



**PENERAPAN ALGORITMA *GREEDY* UNTUK OPTIMALISASI  
PENJADWALAN MATA PELAJARAN PADA SISTEM INFORMASI  
AKADEMIK SMK NEGERI 5 KENDAL BERBASIS WEB**

**TUGAS AKHIR**

**MUHAMMAD FAHRUR RIZAL**

**NPM 20670089**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK DAN INFORMATIKA  
UNIVERSITAS PGRI SEMARANG**

**2024**



**PENERAPAN ALGORITMA *GREEDY* UNTUK OPTIMALISASI  
PENJADWALAN MATA PELAJARAN PADA SISTEM INFORMASI  
AKADEMIK SMK NEGERI 5 KENDAL BERBASIS WEB**

**TUGAS AKHIR**

**Diajukan kepada Fakultas Teknik dan Informatika Universitas PGRI  
Semarang untuk Penyusunan Tugas Akhir**

**MUHAMMAD FAHRUR RIZAL**

**NPM 20670089**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK DAN INFORMATIKA**

**UNIVERSITAS PGRI SEMARANG**

**2024**

**TUGAS AKHIR**  
**PENERAPAN ALGORITMA GREEDY UNTUK OPTIMALISASI**  
**PENJADWALAN MATA PELAJARAN PADA SISTEM INFORMASI**  
**AKADEMIK SMK NEGERI 5 KENDAL BERBASIS WEB**

Disusun dan diajukan oleh  
**MUHAMMAD FAHRUR RIZAL**  
NPM 20670089

Telah disetujui oleh pembimbing untuk dilanjutkan di  
hadapan Dewan Penguji

Semarang, 13 Agustus 2024

Pembimbing Utama

Pembimbing Pendamping



**Bambang Agus Herlambang, S.Kom., M.Kom.**  
NIDN.0601088201



**Noora Qotrun Nada, ST.,M.Eng.**  
NIDN.0626028201

**TUGAS AKHIR**  
**PENERAPAN ALGORITMA GREEDY UNTUK OPTIMALISASI**  
**PENJADWALAN MATA PELAJARAN PADA SISTEM INFORMASI**  
**AKADEMIK SMK NEGERI 5 KENDAL BERBASIS WEB**

Disusun dan diajukan oleh

**MUHAMMAD FAHRUR RIZAL**

**NPM 20670089**

Telah dipertahankan di depan Dewan Penguji pada tanggal 19 Agustus 2024  
dan dinyatakan telah memenuhi syarat Dewan Penguji



Ketua

**Beni Toto Husodo, S.T., M.T.**

**NIP/NPP 136901387**

Sekretaris

**Bambang Agus H., S.Kom, M.Kom.**

**NIP/NPP 148201433**

Penguji I

**Bambang Agus H., S.Kom, M.Kom.**

**NIP/NPP 148201433**

Penguji II

**Noora Qotrun Nada, ST., M.Eng.**

**NIP/NPP 158201485**

Penguji III

**Khoiriya Latifah. S.Kom., M.Kom**

**NIP/NPP 148201443**



## **MOTO DAN PERSEMBAHAN**

### **Motto :**

“... Half of the troubles of this life can be traced to saying yes too quickly and not saying no soon enough...” -Idk who wrote that sentence

### **Persembahan :**

Saya persembahkan tugas akhir ini untuk :

1. Kedua orang tua dan kakaku tercinta
2. Dosen pembimbing yang telah membimbing dan memberikan arahan selama proses penyusunan tugas akhir ini.
3. Teman-teman yang selalu memberikan semangat, bantuan, dan kebersamaan.
4. Seluruh guru dan staf SMK Negeri 5 Kendal yang telah memberikan izin dan bantuan selama penelitian ini berlangsung.
5. Almamater Universitas PGRI Semarang

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Fahrur Rizal

NPM : 20670089

Program Studi : Informatika

Fakultas : Teknik dan Informatika

Menyatakan dengan sebenarnya bahwa tugas akhir yang saya buat ini benar-benar merupakan hasil karya saya sendiri, bukan plagiarisme.

Apabila pada kemudian hari tugas akhir ini terbukti hasil plagiarisme, saya bersedia menerima sanksi atas perbuatan tersebut.

Semarang, 27-8-2024

Yang membuat pernyataan



MUHAMMAD FAHRUR RIAL

NPM 20670089

## ABSTRAK

Perkembangan pesat teknologi informasi telah membawa perubahan signifikan dalam berbagai bidang, termasuk pendidikan. Peningkatan kualitas sistem informasi akademik menjadi kebutuhan mendesak untuk mengatasi masalah seperti efisiensi proses pengolahan data dan akurasi informasi. Penelitian ini bertujuan untuk meningkatkan pemahaman dan mengidentifikasi masalah dalam penggunaan sistem informasi akademik berbasis web di SMK Negeri 5 Kendal, serta mengevaluasi efektivitas dan efisiensinya dalam mendukung proses administrasi. Penelitian ini menggunakan metode Rapid Application Development (RAD), yang menekankan kecepatan dan iterasi dalam pengembangan sistem. Tahapan penelitian mencakup Requirement Planning, User Design, dan Construction, yang diikuti oleh pengujian White Box Testing, Black Box Testing, dan User Acceptance Testing (UAT). Dalam pengembangan sistem, framework Laravel dipilih untuk memfasilitasi pengelolaan data akademik dan meningkatkan pengalaman pengguna administrator sistem. Hasil penelitian menunjukkan bahwa sistem optimasi penjadwalan yang dikembangkan berhasil memenuhi tujuan utama, dengan tingkat keberhasilan 100% pada pengujian Black Box dan 89.2% pada UAT. Algoritma Greedy yang digunakan dalam proses optimasi terbukti efektif dalam menghasilkan jadwal yang optimal dengan meminimalkan konflik waktu. Dengan demikian, sistem ini dapat memberikan rekomendasi jadwal yang lebih baik dan mendukung proses administrasi di SMK Negeri 5 Kendal secara lebih efisien.

Kata Kunci: Sistem Informasi Akademik, Rapid Application Development, Framework Laravel, Algoritma Greedy, Penjadwalan, SMK Negeri 5 Kendal.

## PRAKATA

Segala Puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan segala rahmatNya sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Penerapan Algoritma *Greedy* Untuk Optimalisasi Penjadwalan Mata Pelajaran Pada Sistem Informasi Akademik Smk Negeri 5 Kendal Berbasis Web” guna memenuhi syarat untuk menyelesaikan Program Studi Informatika S-1 pada Fakultas Teknik dan Informatika Universitas PGRI Semarang. .

Proses penyusunan tugas akhir ini tidak lepas dari hambatan dan rintangan.. Namun berkat bimbingan, bantuan nasihat, dan dorongan serta saran-saran dari berbagai pihak, khususnya Pembimbing, segala hambatan dan ringankan serta kesulitan tersebut dapat teratasi dengan baik. Oleh karena itu dalam kesempatan ini dengan tulus hati penulis sampaikan rasa terima kasih kepada :

1. Dr. Sri Suciati, M. Hum.I, selaku Rektor Universitas PGRI Semarang yang telah memberi kesempatan kepada penulis untuk menimba ilmu di Universitas PGRI Semarang.
2. Bapak Ibnu Toto Husodo, S.T., M.T. Dekan Fakultas Teknik Dan Informatika yang telah memberikan izin penulis untuk melakukan penelitian
3. Bambang Agus Herlambang, S.Kom., M.Kom. selaku Ketua Program Studi Informatika dan pembimbing I yang telah mengarahkan penulis dengan penuh ketekunan dan kecermatan
4. Bu Noora Qotrun Nada, S.T., M.Eng selaku pembimbing II yang telah membimbing penulis dengan penuh dedikasi yang tinggi
5. Bapak Ibu Dosen Program Studi Informatika yang telah memberi bekal ilmu kepada penulis selama belajar di universitas PGRI Semarang.
6. Seluruh staf SMK Negeri 5 Kendal yang telah mengizinkan peneliti melakukan penelitian.
7. Orang tua, dan seluruh keluarga yang selalu mendukung dan mendoakan penulis.

Akhirnya penulis berharap semoga tugas akhir ini bermanfaat bagi pendidik khususnya pendidik di dunia Pendidikan dan menengah.

Semarang, .....

Penulis

Muhammad      Fahrur

Rizal

NPM 20670089

## DAFTAR ISI

TUGAS AKHIR.....	i
TUGAS AKHIR.....	ii
TUGAS AKHIR.....	iii
MOTO DAN PERSEMBAHAN.....	v
PERNYATAAN KEASLIAN TULISAN.....	vi
ABSTRAK.....	vii
PRAKATA.....	vii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	1
A. Latar belakang.....	1
B. Identifikasi Masalah.....	2
C. Batasan Masalah.....	2
D. Perumusan Masalah.....	2
E. Tujuan Penelitian.....	3
F. Manfaat Penelitian.....	3
BAB II KAJIAN PUSTAKA.....	5
A. Tinjauan Pustaka.....	5
B. Landasan Teori.....	9
C. Kerangka Berfikir.....	21
BAB III METODE PENELITIAN.....	22
A. Pendekatan Penelitian.....	22

B.	Lokasi Penelitian .....	22
C.	Object dan Subject Penelitian.....	23
D.	Teknik Pengumpulan Data .....	23
E.	Analisis Data .....	24
F.	Metode Pengembangan Sistem .....	29
G.	Langkah Penelitian .....	31
BAB IV HASIL DAN PEMBAHASAN.....		32
A.	Penyajian Hasil.....	32
B.	Penerapan <i>Metode Greedy</i> .....	33
C.	Perancangan Sistem.....	37
D.	Hasil Implementasi.....	87
E.	Pengujian Sistem .....	93
F.	Pembahasan.....	93
BAB V PENUTUP .....		112
A.	Simpulan.....	112
B.	Saran.....	113
DAFTAR PUSTAKA .....		118
LAMPIRAN.....		121

## DAFTAR GAMBAR

Gambar 2.1 Metode RAD .....	12
Gambar 2.2 Kerangka Berfikir.....	22
Gambar 3.1 Lokasi Penelitian .....	23
Gambar 3.2 Greedy .....	27
Gambar 3.3 Langkah Penelitian.....	31
Gambar 4. 1 Constraint.....	34
Gambar 4. 2 Pengoptimalisasi jadwal.....	35
Gambar 4.3 Use Case Diagram.....	38
Gambar 4.4 Activity Diagram.....	39
Gambar 4.5 Mengelola data mapel .....	40
Gambar 4.6 Mengelola data mapel .....	41
Gambar 4.7 Mengelola data ruang kelas.....	42
Gambar 4.8 Mengelola data jurusan .....	43
Gambar 4.9 Megelola data waktu .....	44
Gambar 4.10 Mengelola data siswa .....	45
Gambar 4.11 Mengelola data guru.....	46
Gambar 4.12 Mengelola data jadwal .....	47
Gambar 4.13 Mengelola data absensi siswa .....	48
Gambar 4.14 Mengelola user data siswa.....	49
Gambar 4.15 User data guru .....	50
Gambar 4.16 Sequence diagram dashboard.....	51
Gambar 4.17 Sequence diagtam tambah master data.....	52
Gambar 4.18 Sequence diagram edit master data .....	52
Gambar 4.19 Sequence diagram hapus master data.....	53
Gambar 4.20 Sequence diagram tambah siswa.....	54
Gambar 4.21 Sequence diagram edit siswa.....	54
Gambar 4.22 Sequence diagram delete siswa .....	55
Gambar 4.23 Sequence diagram tambah guru .....	56
Gambar 4.24 Sequence diagram edit guru .....	56
Gambar 4.25 Sequence diagram hapus guru.....	57



Gambar 4.26 Sequence diagram tambah jadwal .....	58
Gambar 4.27 Sequence diagram edit jadwal.....	58
Gambar 4.28 Sequence diagram hapus jadwal .....	59
Gambar 4.29 Sequence diagram tambah absen siswa.....	60
Gambar 4.30 Sequence diagram edit absen siswa .....	60
Gambar 4.31 Sequence diagram delete absen siswa.....	61
Gambar 4.32 Sequence diagram tambah user data .....	62
Gambar 4.33 Sequence diagram edit user data .....	62
Gambar 4.34 Sequence diagram hapus user data.....	63
Gambar 4. 35 Class diagram .....	64
Gambar 4.36 Desain login.....	65
Gambar 4.37 Desain dashboard .....	65
Gambar 4.38 Desain tampilan master data .....	66
Gambar 4.39 Desain tambah master data.....	67
Gambar 4.40 Desain edit master data .....	67
Gambar 4.41 Desain delete master data .....	68
Gambar 4.42 Desain tampilan siswa.....	69
Gambar 4.43 Desain tambah siswa .....	69
Gambar 4.44 Desain edit siswa.....	70
Gambar 4.45 Desain delete siswa .....	71
Gambar 4.46 Desain tampilan guru .....	71
Gambar 4.47 Desain tambah guru.....	72
Gambar 4.48 Desain edit guru .....	73
Gambar 4.49 Desain delete guru.....	73
Gambar 4.50 Desain tampilan jadwal .....	74
Gambar 4.51 Desain tambah jadwal .....	75
Gambar 4.52 Desain edit jadwal .....	75
Gambar 4.53 Desain delete jadwal.....	76
Gambar 4.54 Desain tampilan absensi siswa .....	77
Gambar 4.55 Desain tampilan tambah absensi siswa .....	77
Gambar 4.56 Desain edit absensi siswa .....	78

Gambar 4.57 Desain tampilan delete absensi siswa.....	79
Gambar 4.58 Desain tampilan user data .....	79
Gambar 4.59 Desain tambah user data.....	80
Gambar 4.60 Desain edit user data .....	81
Gambar 4.61 Desain tampilan delete user data.....	81
Gambar 4.62 Desain dashboard siswa .....	82
Gambar 4.63 Desain absen pribadi siswa .....	83
Gambar 4.64 Desain tambah absen pribadi siswa.....	83
Gambar 4.65 Desain edit absen pribadi siswa .....	84
Gambar 4.66 Desain hapus data absen pribadi siswa .....	85
Gambar 4.67 Desain dashboard guru .....	85
Gambar 4.68 Desain wali kelas melihat absensi siswa .....	86
Gambar 4. 69 Tampilan Login .....	87
Gambar 4. 70 Tampilan Dashboard Admin.....	88
Gambar 4. 71 Tampilan Master Data .....	88
Gambar 4. 72 Tampilan Data Siswa.....	89
Gambar 4. 73 Tampilan Data Guru .....	89
Gambar 4. 74 Tampilan Jadwal Kelas.....	90
Gambar 4. 75 Tampilan Absen Siswa .....	90
Gambar 4. 76 Tampilan Data Pengguna.....	91
Gambar 4. 77 Tampilan Dashboard Siswa .....	91
Gambar 4. 78 Tampilan Absen Siswa .....	92
Gambar 4. 79 Tampilan Dashboard Guru .....	92
Gambar 4. 80 Tampilan Wali Kelas Melihat Absen Siswa .....	93
Gambar 4. 81 Flowgraph.....	98

## **DAFTAR TABEL**

Tabel 2.1 Tinjauan Pustaka .....	5
Tabel 2.2 Use Case .....	14
Tabel 2.3 Activity Diagram .....	16
Tabel 2.4 Sequence Diagram.....	17
Tabel 2.5 Class Diagram .....	18
Tabel 4. 1 Cara Kerja Greedy .....	36
Tabel 4. 2 Whitebox Testing.....	94
Tabel 4. 3 Blackbox Testing .....	100
Tabel 4. 4 User Acceptance Testing .....	108

# **BAB I**

## **PENDAHULUAN**

### **A. Latar belakang**

Perkembangan pesat teknologi informasi telah mengubah dunia menjadi lebih efisien dan maju dalam berbagai bidang. Dukungan teknologi komputer telah terbukti mampu meningkatkan efektivitas dan efisiensi dalam mekanisme kerja yang sebelumnya panjang dan berulang. Komputer sangat penting untuk mengorganisasi dan mengelola data dengan sistem basis data, yang telah menjadi dasar bagi banyak organisasi untuk beroperasi. Informasi menjadi elemen yang sangat penting bagi manajemen dalam pengambilan keputusan, dan sistem informasi menjadi sumber utama untuk mendapatkan informasi tersebut. Perkembangan informasi yang pesat menuntut instansi-instansi untuk melakukan perubahan dalam sistem yang mereka miliki, mengingat bahwa masalah yang dihadapi juga semakin berkembang [1].

Salah satu instansi yang sering menghadapi berbagai masalah adalah instansi pendidikan. Masalah yang sering timbul berkaitan dengan sistem informasi, seperti lamanya waktu yang dibutuhkan dalam proses pengolahan data, kurangnya akurasi data, atau bahkan kehilangan data yang diperlukan. Oleh karena itu, peningkatan kualitas instansi pendidikan menjadi suatu tuntutan, dan salah satu langkah yang dapat diambil adalah dengan mengembangkan sistem informasi akademik yang lebih baik. Dengan adanya sistem informasi akademik yang baik, proses pengolahan data dan pelayanan terhadap siswa dapat meningkat, sehingga menjadi keunggulan yang menarik bagi instansi pendidikan tersebut [2].

Sistem Informasi Akademik sangat penting untuk mengelola proses administrasi dan data akademik di lembaga pendidikan. Penggunaan teknologi komputer dan framework pengembangan aplikasi menjadi semakin penting di era digital saat ini [3]. Tujuan pengembangan modul administrasi di SISKEMA adalah untuk meningkatkan pengelolaan data akademik dan

pelayanan kepada bagian administrasi dengan menggunakan framework Laravel. Framework ini dipilih karena memiliki banyak fitur yang kuat, mudah digunakan, dan telah terbukti efektif dalam pengembangan aplikasi web. Dengan menggunakan framework ini, diharapkan pengembangan modul administrasi dapat dilakukan dengan lebih cepat dan efisien.

Selain itu, tujuan pembuatan modul administrasi adalah untuk meningkatkan pengalaman pengguna (user experience) administrator sistem. Dengan modul administrasi yang lebih baik, administrator akan lebih mudah mengakses dan mengelola data akademik, yang membuat proses administrasi lebih efisien dan mengurangi kesalahan input data.

## **B. Identifikasi Masalah**

Belum dikembangkannya sistem penjadwalan yang terkomputerisasi, yang menimbulkan kesulitan dalam mengelola jadwal kelas, guru, dan ruangan secara efisien. Ketiadaan sistem ini menyebabkan proses pengelolaan data dilakukan secara manual dan rentan terhadap kesalahan

## **C. Batasan Masalah**

Agar pengembangan Sistem Informasi Akademik di SMK Negeri 5 Kendal lebih terfokus dan terarah, maka batasan masalah yang ditetapkan hanya akan berfokus pada pengolahan data akademik seperti pengelolaan data guru, murid dan penjadwalan. Dalam penjadwalan mata pelajaran algoritma Greedy akan digunakan untuk mengoptimalkan dan memastikan alokasi waktu, ruang kelas, dan guru yang efektif. Penelitian ini hanya akan sampai pada tahap construction saja dan tidak mencakup proses cutover sesuai dengan metode Rapid Application Development (RAD).

## **D. Perumusan Masalah**

Berdasarkan identifikasi dan batasan masalah yang telah ditetapkan, maka perumusan masalah dalam pengembangan Sistem Optimalisasi Penjadwalan ini adalah:

- a. Bagaimana mengembangkan sistem penjadwalan yang dapat meningkatkan efisiensi dan akurasi dalam pengolahan data akademik di SMK Negeri 5 Kendal?
- b. Bagaimana mengimplementasikan algoritma *Greedy* dalam sistem penjadwalan mata Pelajaran SMK Negeri 5 Kendal?

#### **E. Tujuan Penelitian**

Adapun Tujuan dari Penelitian ini :

- a. Meningkatkan pemahaman tentang penggunaan sistem informasi akademik berbasis web di lingkungan SMK Negeri 5 Kendal.
- b. Mengidentifikasi masalah yang dihadapi dalam penggunaan sistem penjadwalan berbasis web di SMK Negeri 5 Kendal.
- c. Mengevaluasi seberapa efektif dan efisien sistem penjadwalan berbasis web mendukung proses administrasi di SMK Negeri 5 Kendal.

#### **F. Manfaat Penelitian**

Manfaat Penelitian untuk Mahasiswa, Universitas, dan Pembaca:

1. Untuk Mahasiswa:
  - a. Penelitian membantu mahasiswa mengembangkan keterampilan analitis, kritis, dan pemecahan masalah yang penting untuk karier mereka.
  - b. Melakukan penelitian memberikan pengalaman praktis dalam metodologi penelitian, pengumpulan data, dan analisis yang berharga di dunia kerja.
  - c. Penelitian dapat meningkatkan prospek karier mahasiswa dengan menambahkan pengalaman berharga ke CV mereka dan membangun jaringan profesional.
2. Untuk Universitas :
  - a. Penelitian diharapkan dapat meningkatkan reputasi akademik universitas dan menarik lebih banyak mahasiswa.

- b. Penelitian sering melibatkan kolaborasi antar-instansi dan dengan institusi lain, yang memperkuat hubungan dan membuka peluang baru.
3. Untuk Pembaca
- a. Penelitian yang sukses meningkatkan reputasi akademik universitas dan menarik lebih banyak mahasiswa dan dana penelitian.
  - b. Penelitian sering melibatkan kolaborasi antar-departemen dan dengan institusi lain, yang memperkuat hubungan dan membuka peluang baru.

## BAB II KAJIAN PUSTAKA

### A. Tinjauan Pustaka

Penelitian ini akan mendukung temuan penelitian sebelumnya dari sumber ilmiah seperti tugas akhir, jurnal penelitian, dan tesis. Penelitian berikut akan menjadi acuan untuk penelitian ini:

Tabel 2.1 Tinjauan Pustaka

No.	Penulis	Judul	Metode	Hasil Penelitian
1.	Bernardus Willson (2023)	Penerapan Algoritma Greedy pada Perancangan Jadwal Kuliah yang Optimal dengan Memperhatikan Preferensi Dosen	Greedy	Penelitian ini berhasil menyusun jadwal kuliah yang efektif dan memenuhi keinginan dosen,
2.	Mia Fitriawati (2018)	Analisis Dan Pemodelan Prototipe Sistem Informasi Jadwal Kuliah Dengan Algoritma ABC (Artificial Bee Colony)	Artificial Bee Colony	Penelitian ini berhasil membangun sistem penjadwalan otomatis yang dapat meminimalisir bentrok jadwal dengan proses penjadwalan yang cepat.
3.	Restie Maya Puspita (2016)	Pengembangan Aplikasi Penjadwalan Kegiatan Pelatihan Teknologi Informasi Dan Komunikasi Dengan	Genetika	Algoritma Genetika pada aplikasi optimasi penjadwalan kegiatan menghasilkan kinerja yang lebih optimal



No.	Penulis	Judul	Metode	Hasil Penelitian
		Menggunakan Algoritma Genetika		pada proses pemilihan dan pengolahan data.
4.	Muhammad Muhardeny (2023)	Penjadwalan Mata Pelajaran Menggunakan Algoritma Particle Swarm Optimization (PSO) Pada SMPIT Mufidatul Ilmi	Particle Swarm Optimization	Penelitian ini berhasil mengoptimalkan ruangan untuk jadwal Pelajaran.
5.	Akhmad Roihan (2022)	Implementasi Algoritma Greedy Kombinasi dengan Perulangan pada Aplikasi Penjadwalan Praktikum	Greedy	Penelitian ini berhasil menyusun jadwal praktikum dengan lebih efisien dan menghindari bentrok jadwal
6.	Ela Siti Laela (2023)	Optimasi Algoritma Greedy Dalam Penyusunan Jadwal Pelajaran Pada SMK Nurul Islam Cianjur	Greedy	Penelitian ini berhasil mengoptimalkan ruangan untuk jadwal Pelajaran.

Berikut merupakan uraian antara penelitian yang telah dilakukan sebelumnya dengan penelitian yang akan dilaksanakan oleh penulis :

Penelitian mengenai “Penerapan Algoritma *Greedy* pada Perancangan Jadwal Kuliah yang Optimal dengan Memperhatikan Preferensi Dosen” yang dilakukan oleh Bernardus Willson (2023). Penelitian ini membahas bagaimana menciptakan jadwal kuliah yang ideal yang mempertimbangkan preferensi dosen. Tujuan utama pendekatan algoritma *greedy* adalah untuk meminimalkan tumpang tindih waktu antara mata kuliah dan memenuhi preferensi dosen. Selain itu, efisiensi komputasional diperhatikan dalam

penelitian ini untuk mengatasi kompleksitas perancangan jadwal kuliah. Diharapkan hasilnya akan membantu admin sistem pendidikan menyusun jadwal kuliah yang efektif, memenuhi keinginan dosen, dan meningkatkan pengalaman belajar mahasiswa [4]

Penelitian mengenai “Analisis Dan Pemodelan Prototipe Sistem Informasi Jadwal Kuliah Dengan Algoritma ABC (Artificial Bee Colony)” yang dilakukan oleh Mia Fitriawati (2018). Metode pengembangan sistem prototipe dan pendekatan berorientasi objek adalah metode penelitian yang digunakan. Penelitian ini menghasilkan gambaran sistem dan model penerapan algoritma ABC pada sistem informasi jadwal kuliah. Model ini diharapkan dapat berfungsi sebagai referensi untuk tahap awal pembangunan sistem informasi jadwal kuliah, yang memungkinkan proses penjadwalan otomatis yang meminimalkan bentrok jadwal dengan kecepatan yang cepat [5].

Penelitian mengenai “Pengembangan Aplikasi Penjadwalan Kegiatan Pelatihan Teknologi Informasi Dan Komunikasi Dengan Algoritma Genetika” yang dilakukan oleh Restie Maya Puspita (2016). Tujuan dari penelitian ini adalah untuk membuat aplikasi yang menggunakan Algoritma Genetika (AG) untuk mengoptimalkan penjadwalan kegiatan di Balai Pelatihan Riset Teknologi Informasi dan Komunikasi (BPRTIK), yang terletak di Ciputat, Tangerang Selatan. Penelitian ini menerapkan metode Rapid Application Development (RAD) yang mencakup fase perencanaan, desain perancangan, dan fase pelaksanaan dalam proses pengembangan system aplikasi. System dikembangkan menggunakan Bahasa pemrograman PHP dan MySQL sebagai platform utama, Adobe Dreamweaver CS6 dan Microsoft Visio 2007 sebagai infrastruktur grafis. Penggunaan Algoritma Genetika untuk aplikasi penjadwalan kegiatan optimasi menghasilkan proses pemilihan dan pengolahan data yang lebih optimal pada aplikasi yang telah dikelola. Ini menghasilkan produk penjadwalan yang lebih terstruktur [6].

Penelitian mengenai “Penjadwalan Mata Pelajaran Menggunakan Algoritma Particle Swarm Optimization (PSO) Pada SMPIT Mufidatul Ilmi” yang di

lakukan oleh Muhammad Muhardeny (2023). pengembangan aplikasi menggunakan metode Waterfall. Penelitian ini bertujuan untuk membuat aplikasi penjadwalan pembelajaran di SMPIT MufidatulIlmi dengan menggunakan algoritma pengoptimalan panas partikel (PSO) untuk menyusun jadwal pembelajaran. Particle Swarm Optimization adalah algoritma berbasis populasi yang mengeksplorasi individu dalam pencarian. Dalam algoritma ini, populasi disebut swarm dan individu disebut partikel. Tiap partikel bergerak dengan kecepatan yang disesuaikan dengan daya tarik pencarian dan menyimpannya di posisi terbaik yang pernah dicapai. Hasil penelitian ini memberikan beberapa fitur (layanan) data primer, terutama fitur untuk memberikan hasil penjadwalan dari pemrosesan dengan algoritma PSO [7].

Penelitian mengenai “Implementasi Metode *Greedy* Best First Search Untuk Penjadwalan Perkuliahan” yang dilakukan oleh Yuggo Afrianto (2023). Penelitian ini berfokus untuk membuat aplikasi penjadwalan praktikum laboratorium untuk Program Studi Teknik Informatika Fakultas Teknik UISU. Aplikasi ini memiliki data seperti dua ruangan yang digunakan, lima dosen yang mengajar, dan dua belas mata kuliah praktikum yang dipraktikkan. Penelitian ini bertujuan untuk membuat aplikasi penjadwalan yang akan membantu praktikum laboratorium Teknik Informatika membuat jadwal yang lebih mudah dengan menggunakan algoritma *greedy* yang dikombinasikan dengan perulangan untuk optimalisasi pembuatan jadwal. Penelitian ini menghasilkan aplikasi yang dapat digunakan di Laboratorium Program Studi Teknik Informatika. Aplikasi ini akan membantu kepala laboratorium menyusun jadwal praktikum dengan lebih efisien dan menghindari bentrok jadwal [8].

