAN ANTRIAN KLINIK KESEHATAN BERBASIS WEB. *Multitek Indonesia : Jurnal Ilmiah*, 16, 22–36. <http://journal.umpo.ac.id/index.php/multitek>
- Wintana, D., Pribadi, D., & Nurhadi, M. Y. (2022). *Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing*. <http://jurnal.bsi.ac.id/index.php/larik>
- Wulandari, K. D., Taufiq, R., Priyanggodo, D. Y., & Daniarti, Y. (2024). Rancang Bangun Sistem Informasi Penjadwalan Pengiriman Barang pada CV Lestari Mandiri. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 7(3), 1179–1187. <https://doi.org/10.32493/jtsi.v7i3.42889>
- Yuslikhatun Ulya, D., & Supriyono. (2025). *DIGITALISASI SISTEM PELAPORAN PENJADWALAN KEGIATAN PADA UNIT KEPEGAWAIAN PUSKESMAS DAWE*. 2(3), 342–356. <https://doi.org/10.62335>
- Zakharia, Widiatry, & Noor Kamala Sari, N. (2021). Sistem Informasi Penjadwalan Supir Bus Po Logos Berbasis Website. In *JOINTECOMS (Journal of Information Technology and Computer Science)* p-ISSN: xxxx-xxxx (Vol. 1, Issue 1).

## DAFTAR RIWAYAT HIDUP



Nama : Munawir Muhsin  
Tempat, Tanggal Lahir : Tatari, 20 Juli 2003  
Alamat : Kel. Lamberea,  
Kec. Bungku Tengah, Kab. Morowali  
Nomor Telepon/HP : 082295145259  
Email : munawirmuhsin1@gmail.com

### Pendidikan:

1. SDN 2 Bungku Tengah  
Tahun : 2009-2015
2. MTsN Morowali  
Tahun : 2015-2018
3. MAN Morowali  
Tahun : 2018-2021
4. Universitas Tadulako  
Program Studi: S1 Teknik Informatika  
Tahun : 2021-sekarang

### Pengalaman:

Pertukaran Mahasiswa di UPN Veteran Yogyakarta Agustus 2022 – Januari 2023  
Internship KPw BI Prov. Maluku Utara Februari – Juni 2024

## LAMPIRAN

### Lampiran 1. Surat Permintaan Data

	<p><b>KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN TEKNOLOGI</b> <b>UNIVERSITAS TADULAKO</b> <b>FAKULTAS TEKNIK</b> <b>JURUSAN TEKNOLOGI INFORMASI</b> Jl. Soekarno Hatta Km. 9 Telp (0451) 422611 - 422355, Fax. (0451) 454014 Email: <a href="mailto:teknologi.informasi@untad.ac.id">teknologi.informasi@untad.ac.id</a> Palu - Sulawesi Tengah 94118</p>										
<p>Nomor : 2345/UN.28.6/TI.00/2025 Lampiran : - Perihal : Permohonan Izin melakukan Penelitian dan Pengambilan Data</p>											
<p>Kepada Yth. <b>Kepala Kantor Perwakilan Bank Indonesia Provinsi Sulawesi Tengah</b> di- Tempat</p>											
<p>Assalamu'alaikum warahmatullahi wabarakatuh... Dalam rangka untuk melakukan penelitian Tugas Akhir, maka kami memohon kepada Bapak/Ibu agar menerima Mahasiswa kami untuk melakukan penelitian dan pengambilan data di tempat bapak/ibu sebagai upaya untuk penyelesaian Tugas Akhir.</p>											
<p>Berikut Identitas Mahasiswa tersebut:</p> <table style="width: 100%; border-collapse: collapse;"><tr><td style="width: 50%;">Nama</td><td style="width: 50%;">: Munawir Muhsin</td></tr><tr><td>Stambuk</td><td>: F55121017</td></tr><tr><td>Program Studi</td><td>: Teknik Informatika</td></tr><tr><td>Jurusan</td><td>: Teknologi Informasi</td></tr><tr><td>Asal</td><td>: Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Fakultas Teknik Universitas Tadulako</td></tr></table>		Nama	: Munawir Muhsin	Stambuk	: F55121017	Program Studi	: Teknik Informatika	Jurusan	: Teknologi Informasi	Asal	: Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Fakultas Teknik Universitas Tadulako
Nama	: Munawir Muhsin										
Stambuk	: F55121017										
Program Studi	: Teknik Informatika										
Jurusan	: Teknologi Informasi										
Asal	: Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Fakultas Teknik Universitas Tadulako										
<p>Demikian, surat pengantar ini kami buat untuk digunakan sebagai mana mestinya. Atas perhatian dan kebijaksanaan Bapak/Ibu, diucapkan Terima Kasih.</p>											
 S-B7ZQJIUPAADVLMKQ3NFAR	<p>Palu, 21 Juli 2025 Mengetahui,   Anisa Muhamad Basim, S.Hum., M.Cs. NIP. 19731122005012002</p>										