Penelitian mengenai “Optimasi Algoritma *Greedy* Dalam Penyusunan Jadwal Pelajaran Pada SMK Nurul Islam Cianjur” yang dilakukan oleh Ela Siti Laela (2023). Penelitian ini bertujuan untuk menentukan apakah algoritma *greedy* dapat mengoptimalkan pembuatan Sistem Penjadwalan Mata Pelajaran di SMK Nurul Islam Cianjur dan mengetahui jumlah ruangan kelas yang kosong

setelah penggunaan algoritma tersebut. Pengujian dilakukan dengan membandingkan hasil sistem yang menggunakan algoritma greedy dengan hasil sistem manual. Menurut data jadwal pelajaran yang dimasukkan ke dalam sistem, algoritma greedy memiliki lebih banyak ruangan kosong. Ini menunjukkan bahwa algoritma ini mampu mengoptimalkan ruangan untuk jadwal pelajaran di SMK Nurul Islam Cianjur [9].

## **B. Landasan Teori**

Teori sangat penting untuk penelitian karena membantu menentukan jawaban dari rumusan masalah penelitian. Penulis penelitian ini akan menggunakan teori yang mereka pelajari di kuliah sebagai landasan untuk penelitian ini.

### **1. Penjadwalan**

Penjadwalan adalah proses menentukan kapan suatu kegiatan atau serangkaian kegiatan akan dilakukan. Penjadwalan dalam dunia akademik sering merujuk pada penentuan jadwal pelajaran, ujian, dan kegiatan akademik lainnya. Penjadwalan yang efektif sangat penting untuk memastikan bahwa sumber daya yang tersedia, seperti pengajar dan ruang, digunakan secara optimal [10].

### **2. Mata Pelajaran**

Di institusi pendidikan, mata pelajaran adalah unit pembelajaran yang diajarkan. Setiap mata pelajaran memiliki kurikulum yang harus diikuti. Mata pelajaran merupakan komponen inti dari sistem pendidikan dan sangat penting dalam pembentukan jadwal akademik.

### **3. Sistem**

Sistem adalah seperangkat komponen yang saling berhubungan dan saling bekerja sama untuk mencapai beberapa tujuan. Selain itu, sistem terdiri dari komponen, masukan (input), pengolahan (processing), dan keluaran (output) [11].

#### 4. Informasi

Data yang telah diproses sehingga memiliki makna bagi orang yang melihatnya disebut informasi. Dalam sistem informasi akademik, informasi ini mencakup informasi tentang siswa, mata pelajaran, jadwal, nilai, dan aspek akademik lainnya.

#### 5. Akademik

Akademik adalah bidang yang mempelajari kurikulum atau pembelajaran untuk meningkatkan pengetahuan. Sekolah atau lembaga pendidikan dapat mengelolanya [12].

#### 6. Sistem Informasi Akademik

Sistem Informasi Akademik adalah sistem yang digunakan untuk mengelola data dan informasi yang berkaitan dengan kegiatan akademik di institusi pendidikan. Sistem ini bertujuan untuk mempermudah pengelolaan data mahasiswa, jadwal, nilai, dan informasi akademik lainnya. Sistem ini menggunakan teknologi komputer untuk mengelola proses kegiatan akademik [13].

#### 7. Algoritma *Greedy*

Pada awalnya, istilah "algoritma" merujuk pada aturan aritmetis untuk menyelesaikan masalah dengan bilangan Arab. Namun, pada abad ke-18, artinya berkembang menjadi urutan langkah atau prosedur yang jelas yang diperlukan untuk menyelesaikan masalah [9]. Algoritma *Greedy* adalah jenis algoritma yang memilih tindakan paling optimal pada setiap langkah, dengan harapan akan menghasilkan Solusi yang optimal. Optimasi sendiri adalah masalah yang membutuhkan pencarian solusi terbaik dari serangkaian solusi alternatif.

#### 8. Algoritma Penunjang

##### 1. Algoritma *Artificial Bee Colony* (ABC)

Algoritma *Artificial Bee Colony* (ABC) adalah algoritma *Metaheuristic* yang meniru perilaku koloni lebah dalam mencari

makanan. Algoritma ABC membagi koloni lebah menjadi tiga kategori, yaitu :

1. *Scout Bees*

Scout Bees adalah kelompok lebah yang bertanggung jawab untuk mencari sumber makanan di sekitar sarang mereka. Scout Bees akan mencari populasi (kumpulan solusi atau potensial) yang terbaik dan memeriksa apakah memenuhi persyaratan yang diberikan.

2. *Employed Bees*

Lebah Pekerja adalah kelompok lebah yang memiliki peran penting dalam berinteraksi dengan sumber makanan yang telah ditemukan Scout Bees. Tugas utama lebah Pekerja adalah mengumpulkan dan menyimpan informasi mengenai setiap sumber makanan yang ditemukan. Dalam fase ini lebah menemukan jadwal kuliah yang optimal dengan melihat solusi baru di sekitar solusi lama, mengevaluasinya, dan menggantinya jika solusi baru lebih baik.

3. *Onlooker Bees*

Lebah pengamat bertanggung jawab untuk memilih dan memanfaatkan sumber makanan mereka, yang telah disimpan sebelumnya. Setelah menghitung kemungkinan yang ada, langkah berikutnya adalah mengubah slot untuk membuat populasi baru. Ini dilakukan melalui pertukaran slot yang mengalami konflik, yang dipilih secara acak berdasarkan indeks slot hari dan waktu. Nilai kecocokan populasi yang diubah juga dihitung ulang. Hasilnya kemudian dibandingkan dengan nilai kecocokan populasi terbaik yang telah disimpan sebelumnya. Jika nilai kecocokan populasi baru lebih tinggi atau lebih rendah dari nilai kecocokan populasi terbaik yang telah disimpan

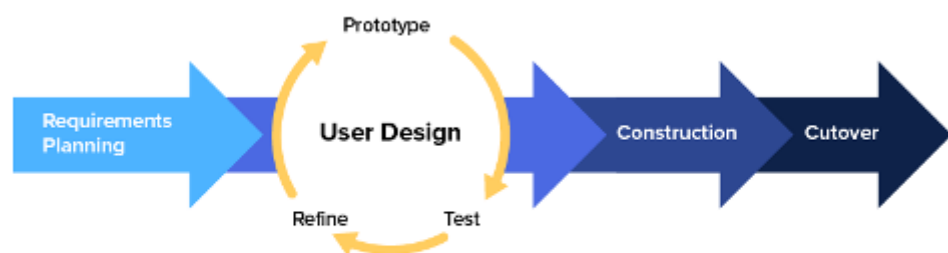
sebelumnya, populasi baru akan menggantikan populasi terbaik yang telah disimpan sebelumnya [14].

## 2. Algoritma *Genetika*

Algoritma *Genetika* (GA) adalah metode optimasi komputasi yang terinspirasi oleh genetika dan prinsip seleksi alam. Mereka digunakan untuk memecahkan masalah yang rumit dengan meniru proses evolusi dan berulang kali meningkatkan populasi ssolusi potensial. Algoritme ini bekerja dengan sekumpulan solusi potensial yang dikodekan dalam bentuk rangkaian digit biner atau struktur data lainnya. Algoritma ini memiliki kemampuan untuk mengoptimumkan solusi untuk berbagai masalah yang dihadapi, salah satunya adalah sistem penjadwalan [15]. Sistem penjadwalan menggunakan diagram alir siklus algoritma genetika, yang mencakup pembangkitan populasi awal, penilaian kekuatan, seleksi *individu*, *crossover*, *mutasi*, dan *regenerasi*.

## 9. Metode *Rapid Application Development* (RAD)

*Rapid Application Development* (RAD) menekankan pada siklus pengembangan yang cepat dan singkat. Model sistem yang dibangun dapat dilihat pada gambar 2.1 berikut :



Gambar 2.1 Metode RAD

### 1. Requirment Planing

Pada tahap ini, pengguna dan analis berkumpul untuk menentukan tujuan sistem yang akan dibangun. Selanjutnya, pengguna akan

menentukan syarat-syarat informasi yang diperlukan untuk mencapai tujuan tersebut, dan terakhir, akan dilakukan evaluasi semua sistem yang dibutuhkan oleh pengguna [16].

## 2. User Design

Pada tahap ini, pengembang akan membuat model desain yang mencakup diagram alur dan struktur data serta prototipe awal untuk menerima umpan balik pengguna dan memastikan desain memenuhi persyaratan.

## 3. Contruction

Tahap ini adalah langkah awal dalam pembuatan sistem yang telah direncanakan. Proses ini melibatkan penulisan kode program, atau yang sering disebut sebagai coding, dengan tujuan mengubah desain sistem yang telah dirancang menjadi sebuah aplikasi yang dapat digunakan sesuai dengan perencanaan [17].

## 4. Cutover

Pada tahap ini, sistem yang telah dibuat diimplementasikan. Selanjutnya, sistem yang baru diuji untuk mengetahui apakah berjalan dengan baik atau memerlukan perbaikan.

## 10. UML

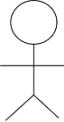




Unified Modeling Language (UML) adalah standar untuk memodelkan sistem perangkat lunak. UML menyediakan berbagai diagram untuk memvisualisasikan, menentukan, dan mendokumentasikan artefak dari sistem perangkat lunak [18]. Berikut beberapa jenis diagram UML:

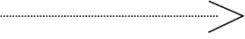

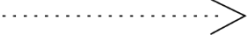


### 1. Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan interaksi antara pengguna (aktor) dan sistem. Use case menjelaskan bagaimana satu atau lebih aktor berinteraksi dengan sistem informasi yang akan dibangun. Secara umum, use case digunakan untuk mengetahui fungsi apa yang ada di dalam sistem dan siapa yang berhak menggunakan fungsi tersebut [19]. Berikut beberapa atribut yang digunakan pada *Use Case Diagram*



Tabel 2.2 Use Case

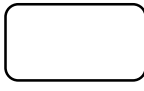



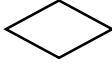
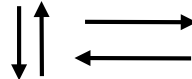
GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Aktor menunjukkan orang yang menggunakan sistem yang dimodelkan, seperti pengguna. Aktor dapat terdiri dari individu, sistem eksternal, atau perangkat.
	<i>System</i>	Batasan di mana use case dan interaksi dengan aktor terjadi dikenal sebagai sistem.
	<i>Use Case</i>	Aktor ingin melakukan tugas atau tujuan tertentu dengan sistem.
	<i>Collaboration</i>	Kolaborasi merujuk pada bagaimana beberapa aktor dan use casesaling bekerja sama.
	<i>Note</i>	Catatan memberikan informasi tambahan atau penjelasan lebih lanjut yang relevan dengan use case atau diagram secara keseluruhan.

GAMBAR	NAMA	KETERANGAN
	<i>Dependency</i>	Dependency menunjukkan hubungan ketergantungan antara dua elemen.
	<i>Generalization</i>	Generalization menggambarkan hubungan pewarisan antara elemen-elemen.
	<i>Include</i>	Include menunjukkan bahwa use case tertentu selalu menyertakan perilaku dari use case lain.
	<i>Extend</i>	Extend menunjukkan bahwa use case tertentu dapat memperluas perilaku dari use case lain dalam kondisi tertentu.
	<i>Association</i>	Association menunjukkan hubungan antara aktor dan use case, menggambarkan interaksi mereka.

## 2. Activity Diagram

*Activity Diagram* menunjukkan alur kerja (workflow) atau aktivitas sistem atau proses bisnis. Berikut beberapa atribut yang digunakan pada *Activit Diagram* :

Tabel 2.3 Activity Diagram


NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Activity adalah kumpulan dari satu atau lebih tindakan yang dilakukan dalam proses.
2		<i>Action</i>	Action adalah langkah spesifik atau pekerjaan yang dilakukan dalam aktivitas.
3		<i>Initial Node</i>	Initial node adalah titik awal dari sebuah alur kerja atau proses dalam Activity Diagram.
4		<i>Activity Final Node</i>	Activity final node adalah titik akhir dari sebuah alur kerja atau proses.
5		<i>Decision</i>	Decision node digunakan untuk menunjukkan titik di mana alur kerja bercabang berdasarkan kondisi tertentu.
6		<i>Line Connector</i>	Line connector adalah garis yang menghubungkan elemen-elemen dalam diagram aktivitas, menunjukkan alur proses dari satu tindakan ke tindakan berikutnya.

### 3. Sequence Diagram

*Sequence diagram* digunakan untuk menggambarkan perilaku aktor pada sebuah sistem secara detail menurut waktu dan menunjukkan interaksi antara objek dalam urutan waktu tertentu. Diagram ini juga menunjukkan pesan yang dikirim antara objek untuk menyelesaikan suatu fungsi, yang membantu memahami bagaimana bagian-bagian

sistem berinteraksi satu sama lain dalam skenario waktu [20]. Berikut beberapa atribut yang digunakan pada *Sequence Diagram* :


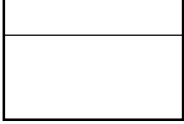
Tabel 2.4 Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambar orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form.
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel.
5		<i>A Focus of Control and A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message.
6		<i>A Message</i>	Menggambarkan pengiriman pesan.

#### 4. Class Diagram

Class Diagram menampilkan struktur kelas dalam sistem, termasuk atribut dan metode setiap kelas serta hubungan antar kelas seperti asosiasi, generalisasi, dan dependensi. Diagram ini penting untuk memodelkan struktur data dan desain sistem berbasis objek. Berikut beberapa atribut yang digunakan pada *Class Diagram* :

Tabel 2.5 Class Diagram

SIMBOL	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana hubungan anak berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk
	<i>Class</i>	Himpunan dari objek-objek yang berbagi attribute serta operasi yang sama

## 11. Tools Penunjang System

### 1. Xampp

Xampp merupakan Software server Apache XAMPP menawarkan database server seperti MySQL dan PHP programming. Keunggulan XAMPP adalah cukup mudah dioperasikan, tidak membutuhkan biaya, dan mendukung instalasi pada Windows dan Linux. Keuntungan lain dari XAMPP adalah hanya perlu melakukan instalasi sekali saja dan memiliki MySQL, apacheweb server, database server PHP, dan modul lainnya [21].

### 2. Visual Studi Code

Visual Studio Code adalah editor kode sumber yang dibuat oleh Microsoft untuk Windows, Linux, dan mac OS. Ini memudahkan penulisan kode dan mendukung berbagai bahasa pemrograman. Selain itu, warnanya dapat disesuaikan sesuai dengan fungsi rangkaian kode. Selain itu, fitur tambahan adalah kemampuan untuk menambah ekstensi, yang memungkinkan pengembang untuk menambah fitur yang tidak ada di Visual Studio Code. Karena Visual Studio Code bersifat open source, source code dapat dilihat oleh siapapun yang ingin berkontribusi pada pengembangan aplikasi, dan kode dapat

diakses melalui link github. Ini membuat Visual Studio Code memiliki banyak penggemar dalam pengembangan aplikasi [22].

### 3. Github

Github adalah program hosting open source yang digunakan oleh tool Git dan juga berfungsi sebagai webhosting. Git sendiri merupakan alat pengendali sistem yang berfungsi untuk mengontrol kode bahasa pemrograman. Penelitian ini bertujuan untuk mengetahui seberapa efektif github sebagai media e-learning untuk mengetahui seberapa efektif proyek berkolaborasi karena github memudahkan pengembangan proyek [23].

### 4. PHP

PHP adalah bahasa pemrograman server-side yang populer untuk pengembangan aplikasi web dinamis. PHP dapat diintegrasikan dengan berbagai database, seperti MySQL, dan sering digunakan bersama HTML, CSS, dan JavaScript untuk membuat aplikasi web yang interaktif dan responsive.

### 5. MySql

MySQL adalah perangkat lunak dengan penggunaan global sekitar 6 juta, dan merupakan sistem manajemen database SQL (database management system) atau DBMS multithread dan multiuser. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis melalui lisensi GNU General Public License (GPL) [21].

### 6. Laravel

Laravel adalah framework PHP berbasis web gratis yang dikembangkan oleh Taylor Otwell dan ditujukan untuk pengembangan aplikasi web yang menggunakan pola arsitektur model-view-controller (MVC). Laravel memiliki banyak fitur, seperti pengembangan sistem modul-modul yang dapat diatur, berbagai cara untuk mengakses database relasional, utilitas yang memudahkan penyebaran aplikasi, dan kemudahan pemeliharaan [3].

## 12. Black Box Testing

Black Box Testing adalah pengujian yang berfokus pada spesifikasi fungsional perangkat lunak; tester dapat mendefinisikan kumpulan kondisi input dan kemudian melakukan pengetesan pada spesifikasi fungsional program [24].

## 13. White Box Testing

Pengujian kotak putih adalah pengujian yang didasarkan pada pengujian detail perancangan dan menggunakan struktur kontrol desain program untuk membagi pengujian ke dalam beberapa kasus pengujian. Kesimpulan dari pengujian kotak putih menunjukkan bahwa Anda dapat mendapatkan program yang benar sepenuhnya [24]. Pengujian white-box berbeda dengan pengujian black-box karena tidak hanya berfokus pada input dan output, tetapi juga pada proses dan alur dari input hingga output dalam suatu sistem atau perangkat lunak. Oleh karena itu, penguji perlu memiliki pemahaman tentang bahasa pemrograman yang digunakan dan kemampuan untuk memeriksa struktur kode sistem atau perangkat lunak yang diuji. Hasil dari pengujian ini mencakup pemahaman tentang alur data, struktur kode, serta identifikasi potensi kegagalan dalam sistem atau perangkat lunak tersebut. Salah satu metrik yang digunakan dalam pengujian white-box adalah kompleksitas siklomatik ( $V(G)$ ). Metrik ini digunakan untuk menghitung jumlah jalur independen dalam kode yang harus diuji guna memastikan cakupan pengujian yang memadai. Rumus untuk menghitung kompleksitas siklomatik adalah :

$$V(G) = E - N + 2P$$

Dimana :

$V(G)$  adalah kompleksitas siklomatik.

$E$  adalah jumlah edge (sisi) dalam grafik alur kontrol.

$N$  adalah jumlah node (simpul) dalam grafik alur kontrol.

$P$  adalah jumlah komponen terhubung atau jumlah program (biasanya  $P=1$  untuk program tunggal).

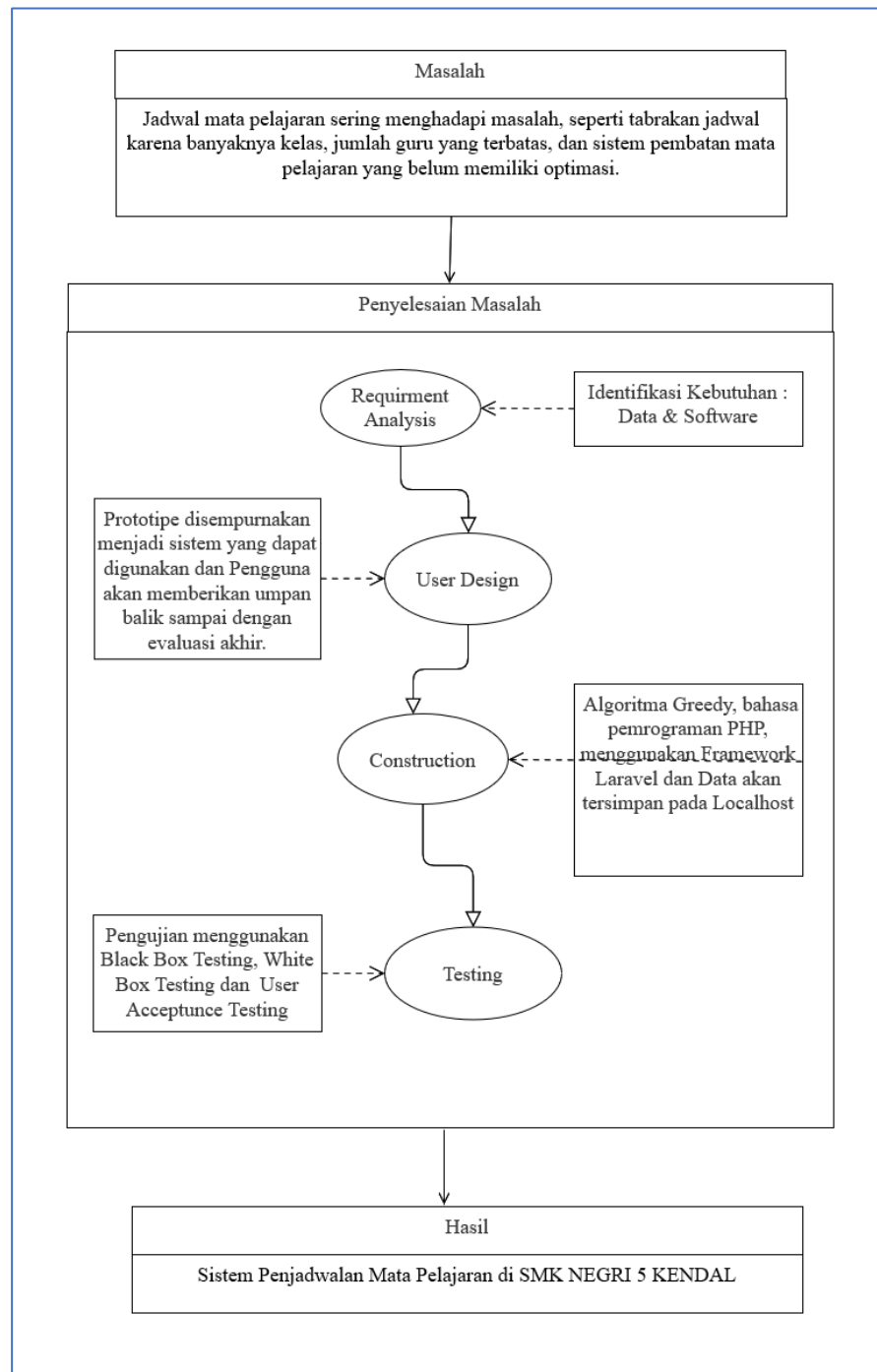
#### 14. User Acceptance Testing (UAT)

User Acceptance Testing (UAT) adalah tahap akhir dalam proses pengujian perangkat lunak sebelum aplikasi atau sistem diresmikan untuk digunakan oleh pengguna akhir. UAT dilaksanakan pada akhir proses pengujian saat sistem siap digunakan. Tujuan utamanya adalah untuk mengembangkan perangkat lunak yang mampu memenuhi kebutuhan pengguna. Bukan hanya sekedar memenuhi spesifikasi sistem dan dapat digunakan saja, tetapi juga untuk memvalidasi apakah sistem dapat diterima atau tidak [25].

#### **C. Kerangka Berfikir**

Kerangka berfikir berfungsi sebagai model konseptual yang menjelaskan bagaimana teori berinteraksi dengan berbagai elemen yang telah ditentukan sebagai masalah penting. Gambar 2.1 menunjukkan rangka kerja penelitian:





Gambar 2.2 Kerangka Berfikir

## **BAB III**

### **METODE PENELITIAN**

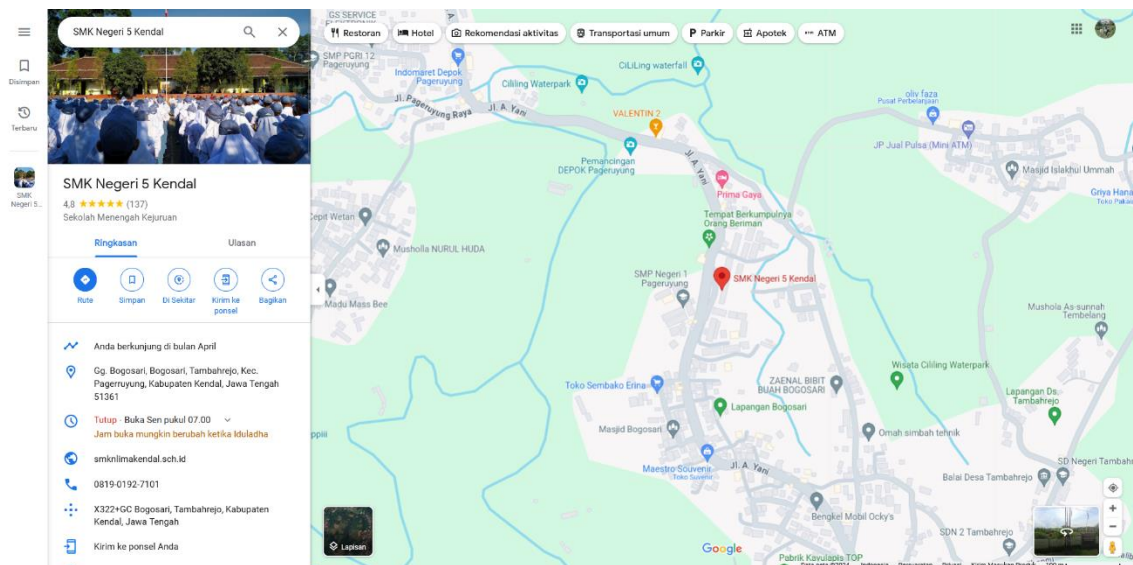
#### **A. Pendekatan Penelitian**

Penelitian tugas akhir ini menggunakan pendekatan penelitian *Rapid Application Development* (RAD). RAD merupakan metode penelitian dengan pendekatan pengembangan perangkat lunak yang cepat dan iteratif. Pendekatan ini menekankan pada siklus pengembangan yang cepat, memungkinkan perubahan dan penyesuaian yang terus-menerus berdasarkan umpan balik pengguna. Metode ini dilakukan dengan pendekatan yang fokus pada pembuatan prototipe dan mendapatkan umpan balik secara berulang untuk memperbaiki dan menyempurnakan aplikasi secara bertahap.

Dalam penelitian ini, teknik analisis data yang digunakan adalah analisis deskriptif. Analisis deskriptif digunakan untuk menggambarkan dan meringkas data yang diperoleh dari berbagai instrumen pengumpulan data seperti wawancara dan observasi. Penelitian ini menggunakan Algoritma *Greedy*. Algoritma *Greedy* adalah pendekatan yang digunakan untuk menyelesaikan masalah optimasi dengan cara membuat serangkaian pilihan yang tampak paling baik pada saat itu tanpa mempertimbangkan konsekuensi di masa depan. Dalam penerapannya untuk optimasi jadwal pelajaran, Algoritma *Greedy* bekerja dengan memilih slot waktu dan ruang kelas yang tersedia yang paling optimal untuk setiap mata pelajaran, berdasarkan preferensi guru dan ketersediaan, sehingga membentuk jadwal yang memenuhi berbagai kriteria dan kendala seperti menghindari bentrok jadwal dan mematuhi preferensi waktu pengajaran.

#### **B. Lokasi Penelitian**

SMK Negeri 5 Kendal Merupakan objek lokasi yang digunakan dalam penelitian ini. SMK Negeri 5 Kendal berlokasi di Kelurahan Tambahrejo, Kecamatan Pageruyung, Kabupaten Kendal, Jawa Tengah 51361. Peta lokasi SMK Negeri 5 Kendal dapat dilihat pada gambar 3.1 berikut:



Gambar 3.1 Lokasi Penelitian

### C. Object dan Subject Penelitian

Objek penelitian ini adalah metode penjadwalan di SMK Negeri 5 Kendal. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi metode penjadwalan yang efektif untuk mendukung kegiatan akademik dan administratif di sekolah. Metode penjadwalan ini mencakup aspek seperti mengatur jadwal pelajaran dan alokasi ruang kelas. Sedangkan subjek penelitian yang terlibat dalam proses pembuatan jadwal, yaitu kepala bagian kurikulum SMK Negeri 5 Kendal.

### D. Teknik Pengumpulan Data

#### 1. Jenis dan Sumber Data

##### a. Data Primer

Data yang dikumpulkan langsung dari sumbernya, termasuk wawancara dengan pemangku kepentingan dan observasi langsung di lokasi penelitian.

##### b. Data Sekunder

Data yang diperoleh dari dokumen, literatur, buku, arsip maupun informasi lainnya yang berhubungan dengan SMK Negeri 5 Kendal dan sistem penjadwalan mata pelajaran menggunakan algoritma *Greedy*.

## 2. Instrumen Pengumpulan Data

### a. Studi Kepustakaan

Untuk memahami teori dan konsep yang mendukung penelitian ini, studi kepustakaan memerlukan penelitian literatur yang relevan. Berbagai sumber, termasuk buku, artikel ilmiah, jurnal, dan dokumen, diselidiki terkait penjadwalan di institusi pendidikan. Fokusnya adalah mendapatkan informasi mengenai penelitian sebelumnya tentang penjadwalan mata pelajaran menggunakan algoritma Greedy. Ini termasuk studi kasus, hasil, dan evaluasi dari peneliti lain. Tujuannya adalah untuk memahami bagaimana algoritma ini dapat digunakan untuk penjadwalan mata pelajaran dan hasilnya, serta *best practices* dan masalah yang terkait dengan penerapan algoritma untuk menyesuaikannya dengan kebutuhan dan kondisi.

### b. Wawancara

Pemangku kepentingan utama di SMK Negeri 5 Kendal diwawancarai, termasuk guru dan staf manajemen sekolah. Tujuan wawancara ini adalah untuk mengumpulkan informasi tentang hal-hal seperti preferensi guru untuk jadwal, ketersediaan ruang kelas dan alokasi waktu. Selain itu, tujuan dari wawancara ini adalah untuk mengidentifikasi masalah yang ada dalam sistem penjadwalan saat ini, mengumpulkan umpan balik dari pemangku kepentingan mengenai fitur yang akan dikembangkan di dalam sistem penjadwalan dan mendapatkan umpan balik tentang prototipe atau ide awal dari sistem penjadwalan yang menggunakan algoritma Greedy untuk memastikan sesuai dengan kebutuhan dan mudah digunakan.

## E. Analisis Data

### 1. Pengaplikasian algoritma greedy

#### a. Pengaplikasian Istilah

Pengaplikasian istilah-istilah yang digunakan dalam algoritma greedy kedalam system penjadwalan mata pelajaran adalah sebagai berikut:

1. Local Optimal Solution

Setiap pilihan terbaik yang diambil untuk mencapai solusi yang diinginkan Memilih slot waktu dan ruangan yang tersedia pertama kali untuk sebuah kelas adalah contoh penjadwalan.

2. Greedy Choice

Pada setiap langkah, pilihan dibuat berdasarkan kriteria yang diyakini akan menghasilkan solusi terbaik. Pada sistem penjadwalan, sistem akan mengatur slot waktu dan ruangan yang tidak akan bentrok dengan kelas lain.

3. Feasibility

Memastikan bahwa pilihan yang diambil tidak melanggar batasan atau aturan yang ada. Pada sistem penjadwalan *Feasibility* akan memastikan dosen atau ruangan tidak bentrok dengan jadwal lain.

4. Optimality

Tujuan akhir dari algoritma greedy adalah mencapai solusi optimal atau mendekati optimal dengan membuat pilihan optimal lokal pada setiap langkah.

5. Completion

Proses menggabungkan semua keputusan lokal untuk membentuk solusi akhir. Pada sistem penjadwalan *Completion* akan membentuk jadwal lengkap untuk semua kelas.

- b. Variabel yang digunakan dalam algoritma *Greedy*

1. Initialization

Inisialisasi struktur data yang diperlukan untuk menyimpan hasil sementara dan akhir. Misalnya, membuat daftar atau matriks untuk menyimpan ketersediaan waktu, ruangan, dan dosen.

2. Iterasi Greedy

Iterasi melalui setiap elemen kelas yang perlu dijadwalkan dan pada setiap iterasi:

- a. Pilih Pilihan Lokal Terbaik: Berdasarkan kriteria greedy, pilih opsi yang paling optimal pada saat itu. Misalnya, pilih

slot waktu dan ruangan yang pertama kali tersedia yang tidak menyebabkan konflik.

- b. Periksa Feasibility: Pastikan pilihan yang diambil tidak melanggar constraints yang ada (tidak ada bentrok jadwal dosen, ruangan tersedia, dll).
- c. Perbarui Struktur Data: Setelah memilih pilihan terbaik, perbarui struktur data untuk mencerminkan perubahan (slot waktu dan ruangan yang telah terpakai).

### 3. Validasi dan Penyesuaian

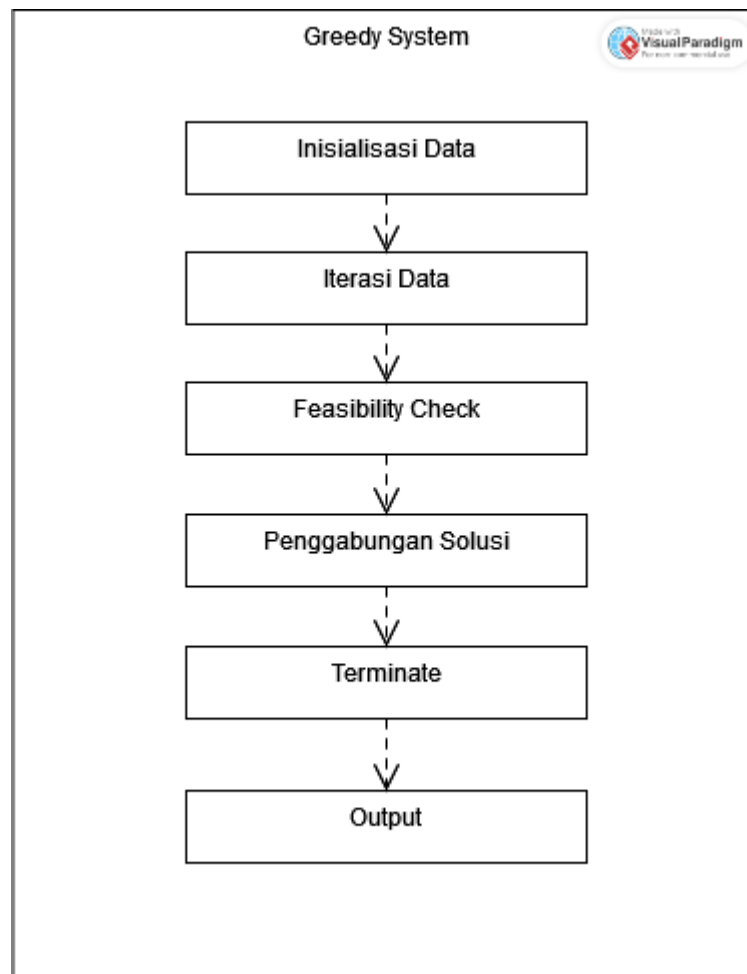
Setelah elemen telah diproses, sistem akan melakukan validasi terhadap solusi yang dihasilkan, memastikan constraints terpenuhi dan mendeteksi konflik dengan melakukan penyesuaian local untuk menyelesaikan masalah tersebut

### 4. Solution Output

Setelah semua langkah iterasi dan penyesuaian selesai, keluarkan solusi akhir yang menunjukkan penjadwalan kelas yang lengkap dan optimal.

### c. Gambaran umum penerapan algoritma *Greedy*

Pada metode algoritma *greedy* memiliki beberapa tahapan. Dalam penelitian ini, metode pembuatan jadwal mata pelajaran memiliki tahapan sebagai berikut :



Gambar 3.2 Greedy

1. Inisialisasi Data

Tahap ini melibatkan pengumpulan dan penyusunan data yang diperlukan, termasuk daftar kelas, mata pelajaran, slot waktu yang tersedia, ruangan, dan preferensi pengajar. Data ini akan menjadi dasar untuk proses penjadwalan.

2. Iterasi Melalui Setiap Kelas

Algoritma akan mengiterasi melalui setiap kelas yang perlu dijadwalkan. Setiap kelas akan diproses satu per satu untuk menentukan jadwal yang optimal.

### 3. Iterasi Melalui Setiap Hari

Untuk setiap kelas, algoritma akan mengiterasi melalui setiap hari dalam minggu akademik. Ini membantu memastikan bahwa jadwal yang dihasilkan mencakup seluruh minggu.

### 4. Iterasi Melalui Setiap Slot Waktu

Di setiap hari, algoritma akan mengiterasi melalui setiap slot waktu yang tersedia. Ini termasuk jam pelajaran pagi, siang, dan sore.

### 5. Iterasi Melalui Setiap Ruangan

Untuk setiap slot waktu, algoritma akan mengiterasi melalui setiap ruangan yang tersedia. Ini membantu memastikan bahwa semua ruangan digunakan secara efisien dan sesuai dengan kebutuhan kelas.

### 6. Pemeriksaan Ketersediaan Slot (Feasibility Check)

Pada tahap ini, algoritma akan memeriksa ketersediaan slot waktu dan ruangan untuk setiap kelas. Pemeriksaan ini memastikan bahwa slot waktu dan ruangan yang dipilih tidak bertabrakan dengan jadwal lainnya dan memenuhi semua persyaratan.

### 7. Penggabungan Solusi

Setelah menemukan slot waktu yang sesuai, algoritma akan menggabungkan solusi ini ke dalam jadwal keseluruhan. Ini memastikan bahwa jadwal yang dihasilkan konsisten dan tidak bertabrakan.

### 8. Terminasi

Algoritma akan berhenti ketika semua kelas telah dijadwalkan atau tidak ada lagi solusi yang memenuhi persyaratan. Ini menandai akhir dari proses penjadwalan.

### 9. Output Solusi

Tahap terakhir adalah menghasilkan output berupa jadwal mata pelajaran yang telah dioptimalkan. Output ini kemudian dapat



digunakan oleh institusi pendidikan untuk pelaksanaan kegiatan akademik.

## **F. Metode Pengembangan Sistem**

Metode Rapid Application Development (RAD) adalah pendekatan yang berfokus pada kecepatan dan iterasi dalam pengembangan sistem. RAD sangat cocok untuk proyek yang memerlukan pengembangan cepat dengan perubahan yang sering berdasarkan umpan balik pengguna. Tahapan – tahapan penelitian terdiri dari Requirement Planning, User Design dan Construction.

### **1. Requirement Planning**

Analisis persyaratan, langkah pertama dalam metode Rapid Application Development (RAD), berfokus pada mengumpulkan dan memahami kebutuhan pengguna dan pemangku kepentingan. Fase ini sangat penting karena di sini semua kebutuhan dan spesifikasi sistem ditentukan. Ini akan memastikan bahwa sistem yang akan dibangun akan memenuhi harapan dan memenuhi kebutuhan. Melakukan wawancara dan survei dengan guru, staf administrasi, dan pemangku kepentingan lainnya untuk mengidentifikasi kebutuhan utama dan persyaratan spesifik, seperti preferensi waktu mengajar guru, alokasi ruangan, dan batasan waktu tertentu dalam penjadwalan.

### **2. User Desain**

Pada tahap ini, peneliti membuat prototipe awal dari sistem berdasarkan kebutuhan yang telah dikumpulkan. Prototipe ini berfungsi sebagai versi awal dari sistem yang dapat diuji dan diulas oleh pengguna, sehingga umpan balik bisa segera diperoleh dan digunakan untuk perbaikan. Menentukan arsitektur sistem, termasuk database untuk menyimpan jadwal, ruangan, dan data guru, antarmuka pengguna (UI) untuk pengelolaan jadwal, serta komponen backend untuk logika bisnis dan algoritma penjadwalan.

### 3. Construction

Tahap Konstruksi adalah fase pengembangan utama dalam metode RAD, di mana prototipe yang telah disempurnakan berdasarkan umpan balik pengguna dikembangkan menjadi sistem yang lengkap. Pengembangan dilakukan secara iteratif dan inkremental, dengan setiap iterasi menghasilkan versi yang lebih baik dari sistem. Mengembangkan sistem dalam iterasi pendek, setiap iterasi menambahkan fitur baru atau memperbaiki fitur yang ada berdasarkan umpan balik pengguna. Kemudian melakukan pengujian secara berulang pada setiap iterasi untuk memastikan bahwa setiap komponen bekerja dengan benar dan sesuai dengan kebutuhan pengguna. Pengujian meliputi black box testing, white box testing, dan User Acceptance Testing (UAT).

#### a. Black box

Pengujian ini fokus pada input dan output serta fungsionalitas sistem. Pengujian dilakukan untuk memastikan bahwa semua fitur dan fungsi yang diharapkan bekerja sesuai dengan spesifikasi. Setiap iterasi akan diuji untuk memastikan bahwa fitur baru berfungsi dengan baik dan tidak ada regresi pada fitur yang sudah ada.

#### b. White box

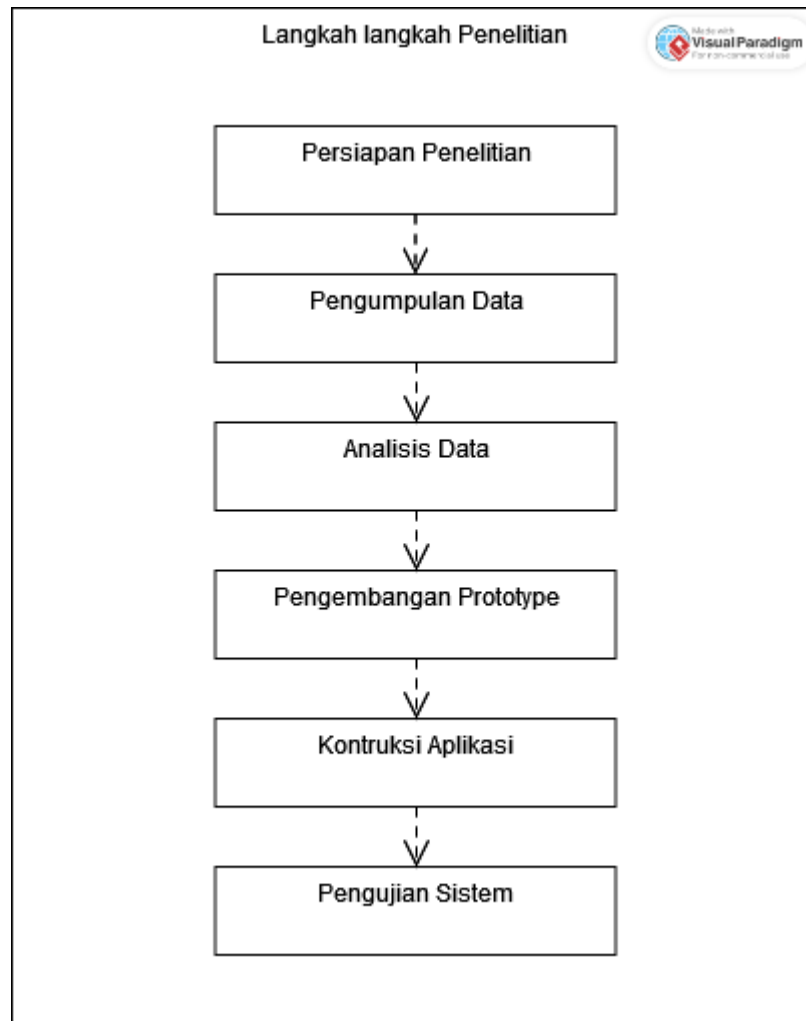
White box testing melibatkan pengujian yang didasarkan pada pengetahuan tentang struktur internal dan kode dari sistem untuk mengetahui apakah terdapat kesalahan pada kode program yang telah dikembangkan.

#### c. User Acceptance testing (UAT)

UAT melibatkan pengguna akhir untuk memastikan bahwa sistem memenuhi kebutuhan yang diharapkan. Umpan balik dari UAT digunakan untuk memperbaiki dan menyempurnakan sistem dalam iterasi berikutnya.

## G. Langkah Penelitian

Dalam sebuah penelitian, diperlukan langkah – langkah yang jelas untuk menggambarkan apa yang akan dilakukan peneliti. Langkah – Langkah yang akan dilakukan dalam penelitian ini adalah sebagai berikut



Gambar 3.3 Langkah Penelitian

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **A. Penyajian Hasil**

Hasil dan pembahasan penelitian penulis akan dibahas pada bagian ini. Analisis pengembangan dan perancangan sistem penjadwalan yang ada di SMK Negeri 5 Kendal berbasis web, adalah bagian dari pekerjaannya. Penelitian ini menggunakan Algoritma Greedy dan pengembangan sistem menggunakan metode RAD. Penulis melakukan hal-hal berikut.

##### 1. Analisis Kebutuhan Sistem

Analisis kebutuhan dilakukan untuk mendapatkan pemahaman lebih lanjut tentang komponen yang diperlukan untuk proses perancangan dan pembangunan sistem penjadwalan berbasis web. Hardware dan software yang digunakan dalam pengembangan sistem adalah sebagai berikut:

##### a. Kebutuhan Software

1. Visual Studio Code
2. XAMPP
3. PhpMyAdmin
4. PHP
5. Laravel
6. Web Browser
7. Figma
8. Visual Paradigma

##### b. Kebutuhan Hardware

Laptop dengan spesifikasi :

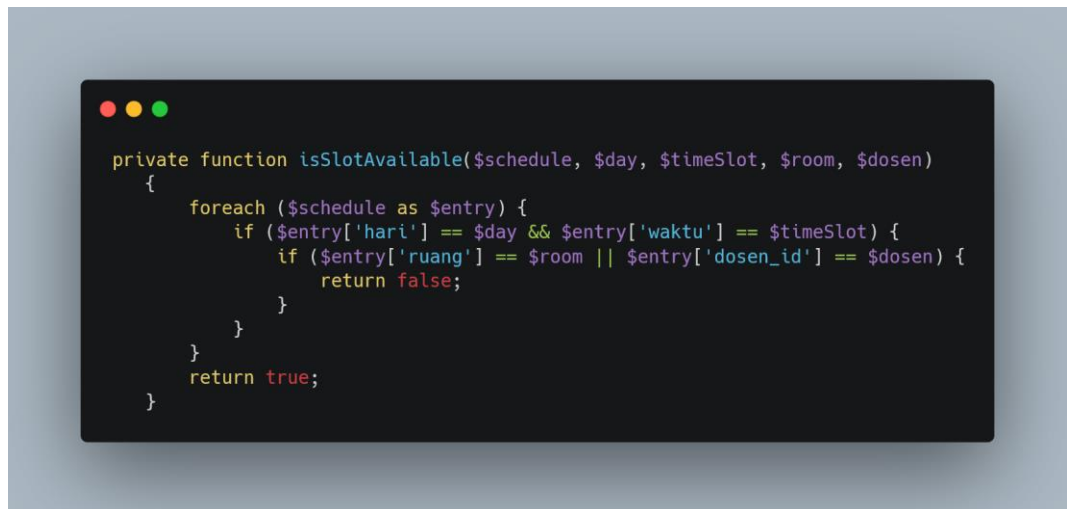
1. Processor Intel Core i5-8250U
2. Grafis Card NVIDIA GTX 950
3. RAM 4 DDR 4

## 2. Analisis Kebutuhan Fungsional

- a. Sistem dapat memeriksa dan mengarahkan pengguna berdasarkan peran mereka saat melakukan login.
- b. Sistem mendukung berbagai jenis pengguna seperti admin, dosen, dan mahasiswa.
- c. Sistem harus memastikan bahwa pengguna hanya dapat mengakses data dan fitur yang sesuai dengan peran mereka.
- d. Sistem dapat mengatur hak akses untuk setiap jenis pengguna yang masuk melalui data table yang tersimpan pada database.
- e. Admin dapat mengelola data pengguna termasuk penambahan, pengeditan, dan penghapusan pengguna.
- f. Admin dapat mengelola data penjadwalan, termasuk menambahkan, mengedit, dan menghapus jadwal.
- g. Admin dapat melihat semua jadwal yang ada dalam sistem.
- h. Sistem dapat menerapkan algoritma greedy untuk mengoptimalkan penjadwalan untuk menghindari konflik waktu, ruangan, dan dosen.

### ***B. Penerapan Metode Greedy***

Prinsip dasar metode greedy adalah membuat keputusan terbaik setiap saat tanpa mempertimbangkan konsekuensi jangka panjang. Penerapan algoritma greedy untuk menghindari konflik penjadwalan, yaitu dengan cara bertahap memilih slot waktu yang tersedia untuk setiap kelas. Berikut adalah ringkasan mengenai constraint yang diterapkan melalui pemeriksaan di dalam fungsi `isSlotAvailable($schedule, $day, $timeSlot, $room, $dosen)`. Jika ditemukan kelas lain yang sudah dijadwalkan pada hari yang sama, slot waktu tersebut tidak akan dipilih. Penerapan code constrain dijelaskan pada gambar 4.1 :



Gambar 4. 1 Constraint

## 1. Constraint

- a) Tidak mengizinkan tabrakan hari: Memastikan tidak ada dua atau lebih mata pelajaran dari satu kelas yang dijadwalkan pada hari yang sama. Jika ditemukan kelas lain yang sudah dijadwalkan pada hari yang sama, slot waktu tersebut tidak akan dipilih
- b) Tidak mengizinkan tabrakan jam: Memastikan tidak ada dua atau lebih mata pelajaran dari satu kelas yang dijadwalkan dengan cara kode memeriksa apakah slot waktu yang sama sudah diambil oleh kelas lain. Jika ya, maka slot tersebut akan dianggap tidak tersedia.
- c) Tidak mengizinkan tabrakan ruangan: Memastikan tidak ada dua atau lebih mata pelajaran dari satu kelas yang dijadwalkan. Jika ruangan sudah diambil oleh kelas lain pada waktu yang sama, maka sistem menggeser waktu dari ruangan.
- d) Tidak mengizinkan tabrakan guru: Memastikan tidak ada satu guru yang dijadwalkan mengajar lebih dari satu mata kuliah pada waktu yang sama.

## 2. Algoritma Greedy

Sistem penjadwalan menggunakan metode greedy untuk mengatur jadwal kelas berdasarkan waktu yang dimiliki oleh ruangan dan guru

yang tersedia. Metode greedy bekerja dengan cara memilih solusi terbaik pada setiap langkah tanpa mempertimbangkan kemungkinan konflik di masa depan. Dalam penerapan ini, sistem menghindari tabrakan jadwal berdasarkan beberapa constraint yang telah ditentukan, seperti hari, jam, ruangan, dan guru. Penerapan algoritma greedy dijelaskan pada gambar 4.2 :



```

public function generateSchedule($classes, $timeSlots, $rooms, $dosen)
{
    $schedule = [];

    foreach ($classes as $class) {
        $scheduled = false;
        $day = $class->hari;

        $availableRooms = array_filter($rooms, function ($roomId) use ($class) {
            return $roomId == $class->ruang_id;
        }, ARRAY_FILTER_USE_KEY);

        foreach ($timeSlots as $timeSlot) {
            foreach ($availableRooms as $roomId => $roomName) {
                if ($this->isSlotAvailable($schedule, $day, $timeSlot, $roomName, $class->dosen_id)) {
                    $schedule[] = [
                        'kode_kelas' => $class->kode_kelas,
                        'makul_id' => $class->makul_id,
                        'progdi_id' => $class->progdi_id,
                        'ruang' => $roomName,
                        'ruang_id' => $roomId,
                        'dosen_id' => $class->dosen_id,
                        'dosen_name' => $dosen[$class->dosen_id],
                        'waktu' => $timeSlot,
                        'hari' => $day,
                        'semester' => $class->semester,
                    ];

                    $scheduled = true;
                    break 2;
                }
            }
        }
    }

    return $schedule;
}

```

Gambar 4. 2 Pengoptimalisasi jadwal

a) Proses Penjadwalan:

- 1) Iterasi Kelas: Proses penjadwalan dimulai dengan mengiterasi setiap kelas yang perlu dijadwalkan. Fungsi generateSchedule(\$classes, \$timeSlots, \$rooms, \$dosen) berperan penting dalam hal ini. Untuk setiap kelas, sistem akan

mencoba menemukan slot waktu yang sesuai dan memastikan tidak ada tabrakan dengan jadwal lain.

- 2) Pemeriksaan Ketersediaan Slot: Sistem memeriksa ketersediaan slot waktu dengan memanggil fungsi `isSlotAvailable()`. Fungsi ini mengecek semua constraint yang ada (hari, jam, ruangan, dan guru) untuk memastikan bahwa kelas bisa dijadwalkan pada slot tersebut.
- 3) Penjadwalan Kelas: Jika sistem menemukan slot waktu dan ruangan yang sesuai tanpa adanya konflik, kelas tersebut akan dijadwalkan pada slot tersebut dan disimpan dalam array `$schedule` yang kemudian dikembalikan oleh fungsi `generateSchedule()`.
- 4) Penanganan Konflik: Jika ditemukan konflik, seperti tabrakan ruangan atau guru, kelas tersebut akan ditunda atau dialokasikan ke slot waktu atau ruangan yang berbeda.

Tabel 4. 1 Cara Kerja Greedy

Kode Kelas	Hari	Jam	Ruang	Guru
KLS001	Senin	07.30-09:00	R01	Guru A
KLS002	Senin	07.30-09:00	R02	Guru A
KLS003	Senin	07.30-09:00	R03	Guru B

#### b) Penjelasan Cara Kerja Sistem

- 1) KLS001: Sistem akan pertama kali menjadwalkan kelas KLS001 pada hari Senin pukul 07.30-09:00 di ruangan R01 dengan Guru A. Karena ini adalah entri pertama, tidak ada konflik yang terdeteksi. Data ini akan disimpan dalam array `$schedule`.
- 2) KLS002: sistem mencoba menjadwalkan KLS002 pada waktu yang sama di ruangan R02 dengan Guru A, `isSlotAvailable` akan mendeteksi bahwa Guru A sudah terjadwal mengajar di KLS001



pada waktu yang sama. Oleh karena itu, sistem menandai adanya tabrakan jadwal guru dan mencari alternatif lain.

- 3) KLS003: Untuk KLS003, sistem mencoba menjadwalkan kelas ini pada hari Senin pukul 07.30-09:00 di ruangan R01 dengan Guru B. Sistem mendeteksi bahwa ruangan R01 sudah ditempati oleh KLS001 pada waktu yang sama, sehingga terjadi tabrakan ruangan.

c) Penyelesaian Konflik

Sistem kemudian akan mencoba untuk menyelesaikan konflik dengan memeriksa slot waktu atau ruangan alternatif yang tersedia. Dalam kasus di atas, sistem akan mencoba:

- 1) KLS002: Mengatur ulang jadwal dengan menggeser waktu KLS002 yang berbeda di mana Guru A tidak memiliki jadwal lain.
- 2) KLS003: Memindahkan KLS003 dengan menggeser waktu ke slot yang berbeda.

Dengan demikian, sistem penjadwalan ini berusaha meminimalkan tabrakan jadwal dengan mempertimbangkan constraint yang ada dan memilih solusi yang optimal pada setiap langkah.

### C. Perancangan Sistem

Salah satu syarat untuk memulai pengembangan sistem adalah perancangan desain sistem. Penulis menggunakan UML (Unified Modeling Language) untuk membuat perancangan serta tahap desain awal sistem.