### Lampiran 2. Dokumentasi Penelitian



### Lampiran 3. Listing Coding

## 1. Halaman *Dashboard*

```
@extends('layouts.main')
@section('content')
<div class="content">
    
    <ul class="mt-1 ms-3">
        <li>{{ $title }}</li>
    </ul>
    <div class="card text-bg-light shadow-lg">
        <h5 class="card-header">Driver Chart</h5>
        <div class="card-body">
            <div id="calendar"></div>
            </div>
        </div>
        </div>
    </main>

    <!-- Modal Detail Jadwal Sopir -->
    <div class="modal fade"
        id="eventDetailModal"
        tabindex="-1" aria-labelledby="eventDetailModalLabel" aria-hidden="true">
        <div class="modal-dialog modal-dialog-scrollable">
            <div class="modal-content shadow-lg">
                <div class="modal-header">
                    <h5 class="modal-title fw-bold"
                        id="eventDetailModalLabel">Detail Jadwal Sopir</h5>
                <button type="button" class="btn-close" data-bs-
```

```
dismiss="modal" aria-  
label="Close">></button>  
</div>  
<div class="modal-  
body">  
<div class="row">  
<div class="col-md  
text-center">  
<h4  
id="eventKegiatan" class="mb-  
1"></h4>  
<p class="mb-  
1">Lokasi: <span  
id="eventLokasi"></span></p>  
<p class="mb-  
1">Mulai: <span  
id="eventTglMulai"></span></p>  
>  
<p class="mb-  
1">Akhir: <span  
id="eventTglAkhir"></span></p>  
>  
<p class="mb-  
1">Jumlah Driver: <span  
id="eventJumlahDriver"></span>  
></p>  
<p class="mb-1">Unit  
<br> <span  
id="eventUnit"></span></p>  
<p class="mb-  
1">Keterangan <br> <span  
id="eventKeterangan"></span>  
</p>  
</div>  
</div>  
</div>  
</div>  
</div>  
</div>  
@endsection  
2. Halaman Penjadwalan  
@extends('layouts/main')  
@section('content')  

<ul class="mt-1 ms-3">
    <li>{{ $title }}</li>
</ul>
<div class="card text-bg-
light shadow-lg">
    <h5 class="card-
header">Reservasi</h5>
    <div class="card-
body">
        <form
            method="POST"
            action="/reservasi" >
            @csrf
            <input
                type="hidden"
                name="driver_id" value="{{
                $defaultDriverId }}">
            <div class="mb-
1 row">
                <label
                    for="pegawai" class="col-
sm-2 col-form-label">Nama
                Pegawai</label>
                <div
                    class="col-sm-9">
                    <input
                        type="text" class="form-
control" name="pegawai"
                        id="pegawai" autofocus
                        required value="{{
                        old('pegawai') }}">
                    @error('pegawai')
                    <div
                        class="invalid-feedback">
                        {{
                        $message }}</div>
                    @enderror
                </div>
                </div>
                <div class="mb-
1 row">
                    <label
                        for="kegiatan" class="col-
sm-2 col-form-label">Nama
                    Kegiatan</label>
                    <div
                        class="col-sm-9">
                        <input
                            type="text" class="form-
control" name="kegiatan"
                            id="kegiatan" required
                            value="{{
                            old('kegiatan') }}">
                    @error('kegiatan')
                    <div
                        class="invalid-feedback">
                        {{
                        $message }}</div>
                    @enderror
                </div>
                </div>
                <div class="mb-
1 row">

```

```
<label
for="lokasi" class="col-sm-2
col-form-label">Lokasi
Kegiatan</label>
<div
class="col-sm-9">
<input
type="text" class="form-
control" name="lokasi"
id="lokasi" value="{}{{
old('lokasi') }}">
@error('lokasi')
<div
class="invalid-feedback">
{{{
$message }}}
</div>
@enderror
</div>
</div>
<div class="mb-
1 row">
<label
for="tgl_mulai" class="col-
sm-2 col-form-
label">Tanggal
Mulai</label>
<div
class="col-sm-9">
<input
type="date" class="form-
control" name="tgl_mulai"
id="tgl_mulai" value="{}{{
old('tgl_mulai') }}">
@error('tgl_mulai')
<div
class="invalid-feedback">
{{{
$message }}}
</div>
@enderror
</div>
</div>
</div>
<div class="mb-
1 row">
<label
for="tgl_akhir" class="col-
sm-2 col-form-
label">Tanggal Akhir</label>
<div
class="col-sm-9">
<input
type="date" class="form-
control" name="tgl_akhir"
id="tgl_akhir" value="{}{{
old('tgl_akhir') }}">
@error('tgl_akhir')
<div
class="invalid-feedback">
{{{
$message }}}
</div>
@enderror
</div>
</div>
<div class="mb-
1 row">
<label
for="waktu" class="col-sm-2
col-form-
label">Waktu</label>
<div
class="col-sm-9">
<input
type="time" class="form-
control" name="waktu"
id="waktu" value="{}{{
old('waktu') }}">
@error('waktu')
<div
class="invalid-feedback">
{{{
$message }}}
</div>
@enderror
</div>
</div>
</div>
```

```

        </div>
        <!-- Input jumlah
driver hanya ditampilkan jika
ada driver yang tersedia -->
@if($countAvailableDrivers
> 0)
    <div
    class="mb-1 row">
        <label
        for="jumlah_driver"
        class="col-sm-2 col-form-
label">Jumlah
Driver</label>
        <div
        class="col-sm-9">
            <select
            class="form-control"
            name="jumlah_driver">
                @for
                ($i = 1; $i <=
                $countAvailableDrivers;
                $i++)
                <option value="{{$i}}>{{$
i}}</option>
            @endfor
            </select>
        </div>
    </div>
    @endif
    <!-- End of Input
jumlah driver -->
    <div class="mb-
1 row">
        <label
        for="unit_id" class="col-sm-
2 col-form-label">Jenis
Unit</label>
        <div
        class="col-sm-9">
            <div
            id="unit-container">
                <select
                class="form-control unit-