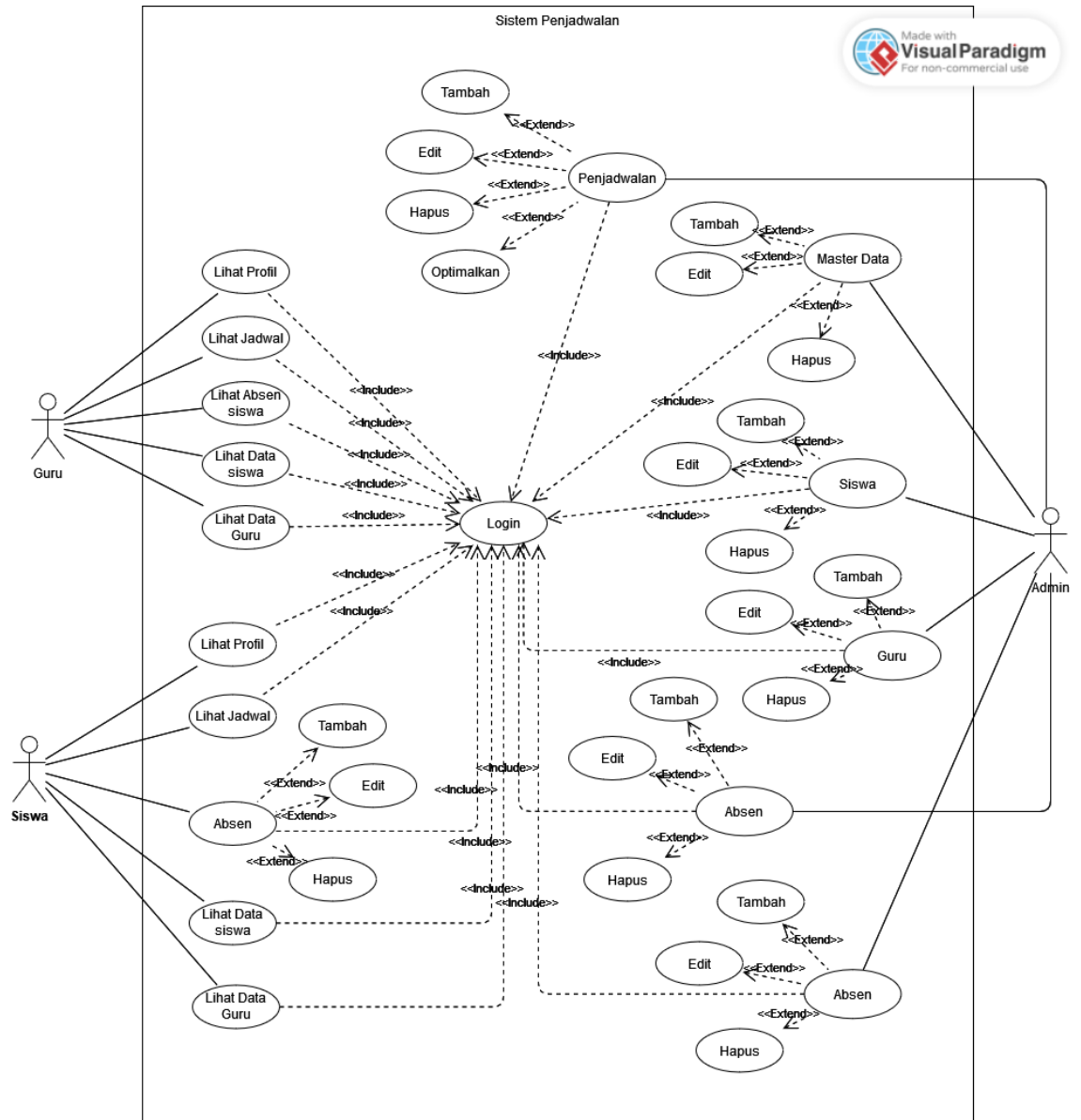
1. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) terdiri dari *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*.

a. *Use Case Diagram*

Diagram use case menunjukkan perspektif user tentang fungsi sistem dan siapa yang berhak menggunakannya. Use Case

Diagram yang diusulkan pada Website sistem penjadwalan di SMK Negeri 5 Kendal dijelaskan pada gambar 4.3 :



Gambar 4.3 Use Case Diagram

1. Admin dapat mengakses login, dashboard, mengelola data di dalam master data, mengelola data guru, siswa, penjadwalan, absensi, user siswa dan user guru.
2. Guru dapat mengakses login, dashboard, profil, jadwal, melihat data guru, siswa dan absensi siswa.

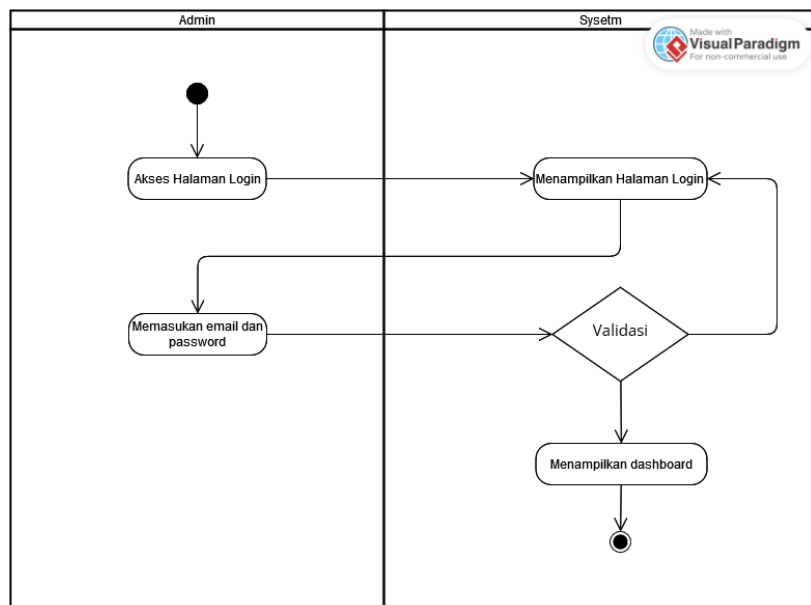
3. Siswa dapat mengakses login, dashboard, profil, jadwal, mengelola data absensi, melihat data guru dan siswa.

b. Activity Diagram

Activity Diagram berikut menunjukkan data secara rinci tanpa mempertimbangkan lingkungan fisik di mana data mengalir, Dimana user/pengguna masuk kedalam sistem.

1. Activity Diagram Login

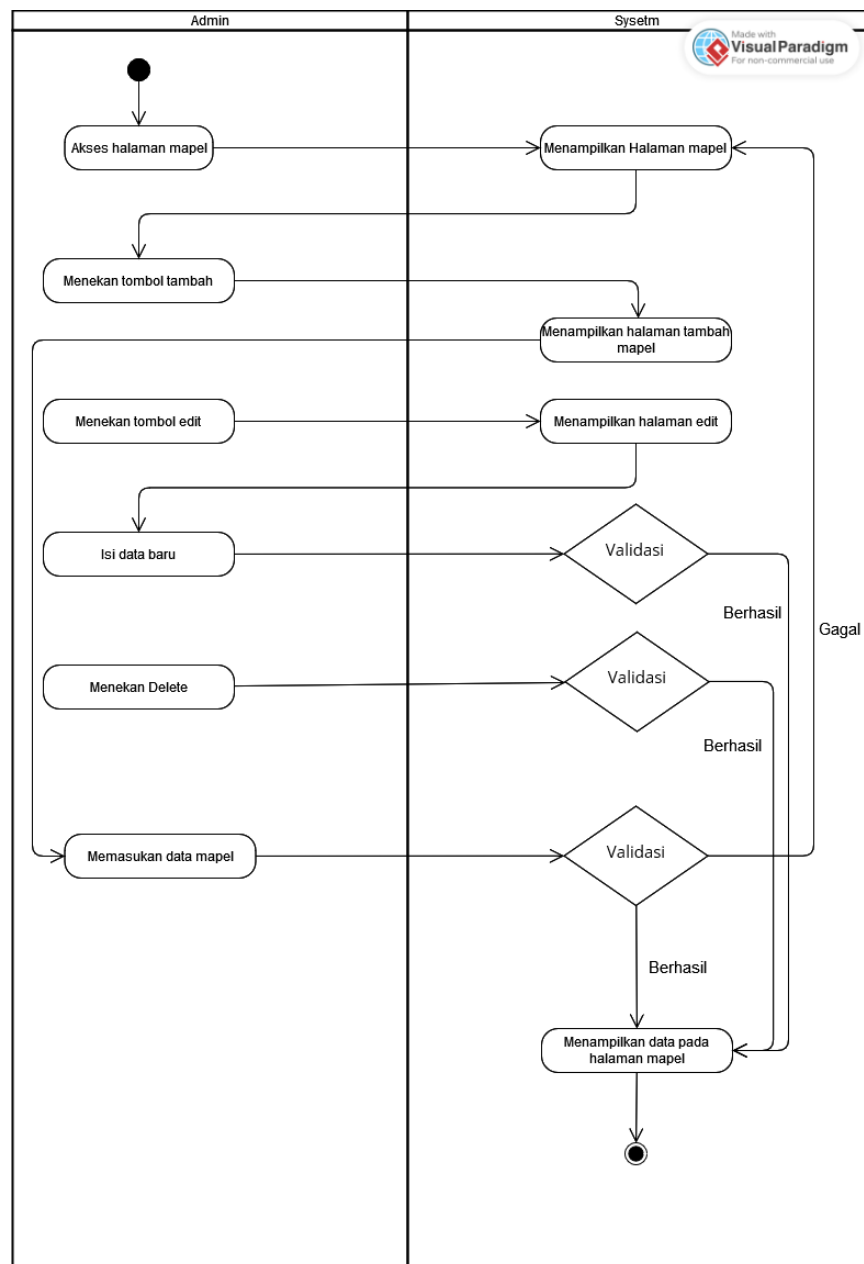
Diagram aktivitas login menunjukkan proses login ke halaman aplikasi sistem penjadwalan. Untuk mengakses aplikasi, pengguna harus memasukkan email dan password. Kemudian sistem akan memeriksa identitas untuk memastikan bahwa email dan password tersebut benar. seperti yang ditunjukkan pada Gambar 4.4.



Gambar 4.4 Activity Diagram

## 2. Activity diagram mengelola data mapel

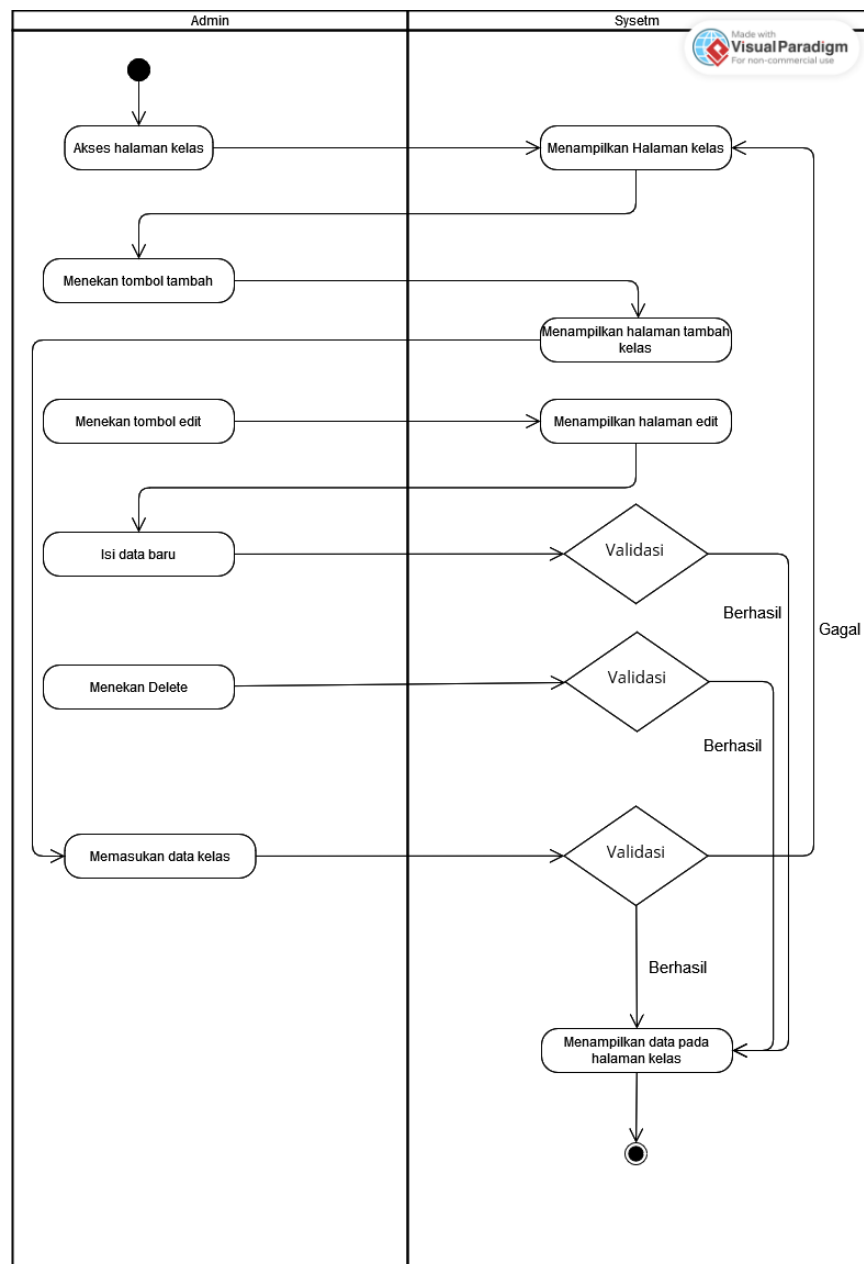
Activity diagram untuk mengolah data mata pelajaran membutuhkan login, kemudian memilih master data dan pilih fitur mapel untuk mengelola data. Admin dapat menghapus, menambahkan, atau mengubah data mapel, seperti yang ditunjukkan pada Gambar 4.5.



Gambar 4.5 Mengelola data mapel

### 3. Activity diagram mengelola data kelas

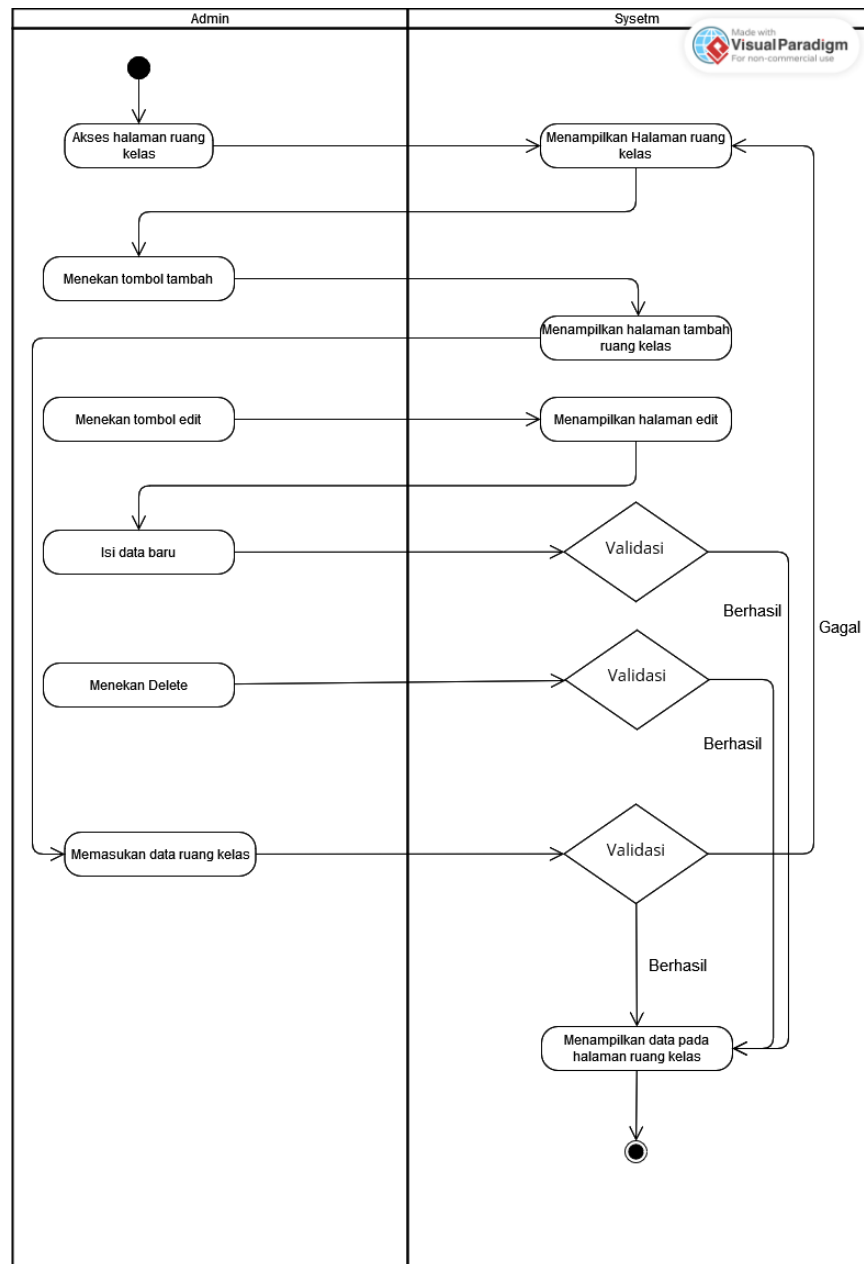
Activity diagram untuk mengolah data kelas membutuhkan login, kemudian memilih master data dan pilih fitur kelas untuk mengelola data. Admin dapat menghapus, menambahkan, atau mengubah data kelas, seperti yang ditunjukkan pada Gambar 4.6.



Gambar 4.6 Mengelola data mapel

#### 4. Activity diagram mengelola data ruang kelas

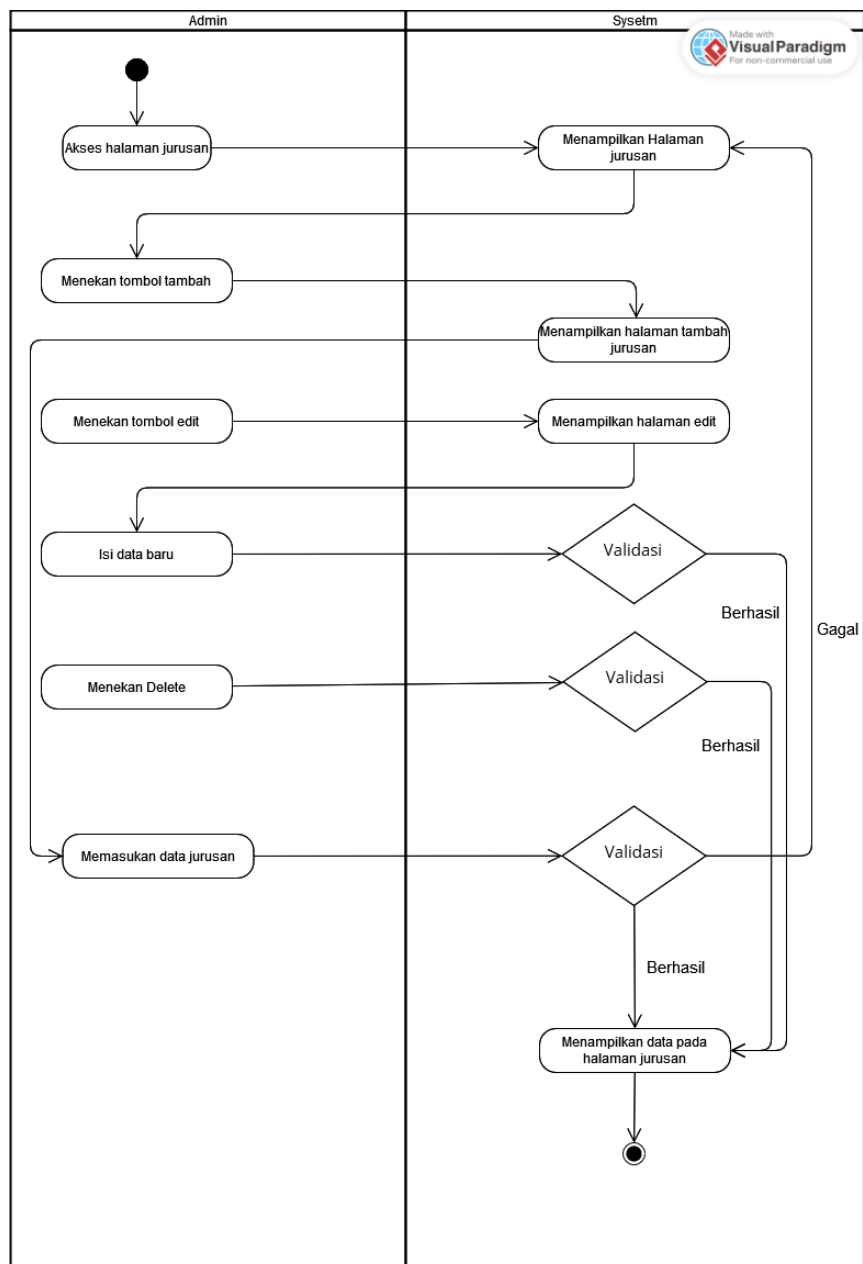
Activity diagram untuk mengolah data ruang kelas membutuhkan login, kemudian memilih master data dan pilih fitur ruang kelas untuk mengelola data. Admin dapat menghapus, menambah, atau mengubah data ruang, seperti yang ditunjukkan pada Gambar 4.7.



Gambar 4.7 Mengelola data ruang kelas

### 5. Activity diagram mengelola data jurusan

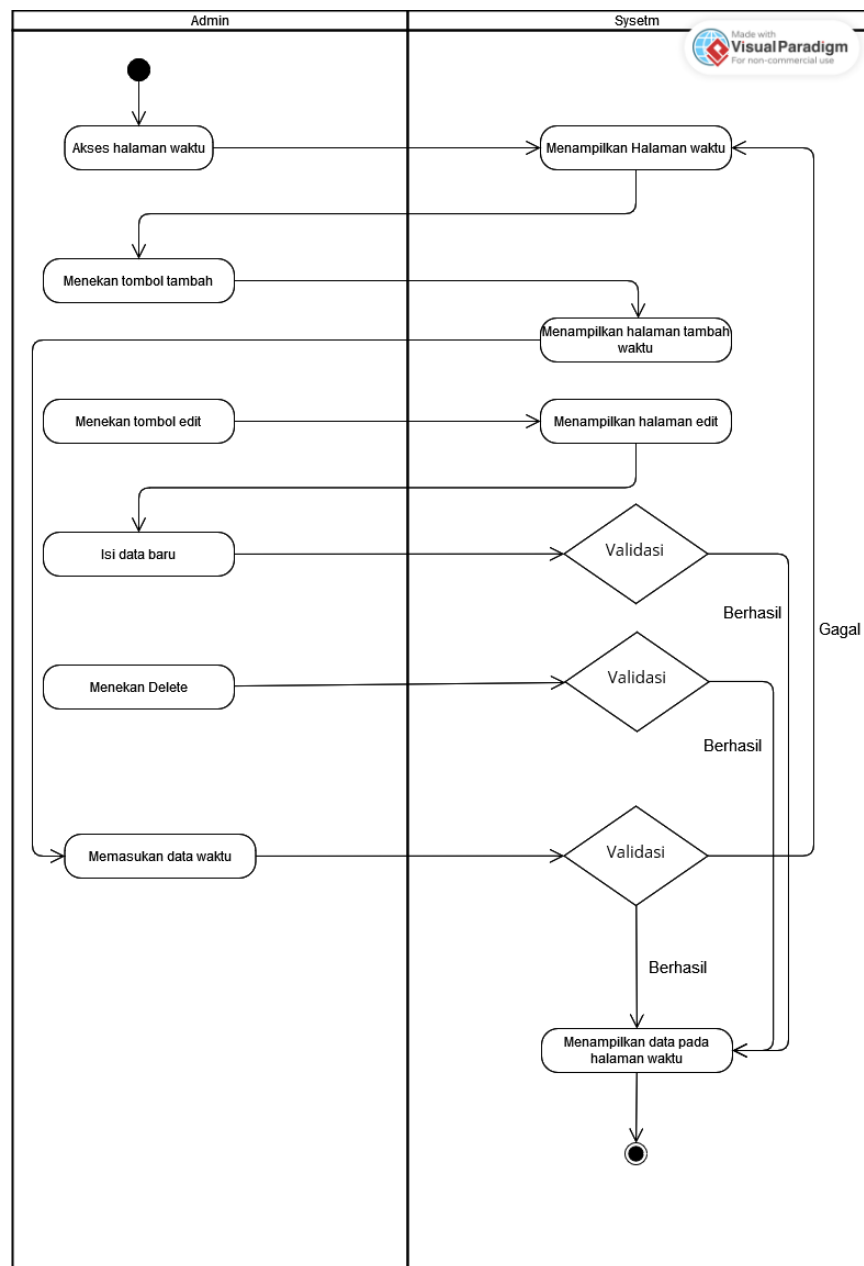
Activity diagram untuk mengolah data jurusan membutuhkan login, kemudian memilih master data dan pilih fitur jurusan untuk mengelola data. Admin dapat menghapus, menambahkan, atau mengubah data jurusan, seperti yang ditunjukkan pada Gambar 4.8.



Gambar 4.8 Mengelola data jurusan

## 6. Activity diagram mengelola data waktu

Activity diagram untuk mengolah data waktu membutuhkan login, kemudian memilih master data dan pilih fitur waktu untuk mengelola data. Admin dapat menghapus, menambahkan, atau mengubah data waktu, seperti yang ditunjukkan pada Gambar 4.9.

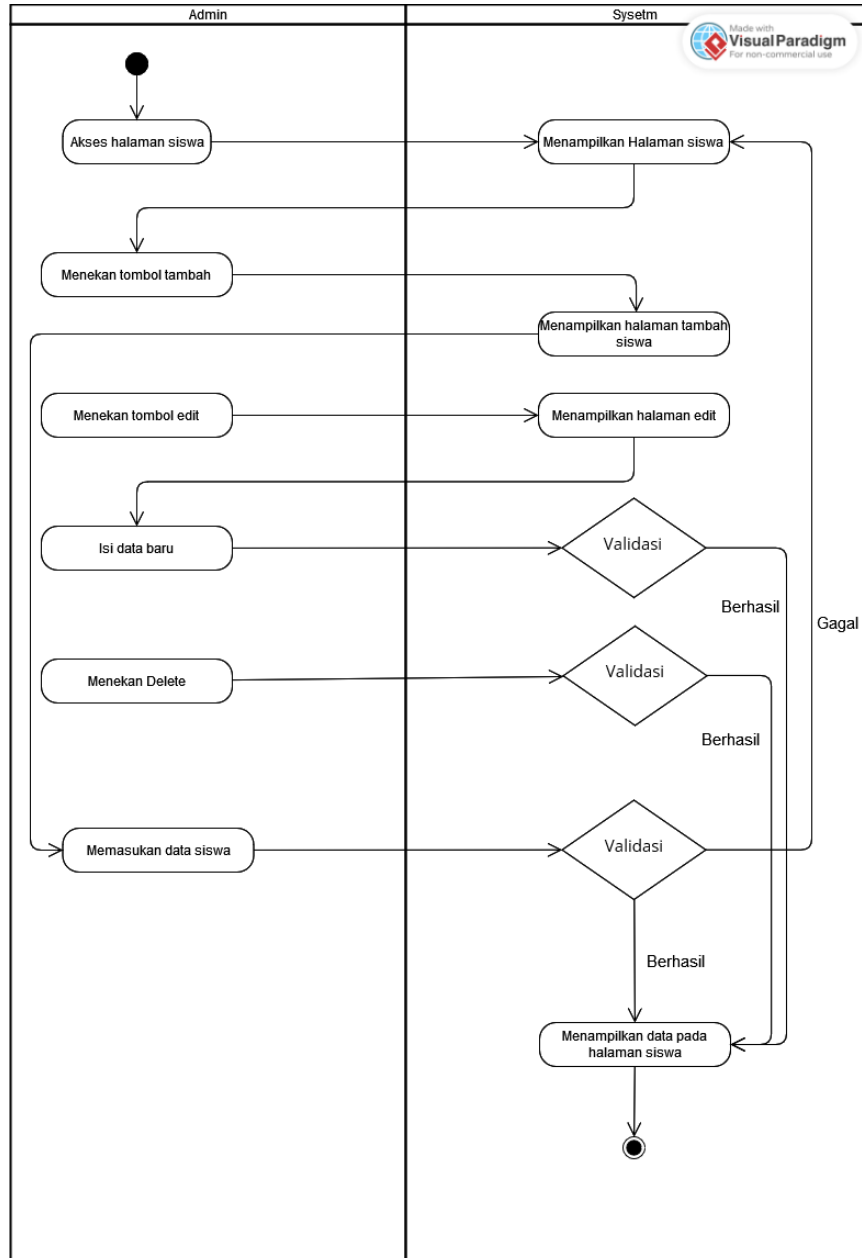


Gambar 4.9 Megelola data waktu



7. Activity diagram mengelola data siswa

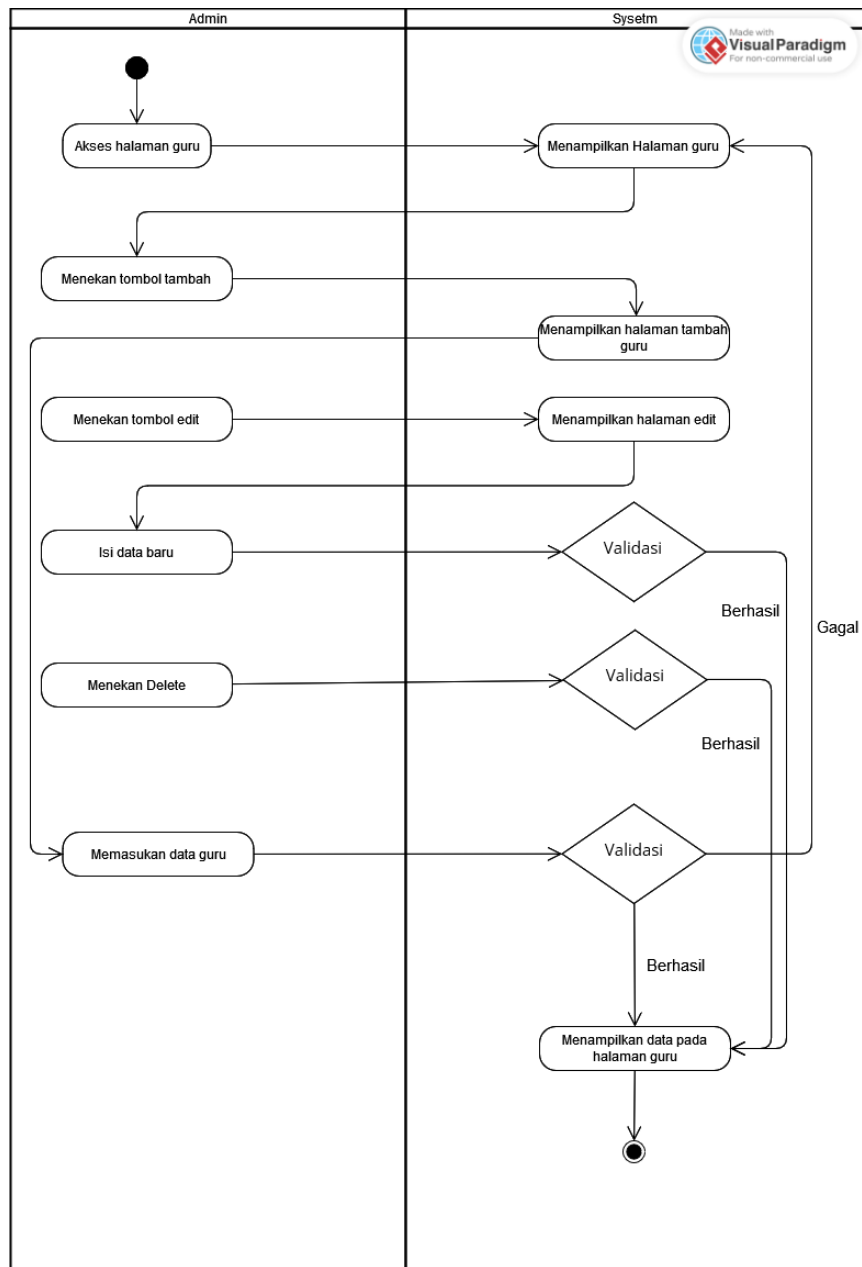
Activity diagram untuk mengolah data siswa membutuhkan login, kemudian pilih fitur siswa untuk mengelola data. Admin dapat menghapus, menambahkan, atau mengubah data siswa, seperti yang ditunjukkan pada Gambar 4.10.



Gambar 4.10 Mengelola data siswa

8. Activity diagram mengelola data guru

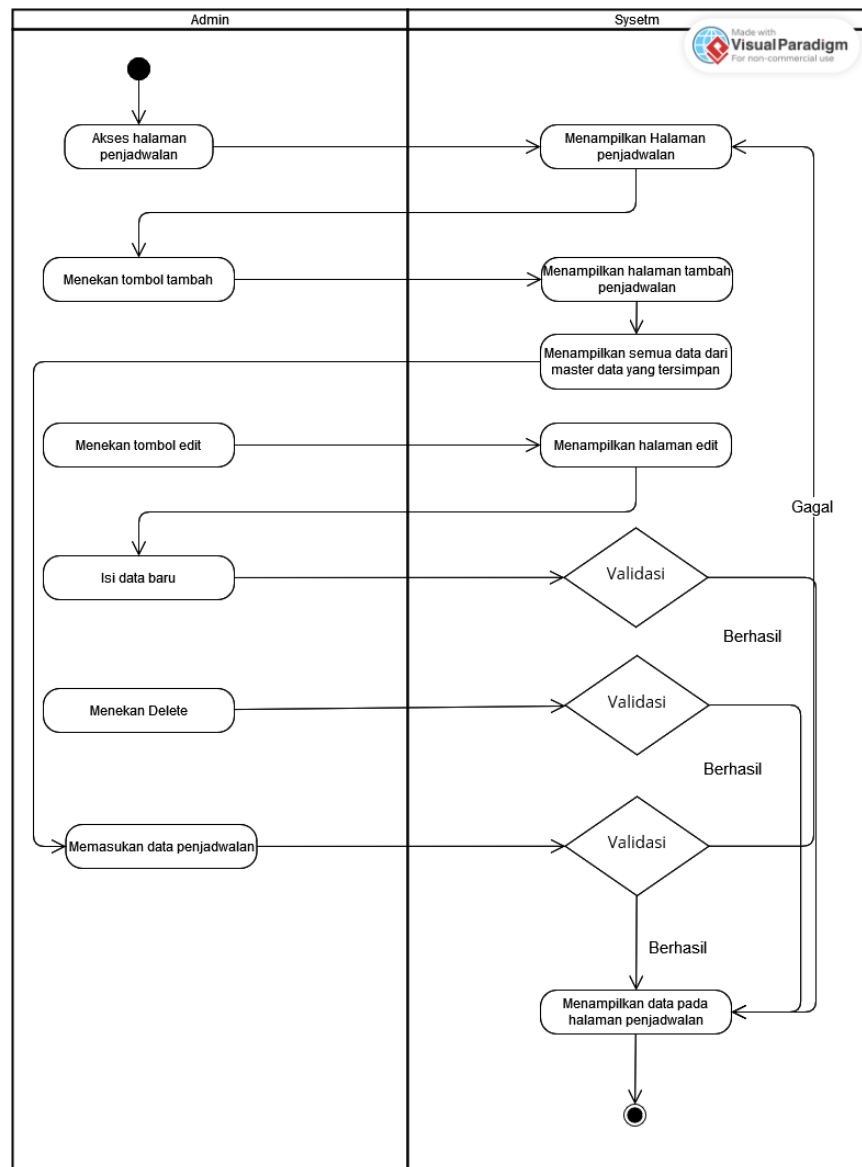
Activity diagram untuk mengolah data guru membutuhkan login, kemudian memilih master data dan pilih fitur guru untuk mengelola data. Admin dapat menghapus, menambahkan, atau mengubah data guru, seperti yang ditunjukkan pada Gambar 4.11.



Gambar 4.11 Mengelola data guru

### 9. Activity diagram mengelola data penjadwal

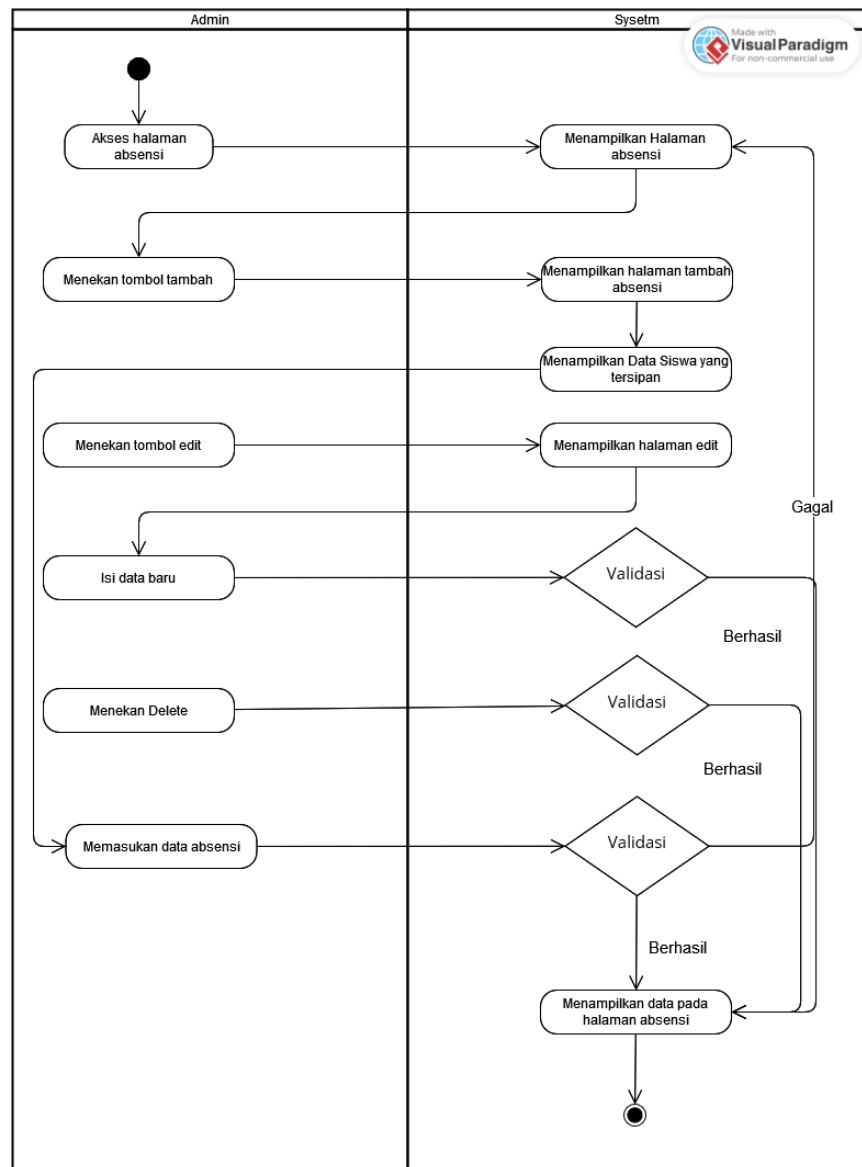
Untuk mengolah data penjadwalan, Admin harus login kemudian memilih fitur jadwal agar dapat memulai mengelola data. Untuk mempermudah pengaturan jadwal, admin akan melihat bahwa semua data yang tersimpan di master data akan di tampilkan kembali saat penginputan data. Seperti yang ditunjukkan pada Gambar 4.12.



Gambar 4.12 Mengelola data jadwal

### 10. Activity diagram mengelola data absen

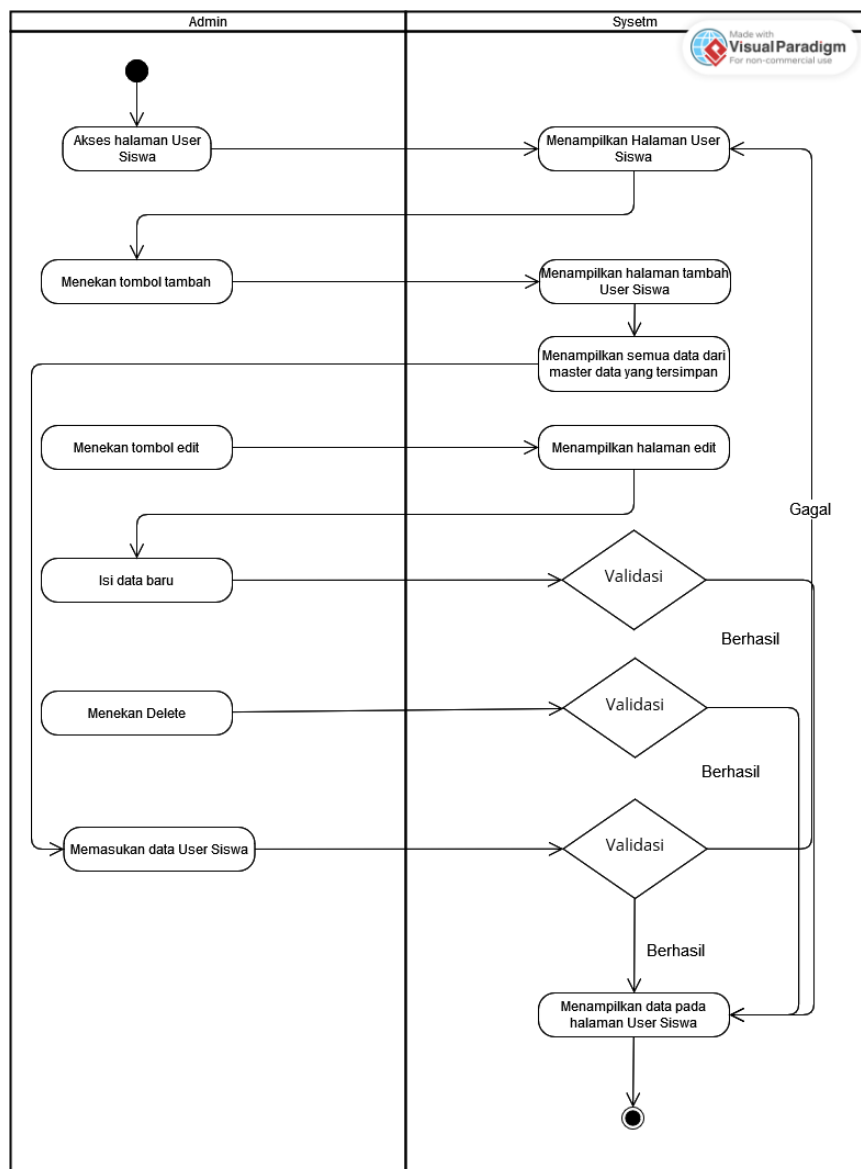
Untuk mengolah data absensi siswa, Admin harus login kemudian memilih fitur absensi siswa agar bisa memulai mengelola data. Untuk mempermudah pengaturan jadwal, administrator akan melihat data siswa yang tersimpan di panggil kembali pada saat penginputan data. Seperti yang ditunjukkan pada Gambar 4.13.



Gambar 4.13 Mengelola data absensi siswa

### 11. Activity diagram mengelola data user siswa

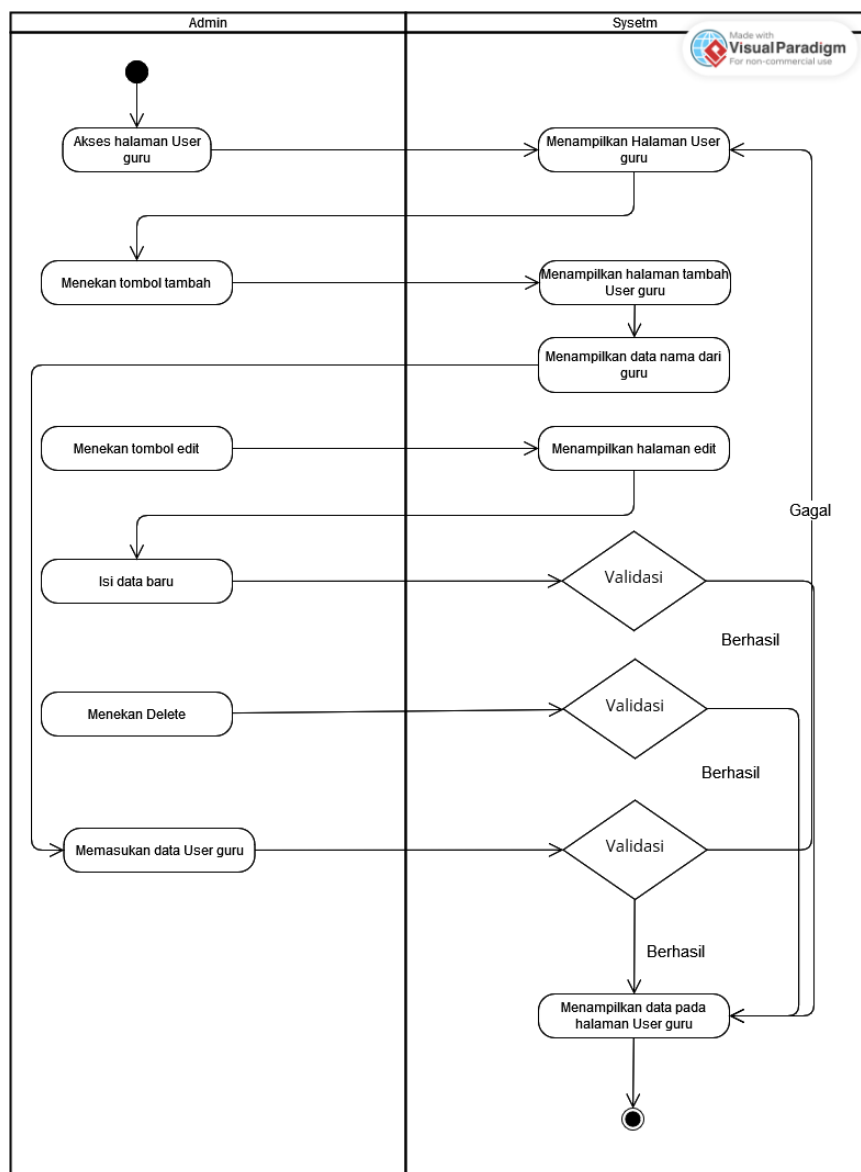
Untuk mengolah data user siswa, Admin harus login kemudian memilih fitur user siswa agar bisa memulai mengelola data. Untuk mempermudah pengaturan user siswa, administrator akan melihat data siswa yang tersimpan di tampilan kembali pada saat penginputan data. Seperti yang ditunjukkan pada Gambar 4.14.



Gambar 4.14 Mengelola user data siswa

## 12. Activity diagram mengelola data user guru

Untuk mengolah data user guru, Admin harus login kemudian memilih fitur user siswa agar bisa memulai mengelola data. Untuk mempermudah pengaturan user guru, administrator akan melihat data guru yang tersimpan di panggil kembali pada saat penginputan data. Seperti yang ditunjukkan pada Gambar 4.15.

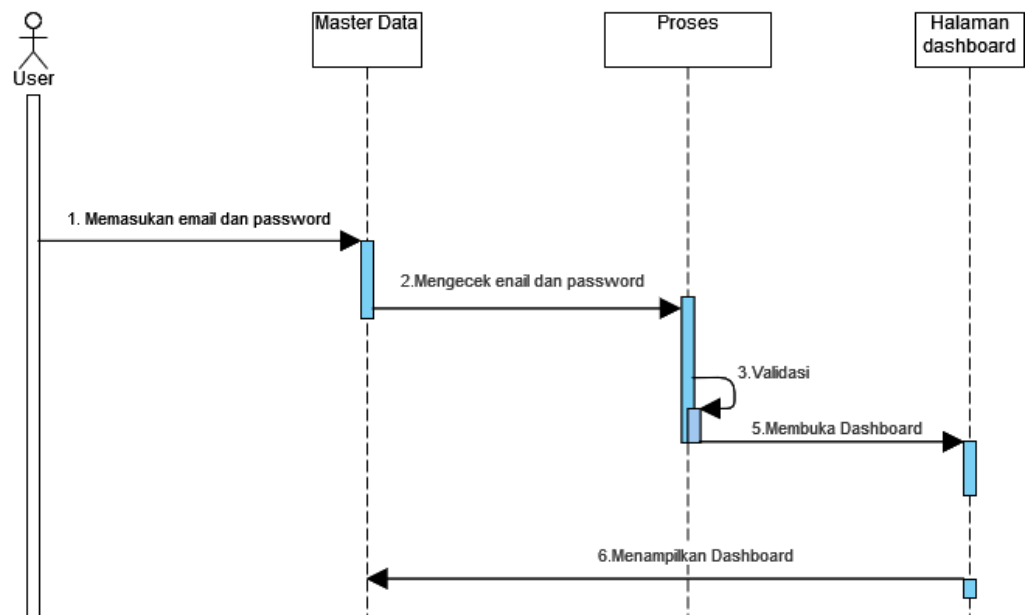


Gambar 4.15 User data guru

### c. Sequence Diagram

#### 1. Sequence diagram masuk dashboard user

Diagram alur kerja user atau superuser, dimulai dengan user masuk ke laman login kemudian user dapat memasukkan email dan password. Setelah itu sistem akan melakukan validasi melalui email dan password yang dimasukkan oleh user untuk membedakan user login sebagai apa. Jika validasi gagal, user akan diarahkan kembali ke laman login, seperti yang ditunjukkan pada gambar 4.16.

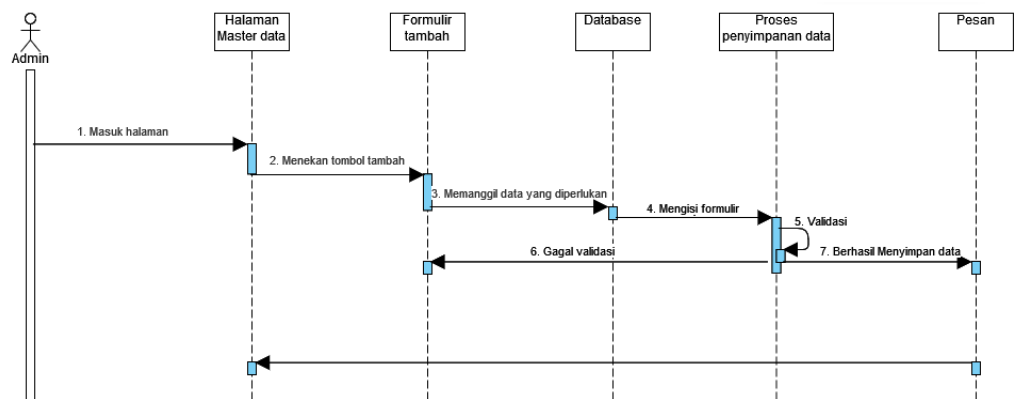


Gambar 4.16 Sequence diagram dashboard

#### 2. Sequence diagram menambahkan master data

Diagram alur kerja admin menambahkan master data, dimulai dengan admin masuk ke laman masterdata kemudian admin dapat menambahkan data dengan mengisi informasi baru di dalam form penambahan master data. Setelah proses penyimpanan berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke

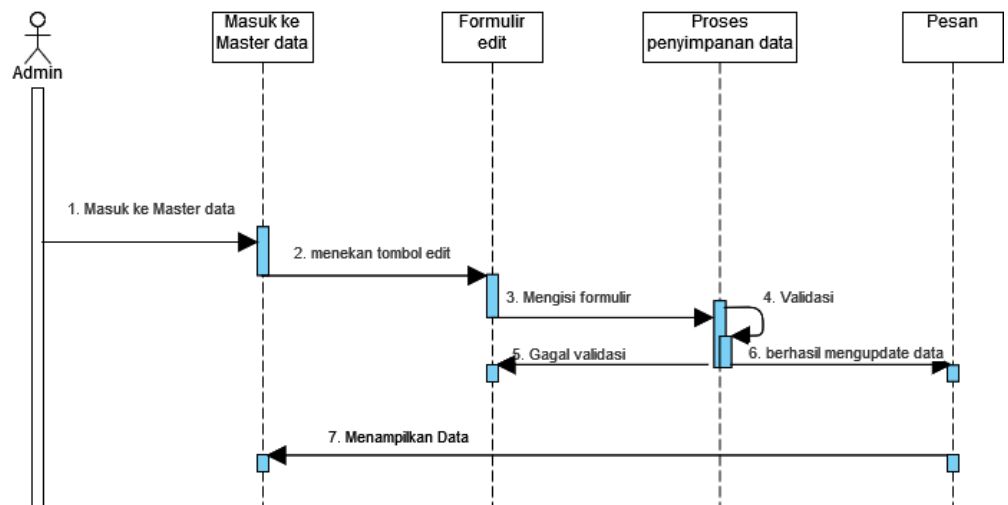
laman pengisian informasi master data, seperti yang ditunjukkan pada gambar 4.17.



Gambar 4.17 Sequence diagtam tambah master data

### 3. Sequence diagram edit master data

Diagram alur kerja admin mengedit master data, dimulai dengan admin masuk ke laman master data kemudian admin dapat mengedit data dengan menekan tombol edit, data yang sebelumnya tersimpan akan ditampilkan dan admin dapat mengisi informasi baru. Setelah proses edit berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman pengeditan informasi master data, seperti yang ditunjukkan pada gambar 4.18.

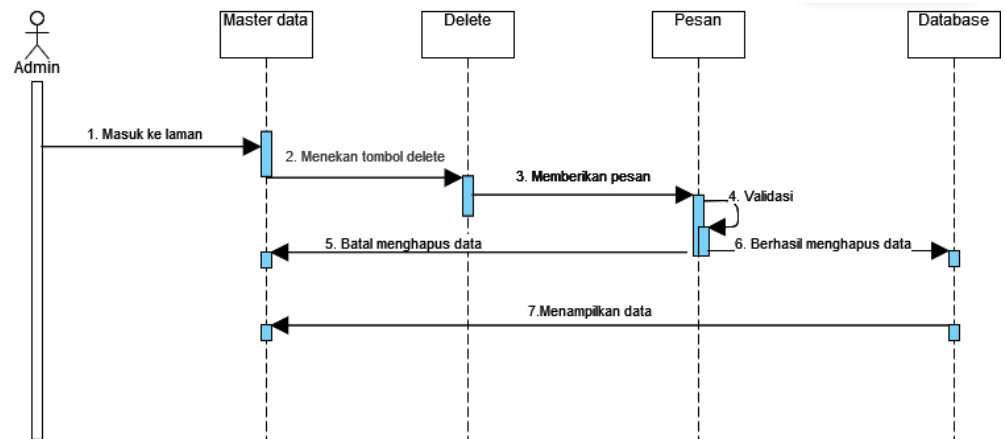


Gambar 4.18 Sequence diagram edit master data



#### 4. Sequence diagram hapus master data

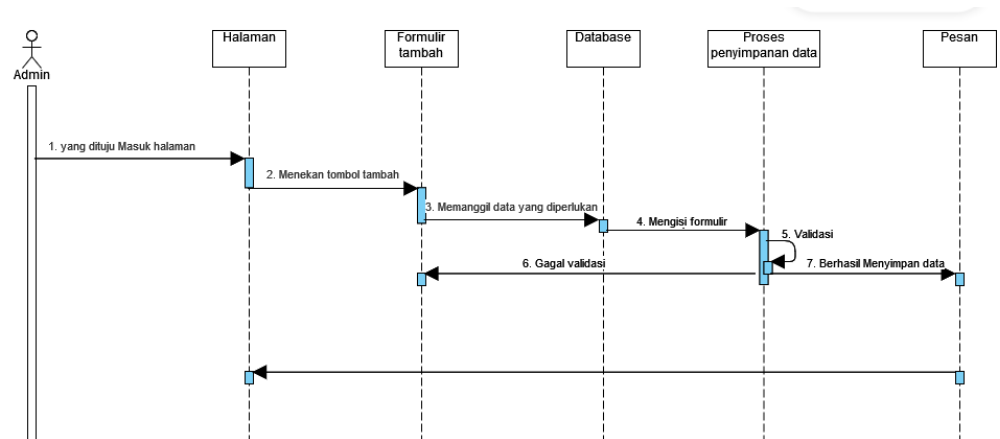
Diagram alur kerja admin mengedit master data, dimulai dengan admin masuk ke laman master data kemudian admin dapat menghapus data dengan menekan tombol hapus. Setelah menekan tombol, admin akan diberikan validasi berupa pesan pop up yang akan ditampilkan. Namun, jika validasi gagal atau admin menekan tombol batal maka admin akan diarahkan kembali ke laman master data, seperti yang ditunjukkan pada gambar 4.19.



Gambar 4.19 Sequence diagram hapus master data

#### 5. Sequence diagram tambah siswa

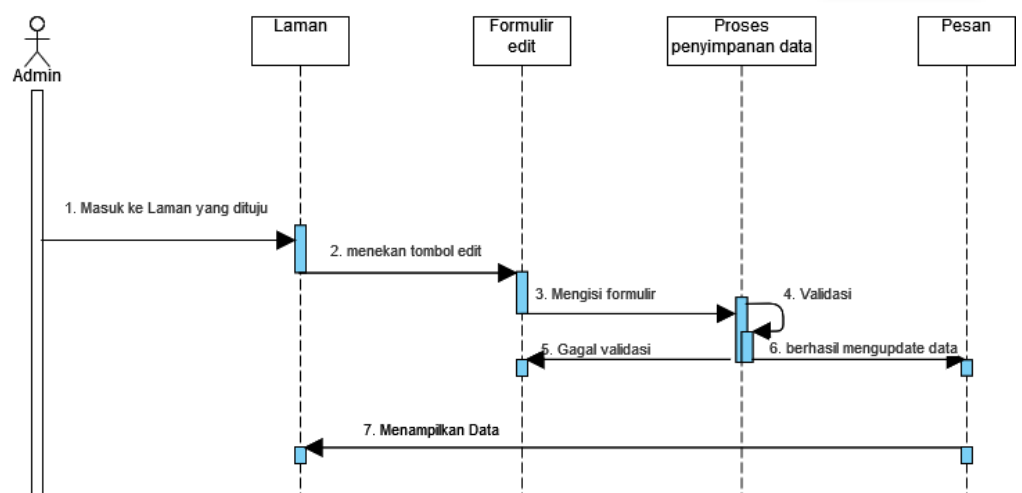
Diagram alur kerja admin menambahkan data siswa, dimulai dengan admin masuk ke laman siswa, kemudian admin dapat menambahkan data dengan mengisi informasi baru di dalam form penambahan data siswa. Setelah proses penyimpanan berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman penambahan informasi data siswa, seperti yang ditunjukkan pada gambar 4.20.