```

```

                select" name="unit_id[]"
                id="unit_id">
                <option value="">Pilih
                Unit</option>
                @foreach ($availableUnits as
                $unit)
                <option value="{{$unit->id
                }}>{{$unit->jenis
                }}</option>
                @endforeach
            </select>
        </div>
        <div
        class="tambah-unit mt-2"
        id="tambah-unit"
        style="cursor: pointer;">
            
            <span>Tambah Unit</span>
        </div>
        @error('unit_id')
            <div
            class="invalid-feedback">
                {{{
                $message }}}
            </div>
        @enderror
        </div>
        <div class="mb-
1 row">
            <label
            for="keterangan" class="col-
sm-2 col-form-
label">Keterangan</label>
            <div
            class="col-sm-9">
                <input
                type="text" class="form-
control" name="keterangan"

```

```

id="keterangan" value="{!! old('keterangan') !!}" @enderror>
@error('keterangan')
    <div class="invalid-feedback">
        {{ $message }}
    </div>
    @enderror
</div>
<div class="mb-1 row">
    <label for="jenis_activity_id" class="col-sm-2 col-form-label">Jenis Kegiatan</label>
    <div class="col-sm-9">
        <select class="form-control" name="jenis_activity_id" id="jenis_activity_id" value=">
            @foreach ($kegiatan as $jenis)
                @if (old('jenis_activity_id') == $jenis->id)
                    <option value="{!! $jenis->id !!}" selected="selected">{!! $jenis->jenis_kegiatan !!}</option>
                @else
                    <option value="{!! $jenis->id !!}">{!! $jenis->jenis_kegiatan !!}</option>
                @endif
            @endforeach
        </select>
        @error('jenis_activity_id')
            <div class="invalid-feedback">
                {{ $message }}
            </div>
        @enderror
    </div>
    <button class="btn btn-primary mt-3 col-md-1" style="margin-left: 190px" type="submit">Ajukan</button>
</form>
</div>
<script>
document.addEventListener('DOMContentLoaded', function() {
    let unitCount = 1;
    const maxUnits = 5;
    const unitContainer = document.getElementById('unit-container');
    const tambahUnitBtn = document.getElementById('ambah-unit');
    let availableUnits = @json($availableUnits);
    function updateAvailableUnits() {
        const selectedUnits = Array.from(unitContainer.querySelectorAll('select')).map(select => select.value).filter(value => value !== '');
        unitContainer.querySelectorAll('select').forEach(select => {
            const selectedValue = select.value;
            const currentOptions =

```

```

Array.from(select.options).map(option => option.value);
availableUnits.forEach(unit => {
    if (!currentOptions.includes(unit.id.toString()) && !selectedUnits.includes(unit.id.toString())) {
        const option = document.createElement('option');
        option.value = unit.id;
        option.textContent = unit.jenis;
        select.appendChild(option);
    }
});
// Sembunyikan opsi yang sudah dipilih di select lain
Array.from(select.options).forEach(option => {
    if (option.value !== "" && option.value !== selectedValue) {
        option.style.display = selectedUnits.includes(option.value) ? 'none' : '';
    }
});
// Disable "Tambah Unit" button if all units are selected or max units reached
const availableOptionsCount = availableUnits.length - selectedUnits.length;
tambahUnitBtn.style.display = (availableOptionsCount === 0 || unitCount >= maxUnits) ? 'none' : 'block';
}
unitContainer.addEventListener('change', updateAvailableUnits);
tambahUnitBtn.addEventListener('click', function() {
    if (unitCount < maxUnits && availableUnits.length > unitCount) {
        unitCount++;
        const newUnitDiv = document.createElement('div');
        newUnitDiv.className = 'mt-2';
        newUnitDiv.innerHTML = `
            <select
            class="form-control unit-select"
            name="unit_id[]"
            >
                <option
                value="">Pilih Unit</option>
            </select>
        `;
        unitContainer.appendChild(newUnitDiv);
        updateAvailableUnits();
    } else {
        alert('Maksimum unit telah ditambahkan atau semua unit telah dipilih.');
    }
});
});
</script>
@endsection

```