Gambar 4.20 Sequence diagram tambah siswa

#### 6. Sequence diagram edit siswa

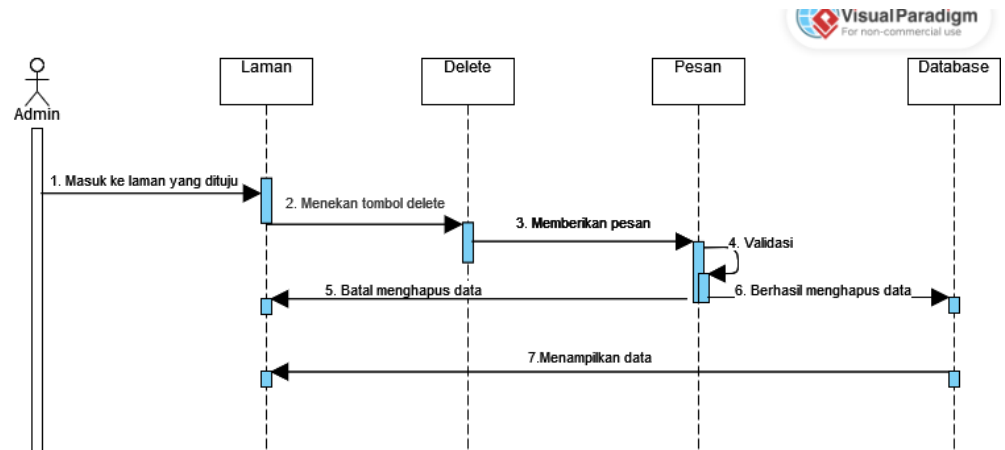
Diagram alur kerja admin mengedit siswa, dimulai dengan admin masuk ke laman siswa kemudian admin dapat mengedit data dengan menekan tombol edit, data yang sebelumnya tersimpan akan ditampilkan dan admin dapat mengisi informasi baru. Setelah proses edit berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman pengeditan informasi siswa, seperti yang ditunjukkan pada gambar 4.21.



Gambar 4.21 Sequence diagram edit siswa

### 7. Sequence diagram hapus siswa

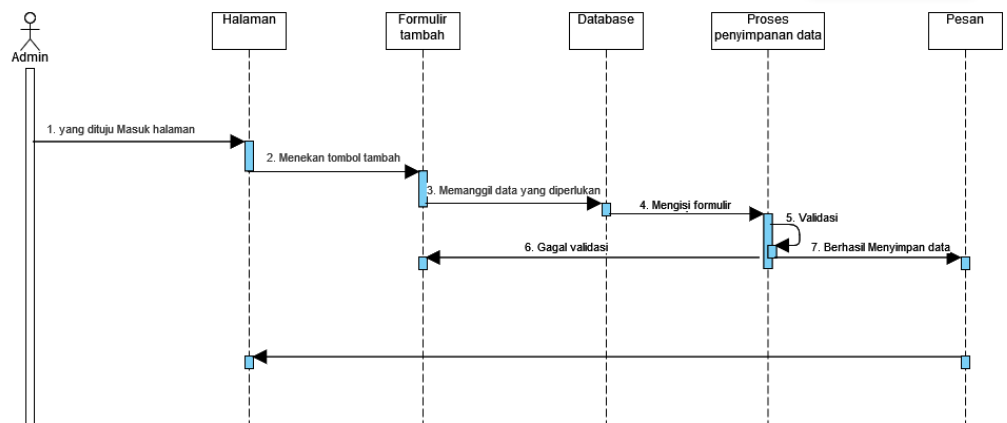
Diagram alur kerja admin menghapus data siswa, dimulai dengan admin masuk ke laman siswa kemudian admin dapat menghapus data dengan menekan tombol hapus. Setelah menekan tombol, admin akan diberikan validasi berupa pesan pop up yang akan ditampilkan. Namun, jika validasi gagal atau admin menekan tombol batal maka admin akan diarahkan kembali ke laman siswa, seperti yang ditunjukkan pada gambar 4.22.



Gambar 4.22 Sequence diagram delete siswa

### 8. Sequence diagram tambah guru

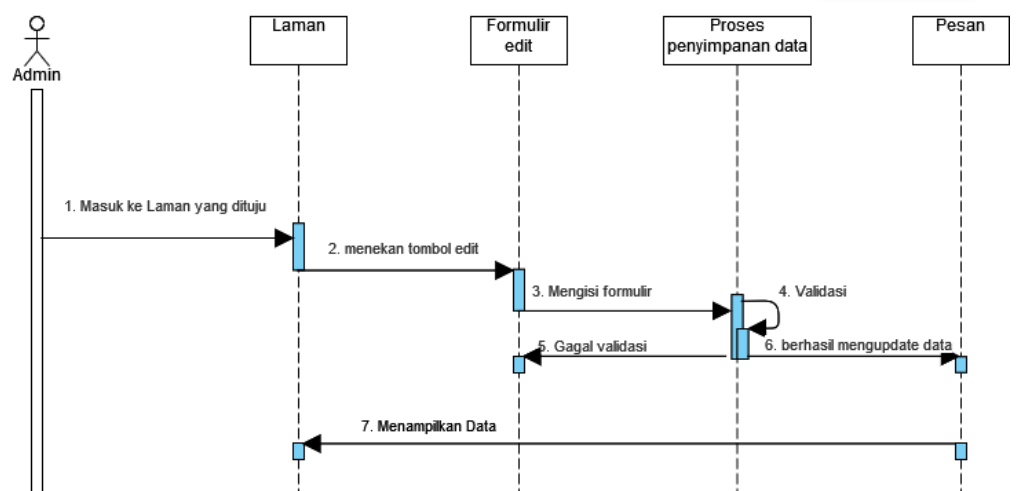
Diagram alur kerja admin menambahkan data guru, dimulai dengan admin masuk ke laman guru, kemudian admin dapat menambahkan data dengan mengisi informasi baru di dalam form penambahan data guru. Setelah proses penyimpanan berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman penambahan informasi data guru, seperti yang ditunjukkan pada gambar 4.23.



Gambar 4.23 Sequence diagram tambah guru

### 9. Sequence diagram edit guru

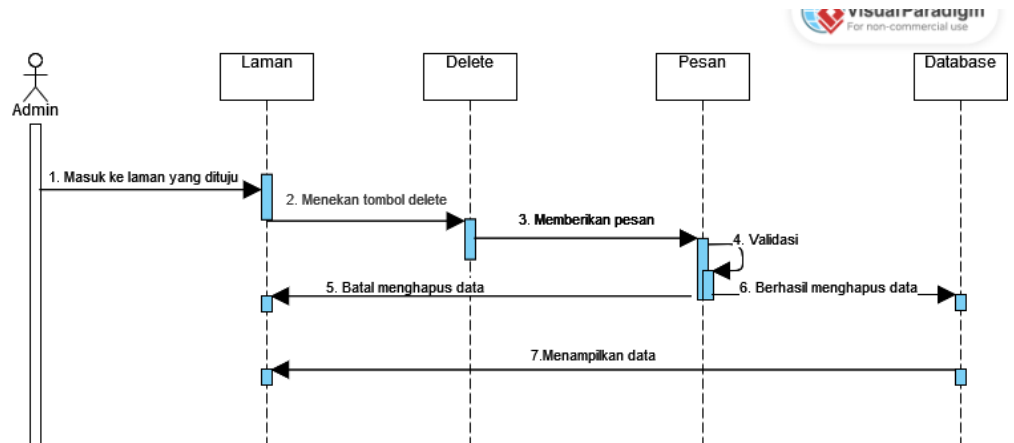
Diagram alur kerja admin mengedit guru, dimulai dengan admin masuk ke laman guru kemudian admin dapat mengedit data dengan menekan tombol edit, data yang sebelumnya tersimpan akan ditampilkan dan admin dapat mengisi informasi baru. Setelah proses edit berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman pengeditan informasi guru, seperti yang ditunjukkan pada gambar 4.24.



Gambar 4.24 Sequence diagram edit guru

### 10. Sequence diagram hapus guru

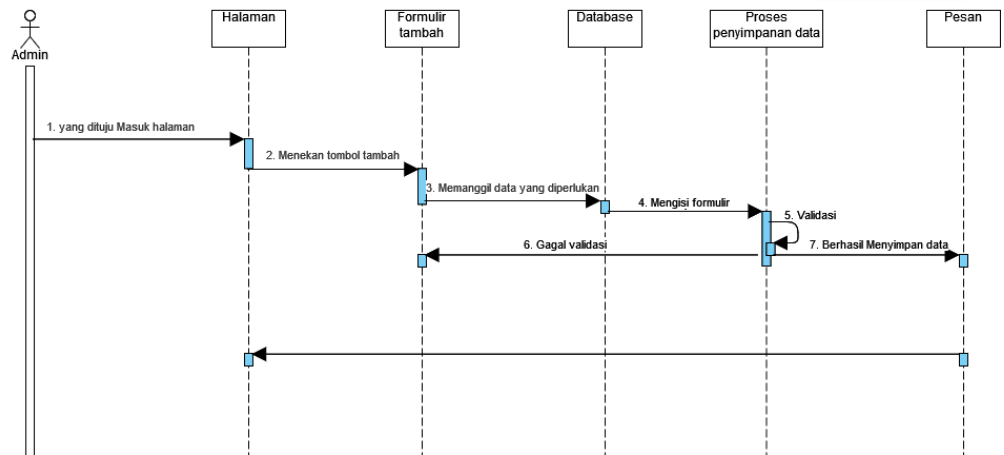
Diagram alur kerja admin menghapus data guru, dimulai dengan admin masuk ke laman guru kemudian admin dapat menghapus data dengan menekan tombol hapus. Setelah menekan tombol, admin akan diberikan validasi berupa pesan pop up yang akan ditampilkan. Namun, jika validasi gagal atau admin menekan tombol batal maka admin akan diarahkan kembali ke laman guru, seperti yang ditunjukkan pada gambar 4.25.



Gambar 4.25 Sequence diagram hapus guru

### 11. Sequence diagram tambah jadwal

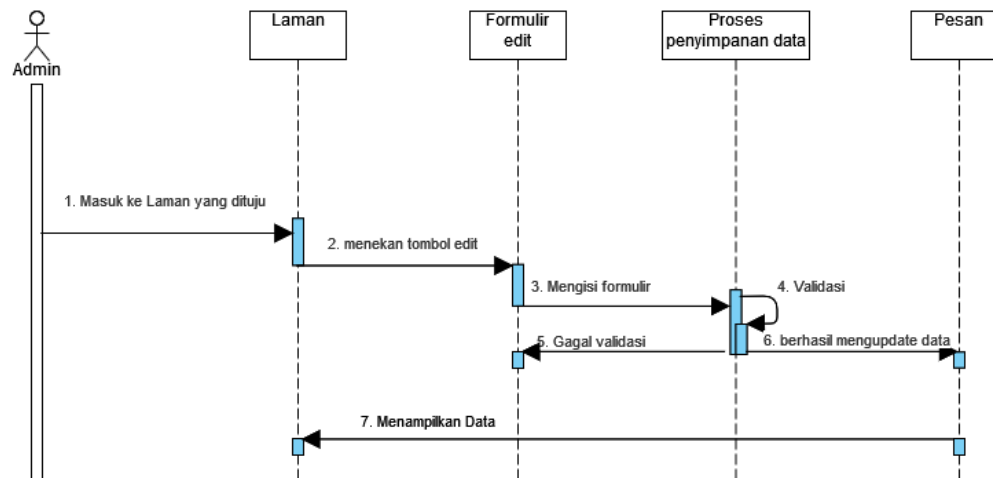
Diagram alur kerja admin menambahkan data jadwal, dimulai dengan admin masuk ke laman jadwal, kemudian admin dapat melihat semua data yang sebelumnya tersimpan di dalam jadwal akan ditampilkan kembali di setiap kolom penambahkan data agar dapat memudahkan admin dalam pengisian jadwal. Setelah proses penyimpanan berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman penambahan informasi data jadwal, seperti yang ditunjukkan pada gambar 4.26.



Gambar 4.26 Sequence diagram tambah jadwal

## 12. Sequence diagram edit jadwal

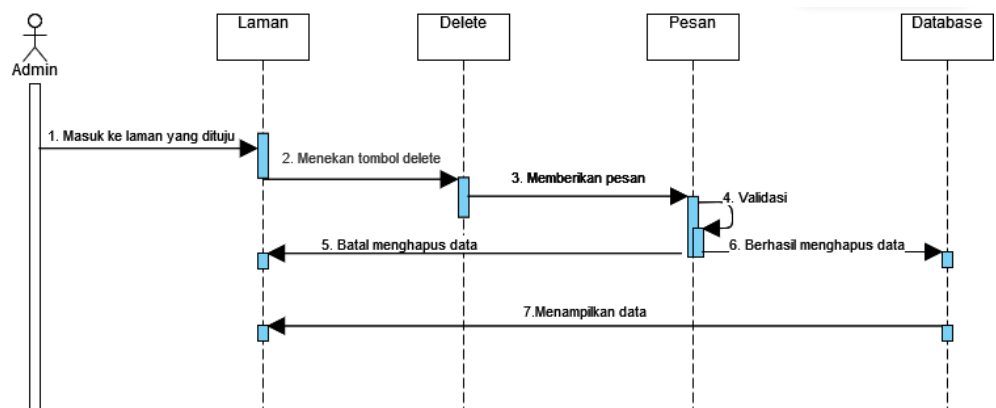
Diagram alur kerja admin mengedit guru, dimulai dengan admin masuk ke laman jadwal, kemudian admin dapat mengedit data dengan menekan tombol edit, data yang sebelumnya tersimpan akan ditampilkan dan admin dapat mengisi informasi baru. Setelah proses edit berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman pengeditan informasi guru, seperti yang ditunjukkan pada gambar 4.27.



Gambar 4.27 Sequence diagram edit jadwal

### 13. Sequence diagram hapus jadwal

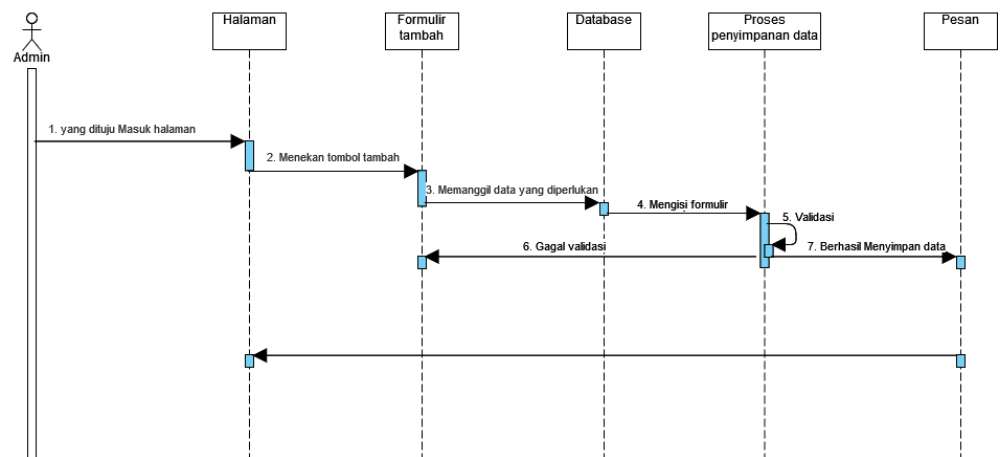
Diagram alur kerja admin menghapus data jadwal, dimulai dengan admin masuk ke laman jadwal kemudian admin dapat menghapus data dengan menekan tombol hapus. Setelah menekan tombol, admin akan diberikan validasi berupa pesan pop up yang akan ditampilkan. Namun, jika validasi gagal atau admin menekan tombol batal maka admin akan diarahkan kembali ke laman jadwal, seperti yang ditunjukkan pada gambar 4.28.



Gambar 4.28 Sequence diagram hapus jadwal

### 14. Sequence diagram tambah absen siswa

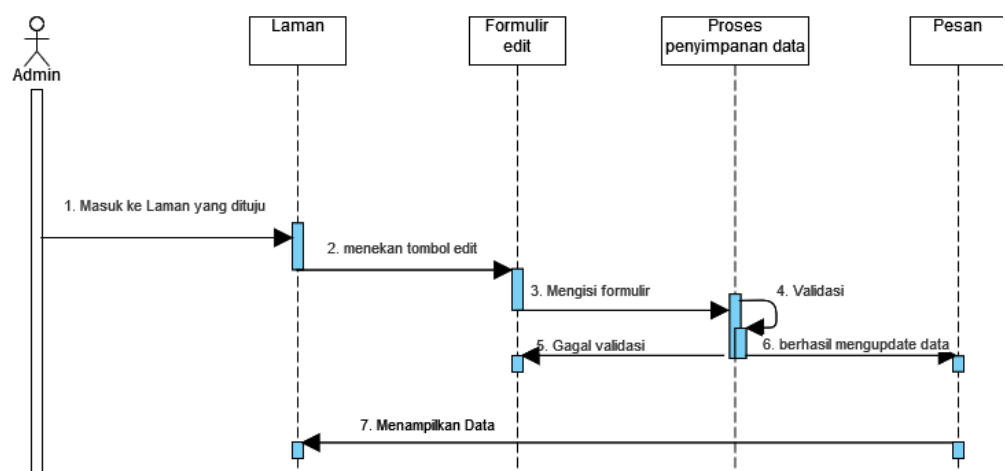
Diagram alur kerja admin menambahkan data absensi siswa, dimulai dengan admin masuk ke laman absensi siswa, kemudian admin dapat melihat semua data siswa yang tersimpan di dalam kolom penambahan absensi siswa agar dapat memudahkan admin dalam pengisian absensi. Setelah proses penyimpanan berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman penambahan informasi data jadwal, seperti yang ditunjukkan pada gambar 4.29.



Gambar 4.29 Sequence diagram tambah absen siswa

#### 15. Sequence diagram edit absen siswa

Diagram alur kerja admin mengedit absen siswa, dimulai dengan admin masuk ke laman absen siswa kemudian admin dapat mengedit data dengan menekan tombol edit, data yang sebelumnya tersimpan akan ditampilkan dan admin dapat mengisi informasi baru. Setelah proses edit berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman pengeditan informasi guru, seperti yang ditunjukkan pada gambar 4.30.

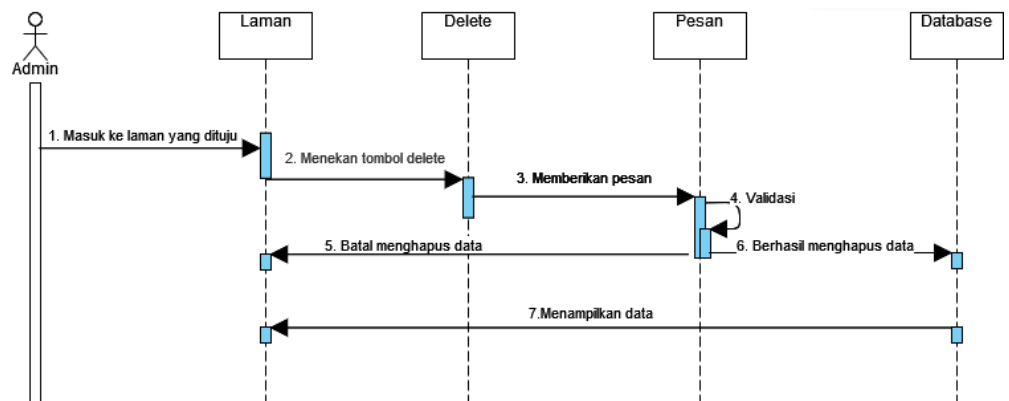


Gambar 4.30 Sequecen diagram edit absen siswa



### 16. Sequence diagram hapus absen siswa

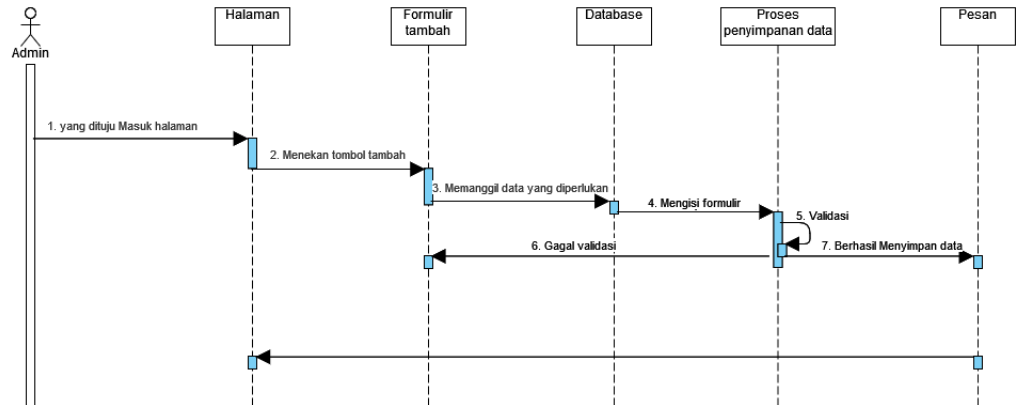
Diagram alur kerja admin menghapus data absen siswa, dimulai dengan admin masuk ke laman absen siswa kemudian admin dapat menghapus data dengan menekan tombol hapus. Setelah menekan tombol, admin akan diberikan validasi berupa pesan pop up yang akan ditampilkan. Namun, jika validasi gagal atau admin menekan tombol batal maka admin akan diarahkan kembali ke laman absen siswa, seperti yang ditunjukkan pada gambar 4.31.



Gambar 4.31 Sequence diagram delete absen siswa

### 17. Sequence diagram tambah user data

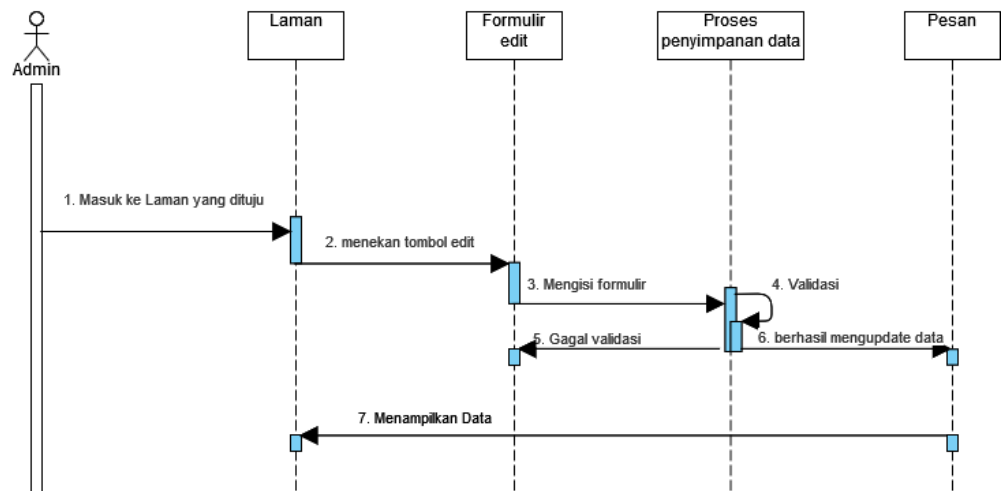
Diagram alur kerja admin menambahkan akun user di sini dimulai dengan admin dapat masuk ke laman user data kemudian admin dapat menambahkan akun user dengan mengisi informasi akun baru melalui data siswa dan guru yang sebelumnya sudah tersimpan. Setelah proses penyimpanan berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman pengisian informasi admin, seperti yang ditunjukkan pada gambar 4.32.



Gambar 4.32 Sequence diagram tambah user data

#### 18. Sequence diagram edit user data

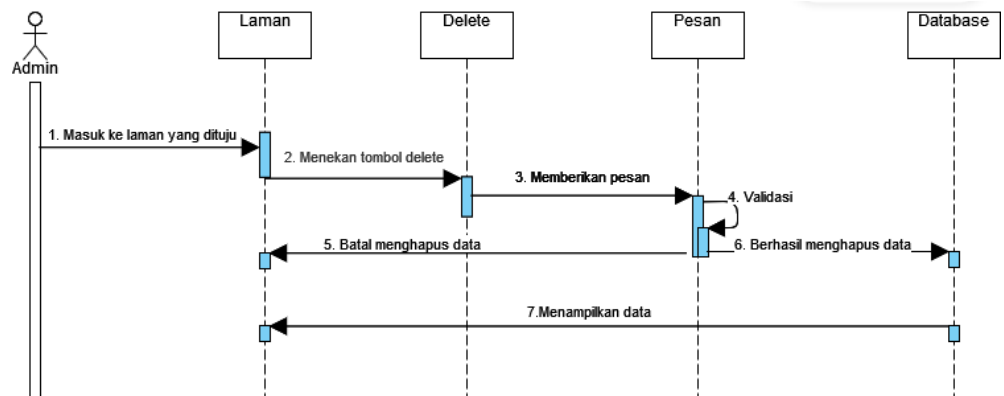
Diagram alur kerja admin mengedit user data, dimulai dengan admin masuk ke laman user data, kemudian admin dapat mengedit data dengan menekan tombol edit, data yang sebelumnya tersimpan akan dipanggil dan admin dapat mengisi informasi baru. Setelah proses edit berhasil, pesan sukses akan ditampilkan. Namun, jika validasi gagal, admin akan diarahkan kembali ke laman pengeditan informasi user data, seperti yang ditunjukkan pada gambar 4.33.



Gambar 4.33 Sequence diagram edit user data

### 19. Sequence diagram hapus user

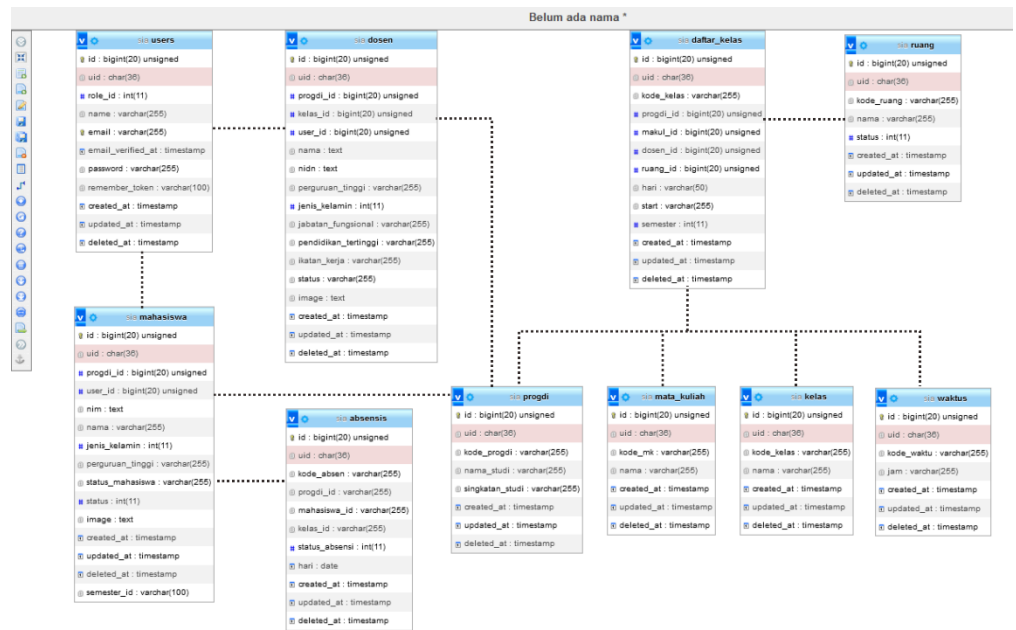
Diagram alur kerja admin menghapus data user data, dimulai dengan admin masuk ke laman user data kemudian admin dapat menghapus data dengan menekan tombol hapus. Setelah menekan tombol, admin akan diberikan validasi berupa pesan pop up yang akan ditampilkan. Namun, jika validasi gagal atau admin menekan tombol batal maka admin akan diarahkan kembali ke laman user data, seperti yang ditunjukkan pada gambar 4.34.



Gambar 4.34 Sequence diagram hapus user data

#### d. Class Diagram

Class diagram berfungsi untuk menjelaskan tipe data dari objek di dalam sistem dan hubungan antar objek lain. Class diagram sistem penjadwalan dapat dilihat pada gambar 4.35.

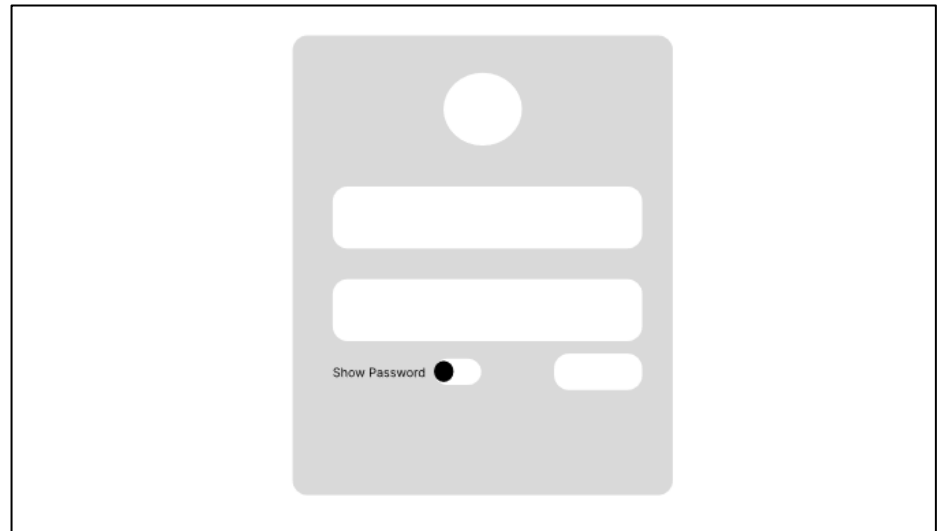


Gambar 4. 35 Class diagram

## 2. Desain Perancangan Interface

### 1) Rancangan tampilan login

Rancangan tampilan halaman login. Form login berisi email dan password serta tombol untuk login. Rancangan tampilan login dapat dilihat pada gambar 4.36 berikut :



Gambar 4.36 Desain login

## 2) Rancangan tampilan dashboard

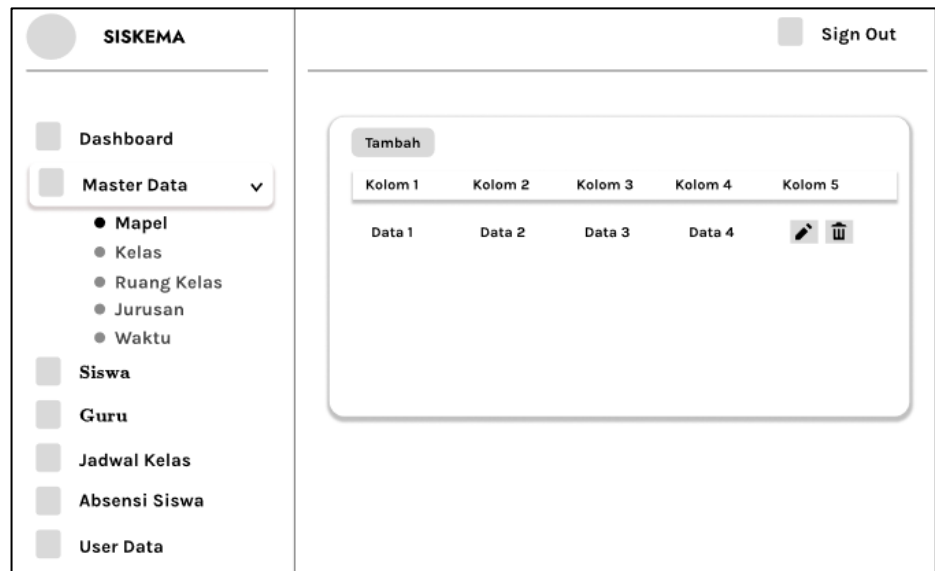
Rancangan tampilan dashboard admin berisi judul web, daftar menu atau sidebar, informasi mengenai jumlah siswa dan dosen yang terdaftar pada database. Rancangan tampilan dashboard admin dapat dilihat pada gambar 4.37 :



Gambar 4.37 Desain dashboard

### 3) Rancangan master data

Rancangan tampilan master data berisi judul web, daftar menu, table data, tombol tambah dan hapus data. Rancangan tampilan halaman master data dapat dilihat pada gambar 4.38 Berikut :



Gambar 4.38 Desain tampilan master data

### 4) Rancangan tambah master data

Rancangan tampilan tambah master data berisi form untuk menambahkan data dari masing masing fitur yang ada serta dilengkapi dengan tombol save dan tombol kembali. Rancangan tampilan halaman tambah dapat di lihat pada gambar 4.39 berikut:

Gambar 4.39 Desain tambah master data

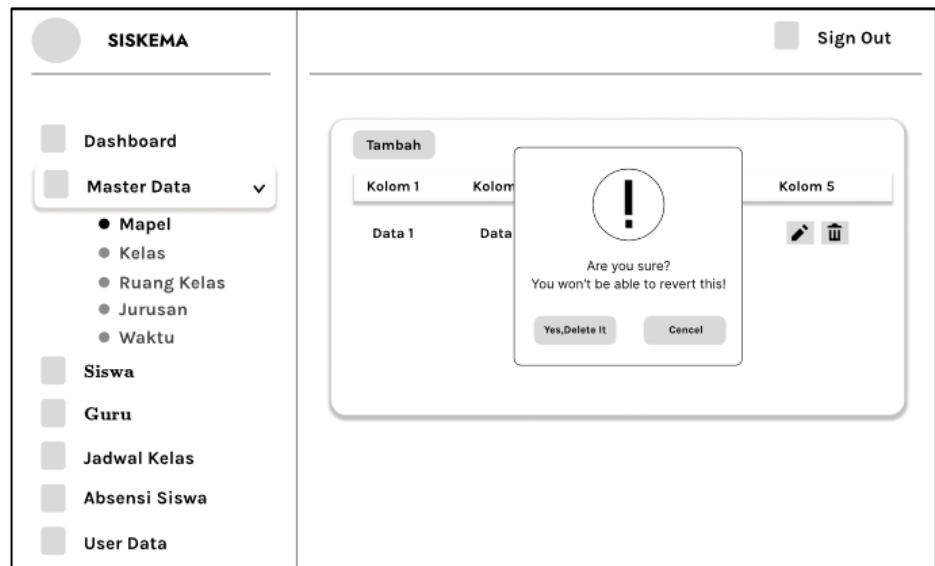
## 5) Rancangan edit master data

Rancangan edit master data berisi form untuk mengedit data dari masing masing fitur yang ada seperti mapel, kelas, ruang kelas, jurusan dan waktu serta dilengkapi dengan tombol save dan kembali. Rancangan tampilan halaman edit master data dapat di lihat pada gambar 4.40 :

Gambar 4.40 Desain edit master data

6) Rancangan hapus master data

Rancangan tampilan hapus master data berupa pop up setelah admin menekan tombol hapus pada halaman. Rancangan tampilan halaman hapus master data dapat di lihat pada gambar 4.41 :



Gambar 4.41 Desain delete master data

7) Rancangan tampilan siswa

Rancangan tampilan siswa berisi judul web, daftar menu, cari data, table data, tombol tambah dan hapus data. Rancangan tampilan halaman siswa dapat dilihat pada gambar 4.42 Berikut :

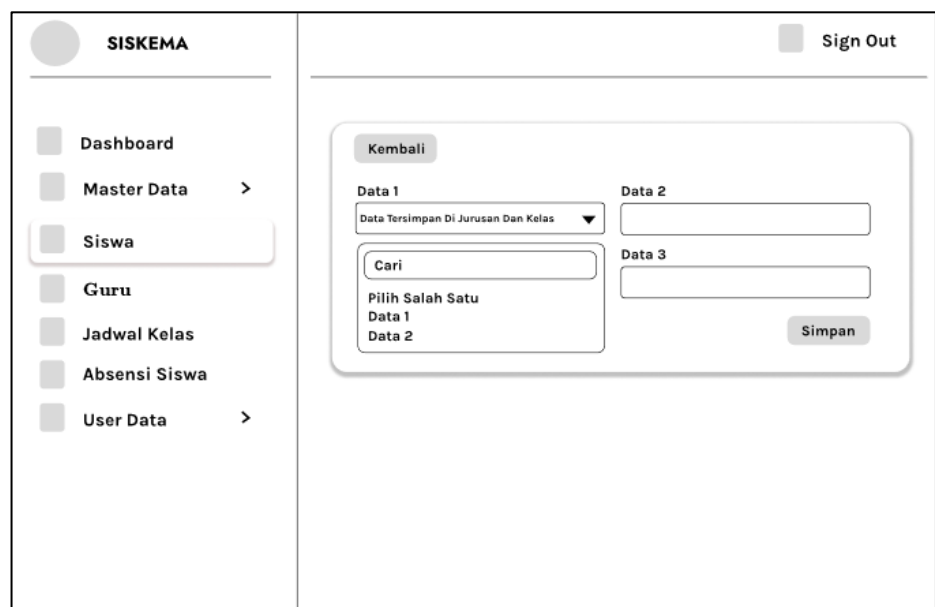




Gambar 4.42 Desain tampilan siswa

## 8) Rancangan tambah siswa

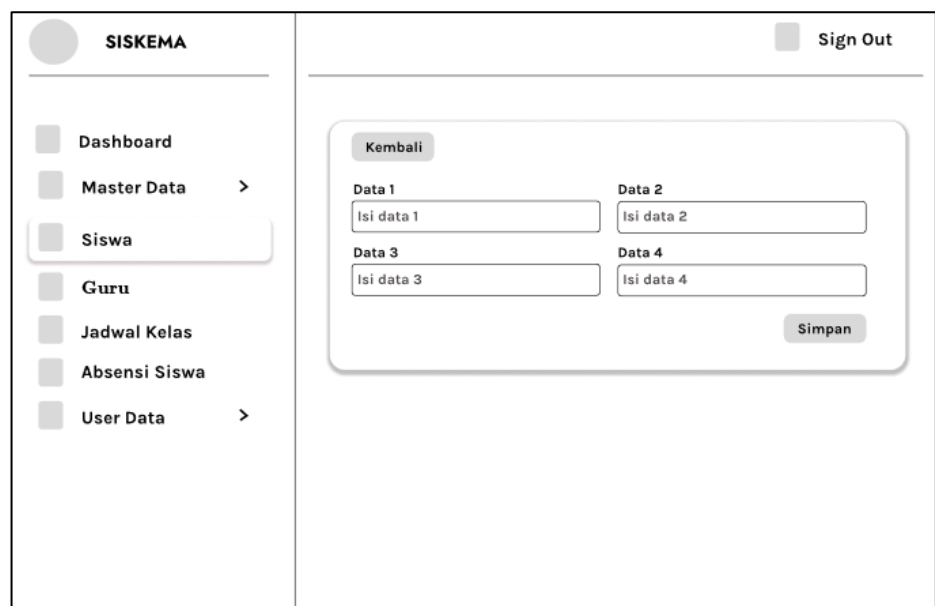
Rancangan tampilan tambah siswa berisi form untuk menambahkan data dari siswa yang dilengkapi fitur menampilkan data dari jurusan dan kelas serta memiliki tombol save dan kembali. Rancangan tampilan halaman tambah dapat di lihat pada gambar 4.43 berikut:



Gambar 4.43 Desain tambah siswa

## 9) Rancangan edit siswa

Rancangan tampilan edit siswa berisi form untuk mengedit data dari siswa yang sebelumnya sudah tersimpan akan di tampilkan kembali dan admin bisa memulai mengedit data dengan memasukan data baru kemudian menekan tombol save untuk menyimpan data yang baru dirubah atau menekan tombol kembali untuk kembali ke halaman beranda dari siswa. Rancangan tampilan halaman edit siswa dapat di lihat pada gambar 4.44 berikut:

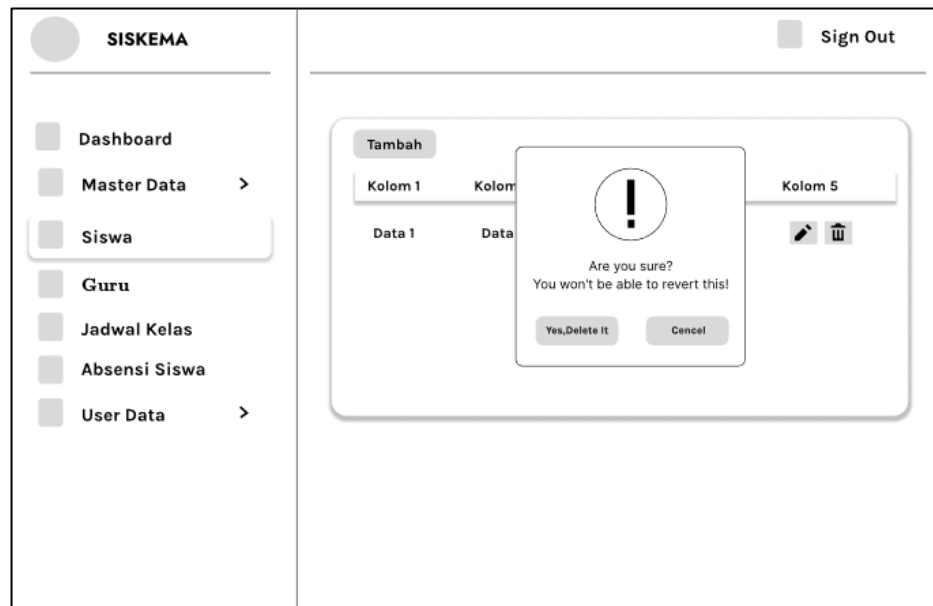


The image shows a web application interface for editing student data. On the left is a sidebar menu for 'SISKEMA' with items: Dashboard, Master Data, Siswa (highlighted), Guru, Jadwal Kelas, Absensi Siswa, and User Data. The main area has a 'Sign Out' button in the top right. A central form titled 'Kembali' contains four input fields: 'Data 1' (Isi data 1), 'Data 2' (Isi data 2), 'Data 3' (Isi data 3), and 'Data 4' (Isi data 4). A 'Simpan' button is at the bottom right of the form.

Gambar 4.44 Desain edit siswa

## 10) Rancangan hapus siswa

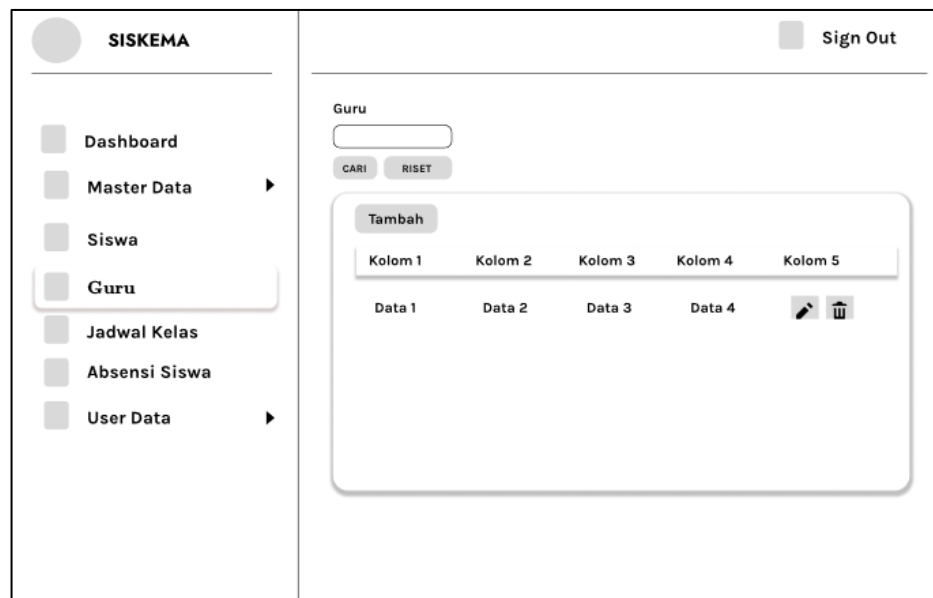
Rancangan tampilan hapus siswa berupa pop up setelah admin meekan tombol hapus pada tampilan. Rancangan tampilan halaman hapus data siswa dapat di lihat pada gambar 4.45 berikut:



Gambar 4.45 Desain delete siswa

## 11) Rancangan tampilan guru

Rancangan tampilan guru berisi judul web, daftar menu, cari data, table data, tombol tambah dan hapus data. Rancangan tampilan halaman guru dapat dilihat pada gambar 4.46 Berikut :



Gambar 4.46 Desain tampilan guru

### 12) Rancangan tambah guru

Rancangan tampilan tambah guru berisi form untuk menambahkan data dari guru yang dilengkapi fitur menampilkan data dari jurusan dan kelas agar memudahkan admin dalam menambahkan data serta memiliki tombol save untuk menyimpan data dan tombol kembali untuk kembali ke halaman beranda dari guru. Rancangan tampilan halaman tambah dapat di lihat pada gambar 4.47 berikut:

Gambar 4.47 Desain tambah guru

### 13) Rancangan edit guru

Rancangan tampilan edit guru berisi form untuk mengedit data dari guru yang sebelumnya sudah tersimpan akan di tempilkan kembali dan admin bisa memulai mengedit data dengan memasukan data baru kemudian menekan tombol save untuk menyimpan data yang baru dirubah atau menekan tombol kembali untuk kembali ke halaman beranda dari guru. Rancangan tampilan halaman edit siswa dapat di lihat pada gambar 4.48 berikut:

Gambar 4.48 Desain edit guru

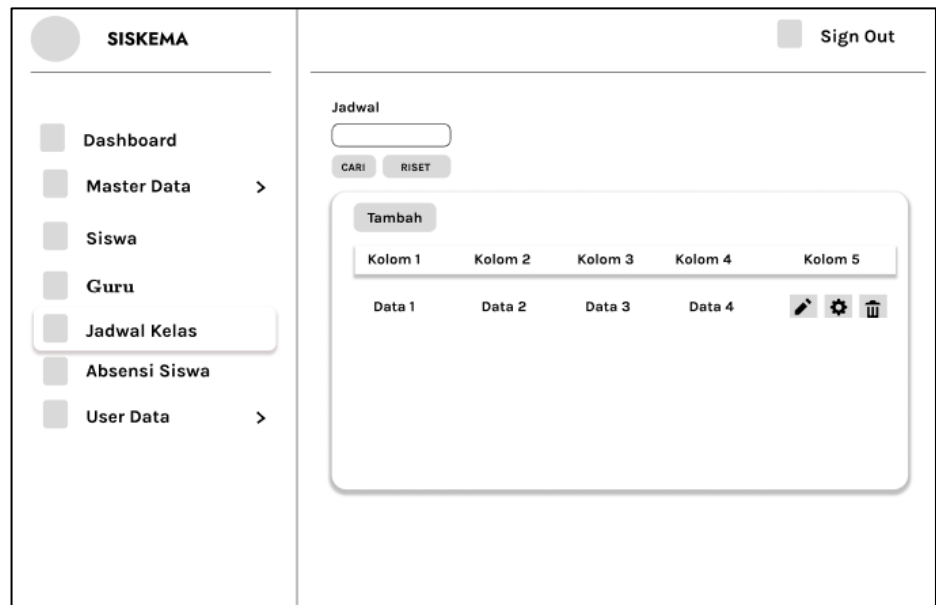
## 14) Rancangan hapus guru

Rancangan tampilan hapus guru berupa pop up setelah admin meekan tombol hapus pada tampilan. Rancangan tampilan halaman hapus data guru dapat di lihat pada gambar 4.49 berikut:

Gambar 4.49 Desain delete guru

### 15) Rancangan tampilan Jadwal

Rancangan tampilan jadwal berisi judul web, daftar menu, cari data, table data, tombol tambah dan hapus data. Rancangan tampilan halaman jadwal dapat dilihat pada gambar 4.50 Berikut :



Gambar 4.50 Desain tampilan jadwal

### 16) Rancangan tambah jadwal

Rancangan tampilan tambah jadwal berisi form untuk menambahkan data dari jadwal yang dilengkapi fitur menampilkan data dari master data agar memudahkan admin dalam menambahkan data serta memiliki tombol save untuk menyimpan data dan tombol kembali untuk kembali ke halaman beranda dari jadwal. Rancangan tampilan halaman tambah dapat di lihat pada gambar 4.51 berikut:

The image shows a web application interface for 'SISKEMA'. On the left is a sidebar with a menu: Dashboard, Master Data, Siswa, Guru, Jadwal Kelas (highlighted), Absensi Siswa, and User Data. The top right has a 'Sign Out' button. The main area contains a 'Kembali' button and a form with four data sections: Data 1, Data 2, Data 3, and Data 4. Each section has a dropdown menu (e.g., 'Data Tersimpan Di Master Data'), a search box labeled 'Cari', and a selection box labeled 'Pilih Salah Satu'. A 'Simpan' button is located at the bottom right of the form.

Gambar 4.51 Desain tambah jadwal

## 17) Rancangan edit jadwal

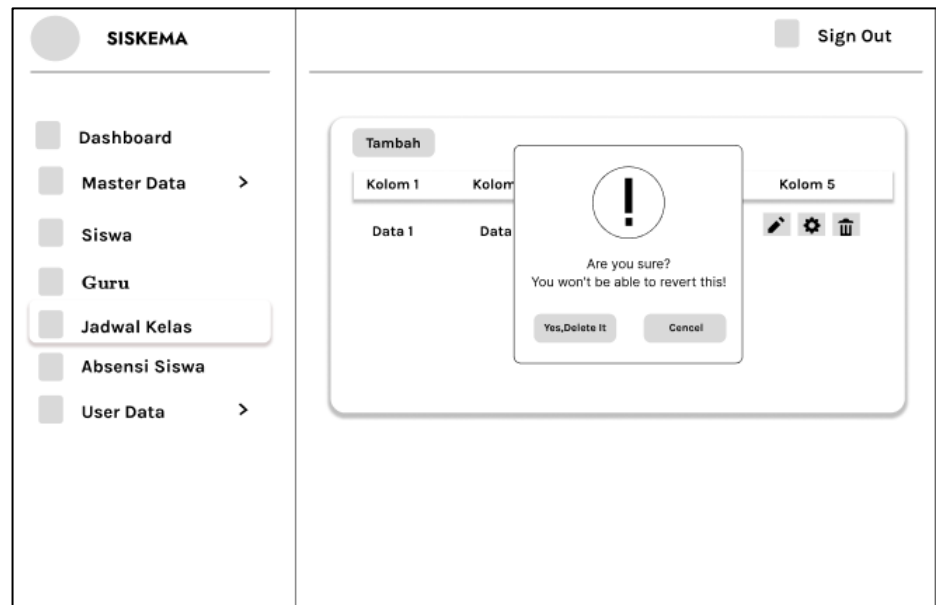
Rancangan edit jadwal berisi form untuk mengedit data yang sebelumnya sudah tersimpan dilengkapi dengan tombol save dan kembali. Rancangan halaman edit jadwal dapat di lihat pada gambar 4.52 :

This image shows the 'Edit Jadwal' form in the SISKEMA application. It features the same sidebar and 'Sign Out' button as Gambar 4.51. The main form area includes a 'Kembali' button at the top left. The data input sections (Data 1, Data 2, Data 3, Data 4) are identical to those in Gambar 4.51, each containing a dropdown menu, a search box, and a selection box. A 'Simpan' button is positioned at the bottom right of the form.

Gambar 4.52 Desain edit jadwal

## 18) Rancangan hapus jadwal

Rancangan tampilan hapus jadwal berupa pop up setelah admin meekan tombol hapus pada tampilan. Rancangan tampilan halaman hapus data jadwal dapat di lihat pada gambar 4.53 berikut:



Gambar 4.53 Desain delete jadwal

## 19) Rancangan tampilan absensi siswa

Rancangan tampilan absensi siswa berisi judul web, daftar menu, table data, tombol tambah dan hapus data. Rancangan tampilan halaman absensi siswa dapat dilihat pada gambar 4.54 Berikut :

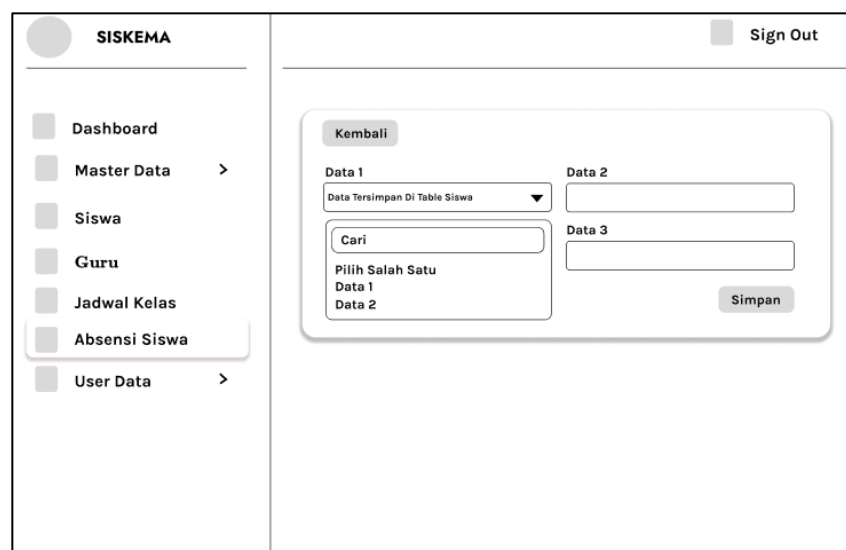




Gambar 4.54 Desain tampilan absensi ssiwa

## 20) Rancangan tambah absensi siswa

Rancangan tampilan tambah absensi siswa berisi form untuk menambahkan data dari absensi siswa yang dilengkapi fitur menampilkan data dari siswa, tombol save dan tombol kembali. Rancangan tampilan halaman tambah dapat di lihat pada gambar 4.55 :



Gambar 4.55 Desain tampilan tambah absensi siswa

## 21) Rancangan edit absensi siswa

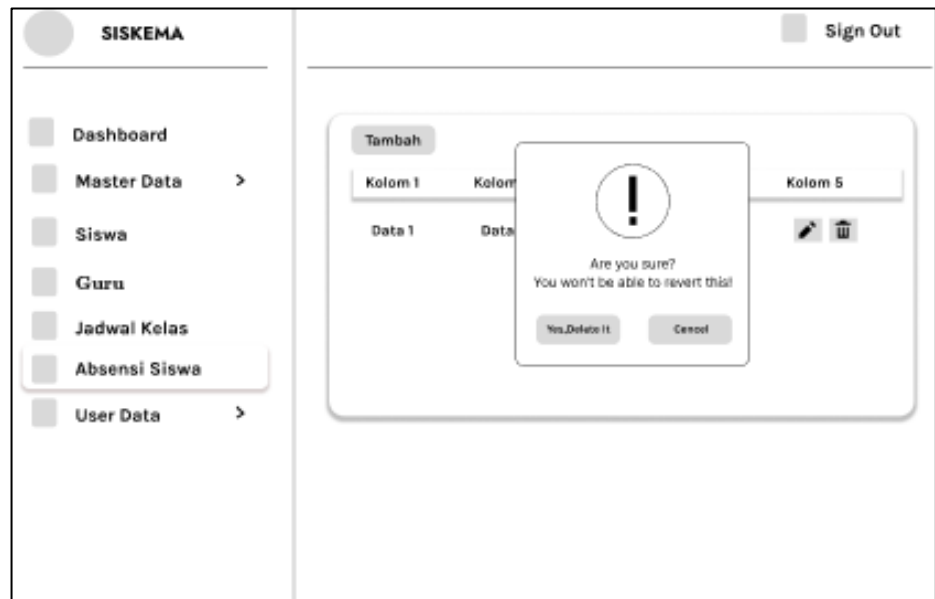
Rancangan tampilan edit absen siswa berisi form untuk mengedit data dari absen siswa yang sebelumnya sudah tersimpan akan di tampilkan kembali dan admin bisa memulai mengedit data dengan memasukan data baru kemudian menekan tombol save untuk menyimpan data yang baru dirubah atau menekan tombol kembali untuk kembali ke halaman beranda dari absensi siswa. Rancangan tampilan halaman edit absen siswa dapat di lihat pada gambar 4.56 berikut:

The image shows a web application interface for editing student absences. On the left is a sidebar menu titled 'SISKEMA' with a 'Sign Out' button in the top right. The menu items are: Dashboard, Master Data, Siswa, Guru, Jadwal Kelas, Absensi Siswa (highlighted), and User Data. The main content area is titled 'Kembali' and contains a form with three data input fields: 'Data 1' (containing 'Data Tersimpan 1'), 'Data 2' (containing 'Data Tersimpan 2'), and 'Data 3' (containing 'Data Tersimpan 3'). Below these is a search box labeled 'Cari' and a radio button labeled 'Pilih Salah Satu' with options 'Data 1' and 'Data 2'. A 'Simpan' button is located at the bottom right of the form.

Gambar 4.56 Desain edit absensi siswa

## 22) Rancnagan hapus absensi siswa

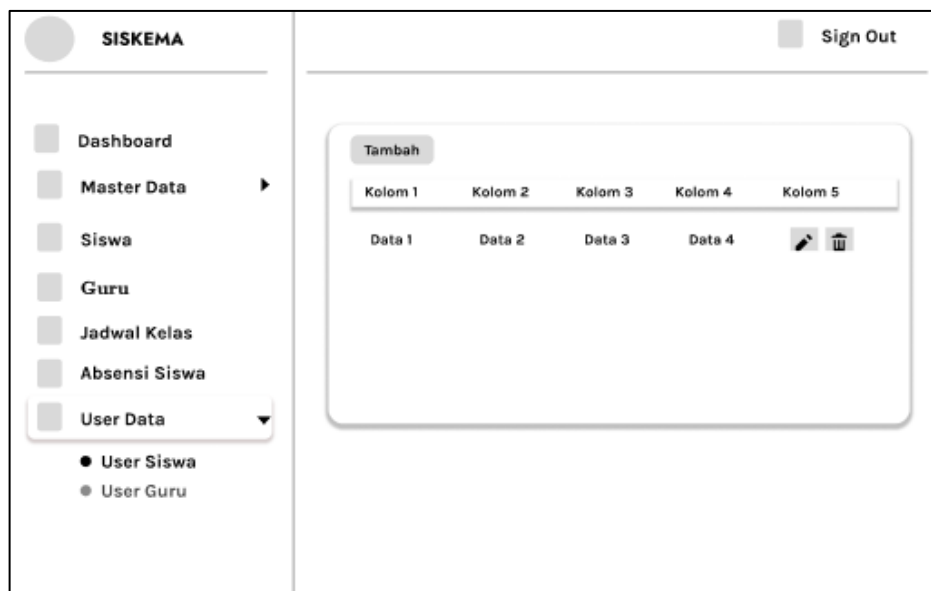
Rancangan tampilan hapus absensi siswa berupa pop up setelah admin meekan tombol hapus pada tampilan. Rancangan tampilan halaman tambah dapat di lihat pada gambar 4.57 berikut:



Gambar 4.57 Desain tampilan delete absensi siswa

### 23) Rancangan tampilan user data

Rancangan tampilan user data berisi judul web, daftar menu, table data, tombol tambah dan hapus data. Rancangan tampilan halaman user data dapat dilihat pada gambar 4.58 Berikut :



Gambar 4.58 Desain tampilan user data

#### 24) Rancangan tambah user data

Rancangan tampilan tambah siswa berisi form untuk menambahkan data dari siswa. Dilengkapi fitur menampilkan data dari jurusan dan kelas agar memudahkan admin dalam menambahkan data serta memiliki tombol save untuk menyimpan data dan tombol kembali untuk kembali ke halaman beranda dari user data. Rancangan tampilan halaman tambah dapat di lihat pada gambar 4.59 berikut:

Gambar 4.59 Desain tambah user data

#### 25) Rancangan edit user data

Rancangan tampilan edit absen siswa berisi form untuk mengedit data dari absen siswa yang sebelumnya sudah tersimpan akan di tampilkan kembali dan admin bisa memulai mengedit data dengan memasukan data baru kemudian menekan tombol save untuk menyimpan data yang baru dirubah atau menekan tombol kembali untuk kembali ke halaman beranda dari user data. Rancangan tampilan halaman edit absen siswa dapat di lihat pada gambar 4.60 berikut:

Gambar 4.60 Desain edit user data

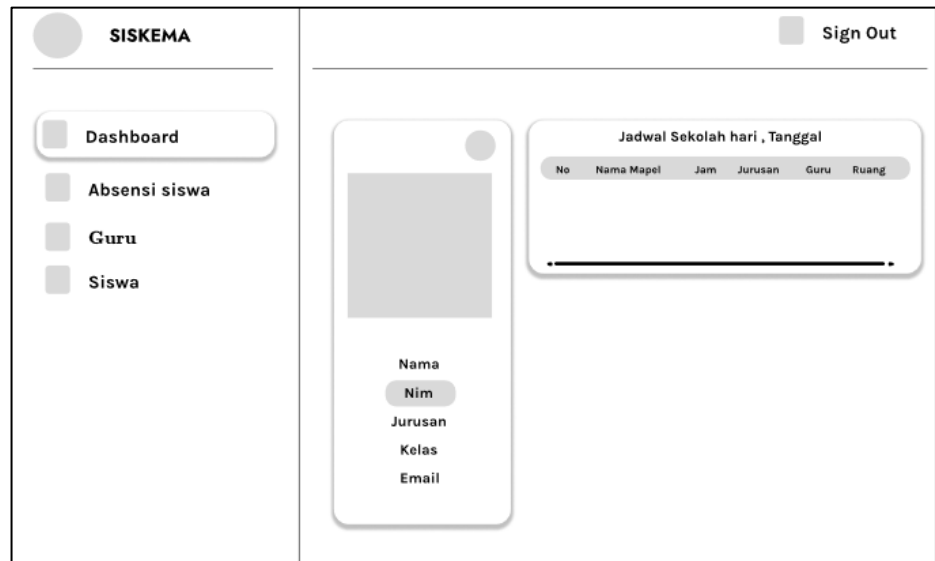
## 26) Rancangan hapus user data

Rancangan tampilan hapus user data berupa pop up setelah admin meekan tombol hapus pada tampilan. Rancangan tampilan halaman hapus user data dapat di lihat pada gambar 4.61 berikut:

Gambar 4.61 Desain tampilan delete user data

## 27) Tampilan dashboard siswa

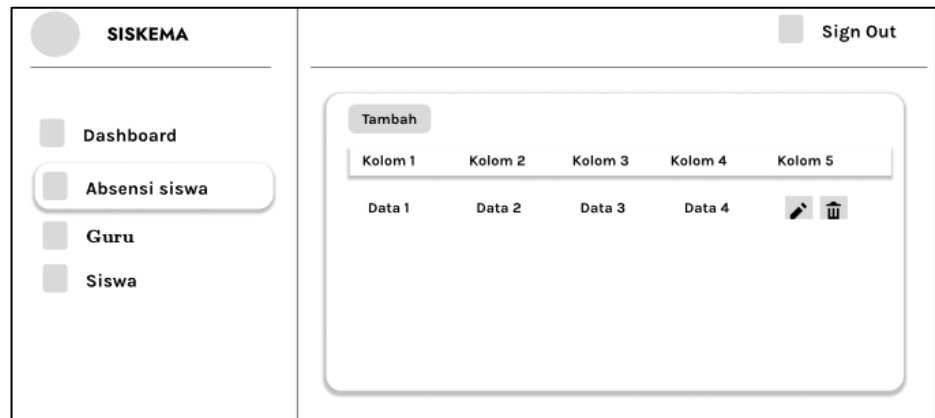
Rancangan tampilan dashboard siswa berisi judul web, daftar menu, tanggal, table data yang berisi jadwal dan profil dari siswa. Rancangan tampilan halaman dashboard siswa dapat dilihat pada gambar 4.62 berikut :



Gambar 4.62 Desain dashboard siswa

## 28) Tampilan absen siswa

Rancangan tampilan absen siswa berisi judul web, daftar menu, table data, tombol tambah dan hapus data. Rancangan tampilan halaman absensi siswa dapat dilihat pada gambar 4.63 Berikut :



Gambar 4.63 Desain absen pribadi siswa

#### 29) Tampilan tambah absen siswa

Rancangan tampilan tambah absen siswa berisi form untuk menambahkan data dari absensi siswa yang dilengkapi fitur menampilkan pribadi dari siswa yang sedang ingin melakukan absensi, memiliki tombol save untuk menyimpan data dan tombol kembali untuk kembali ke halaman beranda dari absensis siswa. Rancangan tampilan halaman tambah dapat di lihat pada gambar 4.64 berikut:



Gambar 4.64 Desain tambah absen pribadi siswa

### 30) Tampilan edit absen siswa

Rancangan tampilan edit absen siswa berisi form untuk mengedit data dari absen siswa yang sebelumnya sudah tersimpan akan di tampilkan kembali dan siswa bisa memulai mengedit data dengan memasukan data baru kemudian menekan tombol save untuk menyimpan data yang baru dirubah atau menekan tombol kembali untuk kembali ke halaman beranda dari absensi siswa. Rancangan tampilan halaman edit absen siswa dapat di lihat pada gambar 4.65 berikut:

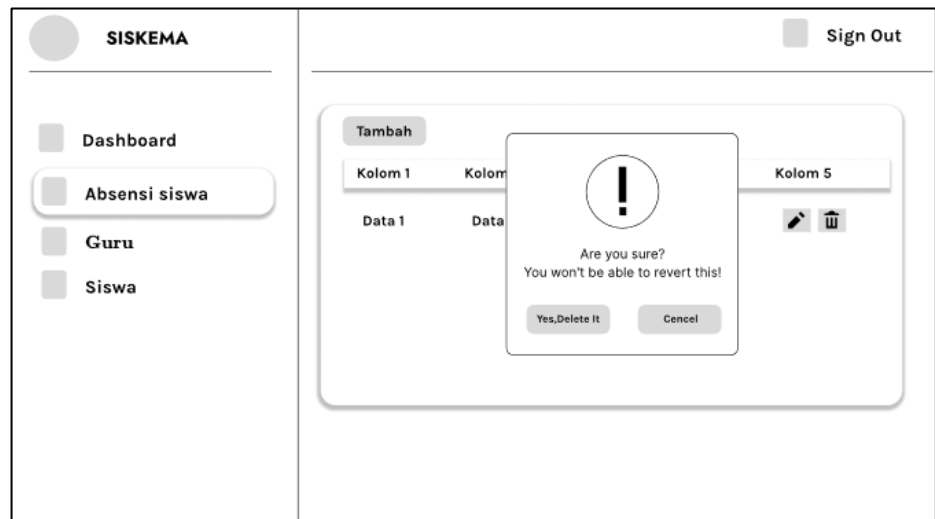


Gambar 4.65 Desain edit absen pribadi siswa

### 31) Tampilan hapus absen siswa

Rancangan tampilan hapus absensi berupa pop up setelah admin meekan tombol hapus pada tampilan. Rancangan tampilan halaman hapus absen siswa dapat di lihat pada gambar 4.66 berikut:

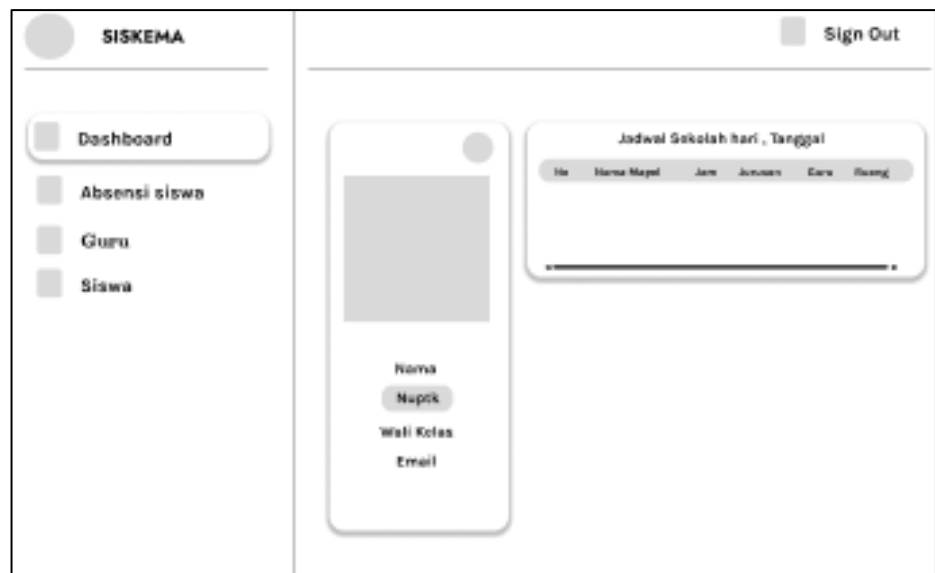




Gambar 4.66 Desain hapus data absen pribadi siswa

### 32) Tampilan dashboard guru

Rancangan tampilan dashboard guru berisi judul web, daftar menu, tanggal, table data yang berisi jadwal dan profil dari guru. Rancangan tampilan halaman dashboard guru dapat dilihat pada gambar 4.67 berikut



Gambar 4.67 Desain dashboard guru

### 33) Tampilan absen siswa

Rancangan tampilan dashboard guru berisi judul web, daftar menu, tanggal, table data dari siswa yang telah melakukan absensi, data dari siswa yang akan di tampilkan akan di filter menurut jurusan dan kelas . Rancangan tampilan halaman dashboard guru dapat dilihat pada gambar 4.68 Berikut :



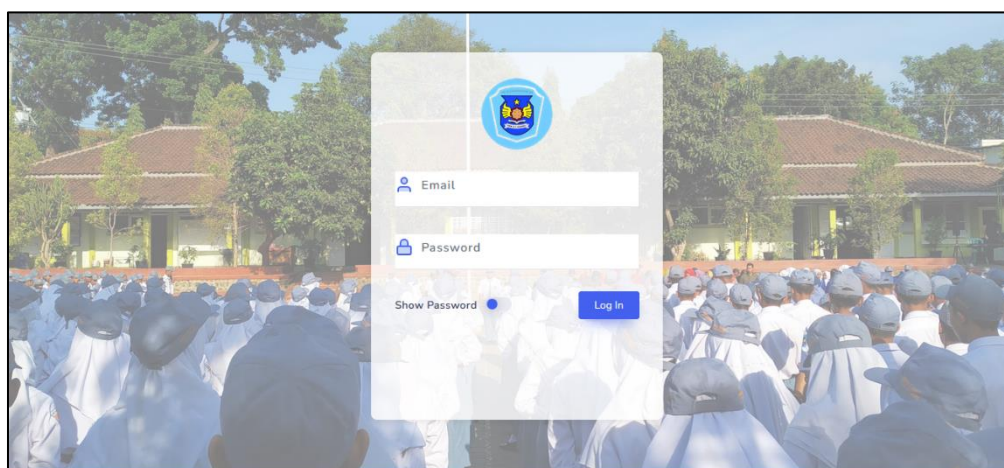
Gambar 4.68 Desain wali kelas melihat absensi siswa

## D. Hasil Implementasi

Untuk memastikan bahwa optimalisasi penjadwalan mata pelajaran berfungsi dengan baik, implementasi melibatkan pengetesan dan uji coba untuk menghindari kesalahan yang mungkin terjadi.

### 1. Login

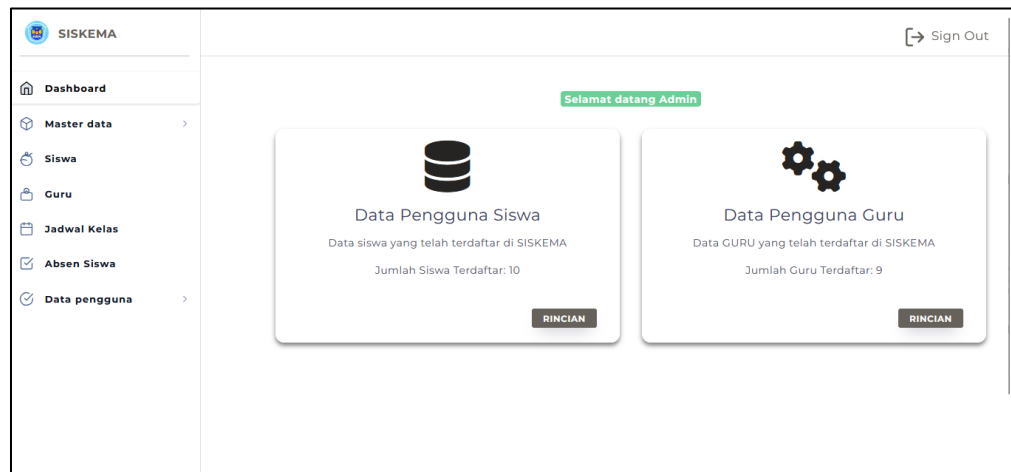
Fitur login adalah pintu masuk utama ke dalam sistem, di mana pengguna (admin, guru, atau siswa) harus memasukkan email dan password mereka untuk mengakses dashboard dan fitur lainnya. Proses login memastikan bahwa hanya pengguna yang terdaftar yang dapat mengakses informasi dan fungsi sistem. Tampilan halaman login dapat dilihat pada Gambar 4.69.



Gambar 4. 69 Tampilan Login

### 2. Dashboard Admin

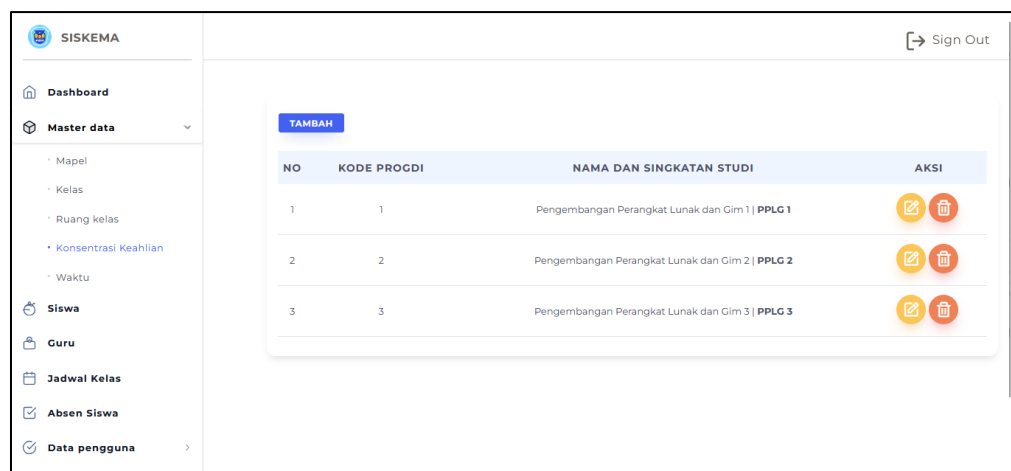
Dashboard Admin memiliki fitur yang memungkinkan admin untuk melihat ringkasan data siswa dan guru. Tampilan dashboard admin dapat dilihat pada Gambar 4.70.



Gambar 4. 70 Tampilan Dashboard Admin

### 3. Master Data

Menu Master Data memungkinkan admin untuk mengelola data utama yang digunakan dalam sistem. Ini termasuk data tentang mapel, kelas, ruang kelas, konsentrasi keahlian dan waktu. Tampilan master data dapat dilihat pada Gambar 4.71.



Gambar 4. 71 Tampilan Master Data

#### 4. Menu Data Siswa

Menu data siswa digunakan untuk mengelola informasi tentang siswa, termasuk profil, konsentrasi keahlian dan kelas. Tampilan menu data guru dapat dilihat pada Gambar 4.72.

NO	NIS	NISN	SISWA	JURUSAN	KELAS	FOTO	AKSI
1	5904	0094576235	Alvia Nav'A Aisyah	Pengembangan Perangkat Lunak dan Gim 1	X	1722680209-Siswi.png	[Edit] [Delete]
2	5905	0091773634	Amalia Ramadhany	Pengembangan Perangkat Lunak dan Gim 1	X	1722680760-Siswi.png	[Edit] [Delete]
3	5906	0091176659	Anindita Revi Aulia	Pengembangan Perangkat Lunak dan Gim 1	X	1722680832-Siswi.png	[Edit] [Delete]
4	5907	0095159101	Aufa Muhammad Yasin	Pengembangan Perangkat Lunak dan Gim 1	X	1722680859-siswa.jpg	[Edit] [Delete]

Gambar 4. 72 Tampilan Data Siswa

#### 5. Menu Data Guru

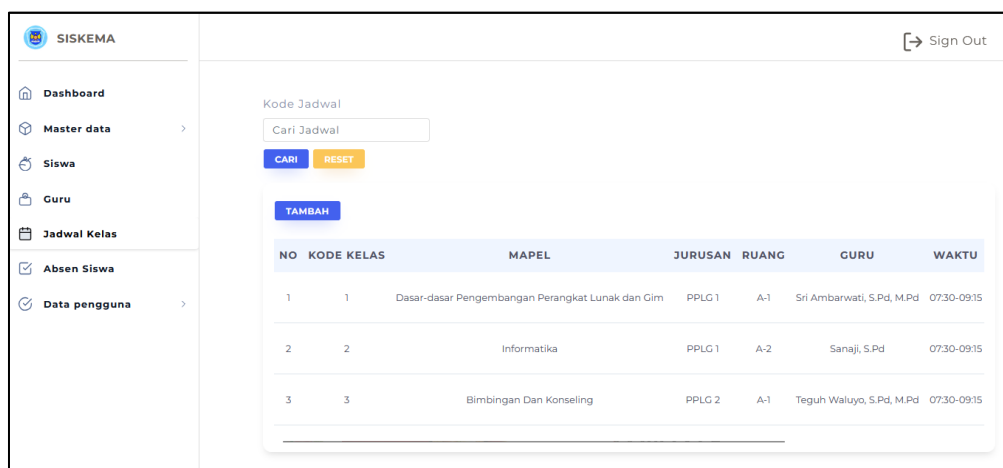
Menu data guru digunakan untuk mengelola informasi tentang guru, termasuk profil dan wali kelas. Tampilan menu data guru dapat dilihat pada Gambar 4.73.

NO	NUPTK	GURU	WALI KELAS	JABATAN FUNGSIONAL	AKSI
1	197508212008012005	Sri Ambarwati, S.Pd, M.Pd	Pengembangan Perangkat Lunak dan Gim 1   X	Guru	[Edit] [Delete]
2	196809022008011007	Sanaji, S.Pd	Pengembangan Perangkat Lunak dan Gim 2   X	Guru	[Edit] [Delete]
3	197007152008011008	Teguh Waluyo, S.Pd, M.Pd	Pengembangan Perangkat Lunak dan Gim 2   X	Guru	[Edit] [Delete]
4	196906062008011020	Masduki, S.Pd	Pengembangan Perangkat Lunak dan Gim 1   X	Guru	[Edit] [Delete]

Gambar 4. 73 Tampilan Data Guru

## 6. Jadwal Kelas

Menu jadwal kelas memungkinkan admin untuk mengatur dan melihat jadwal pelajaran untuk setiap kelas. Ini memastikan tidak ada konflik waktu dan semua mata pelajaran diatur dengan baik. Tampilan jadwal kelas dapat dilihat pada Gambar 4.74.

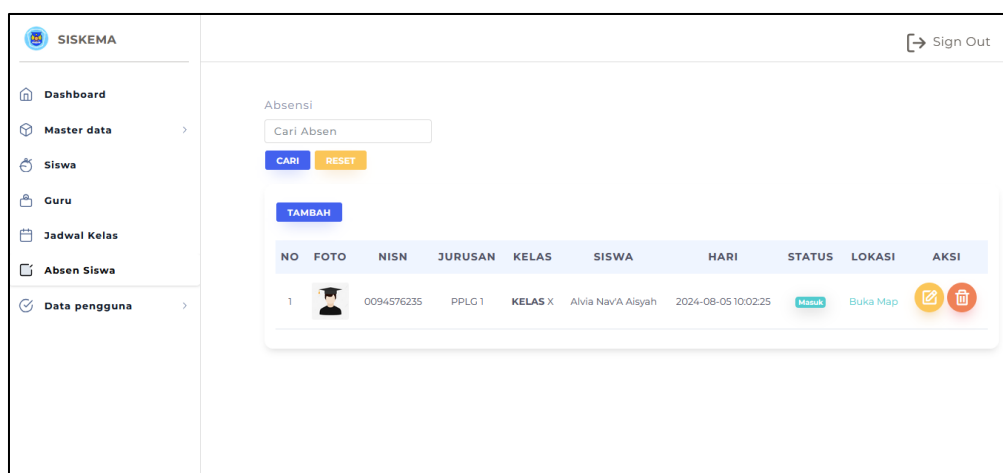





NO	KODE KELAS	MAPEL	JURUSAN	RUANG	GURU	WAKTU
1	1	Dasar-dasar Pengembangan Perangkat Lunak dan Gim	PPLG 1	A-1	Sri Ambarwati, S.Pd, M.Pd	0730-0915
2	2	Informatika	PPLG 1	A-2	Sanaji, S.Pd	0730-0915
3	3	Bimbingan Dan Konseling	PPLG 2	A-1	Teguh Waluyo, S.Pd, M.Pd	0730-0915

Gambar 4. 74 Tampilan Jadwal Kelas

## 7. Absensi Siswa

Menu absensi siswa memungkinkan admin untuk mengelola kehadiran siswa. Data absensi dapat diinput, dilihat, dan diunduh untuk keperluan administrasi. Tampilan absensi siswa dapat dilihat pada Gambar 4.75.

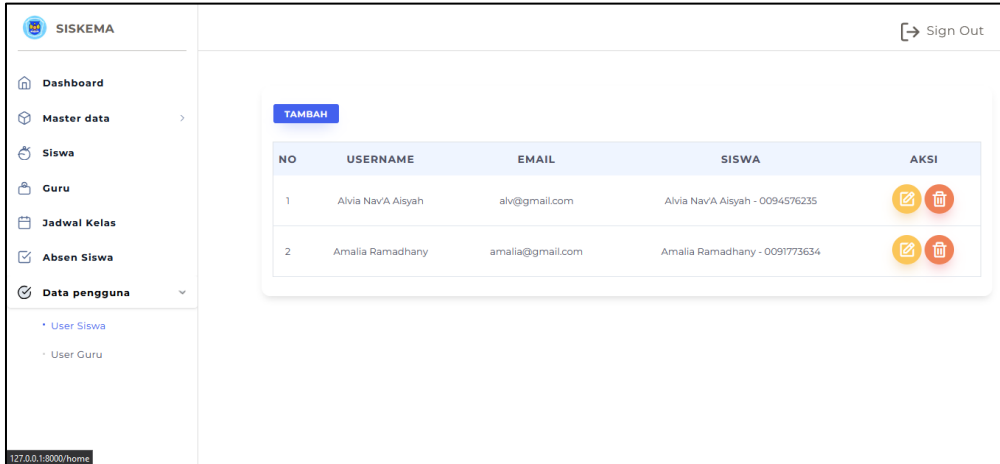






NO	FOTO	NISN	JURUSAN	KELAS	SISWA	HARI	STATUS	LOKASI	AKSI
1		0094576235	PPLG 1	KELAS X	Alvia Nav'A Aisyah	2024-08-05 10:02:25	Masuk	Buka Map	 

Gambar 4. 75 Tampilan Absen Siswa

## 8. User Data

Bagian User Data digunakan untuk mengelola data pengguna sistem, termasuk guru dan siswa. Fitur ini memastikan bahwa hanya pengguna yang sah yang memiliki akses ke sistem. Tampilan user data dapat dilihat pada Gambar 4.76.

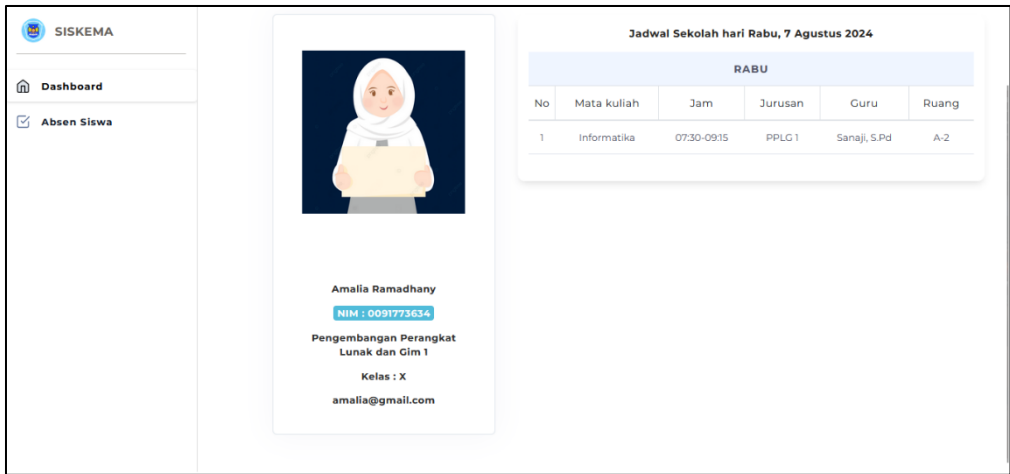


NO	USERNAME	EMAIL	SISWA	AKSI
1	Alvia Nav'A Aisyah	alv@gmail.com	Alvia Nav'A Aisyah - 0094576235	 
2	Amalia Ramadhany	amalia@gmail.com	Amalia Ramadhany - 0091773634	 

Gambar 4. 76 Tampilan Data Pengguna

## 9. Dashboard Siswa

Dashboard siswa memberikan tampilan khusus untuk siswa, di mana siswa dapat melihat profil, hari, tanggal, dan jadwal kelas. Tampilan dashboard siswa dapat dilihat pada Gambar 4.77.

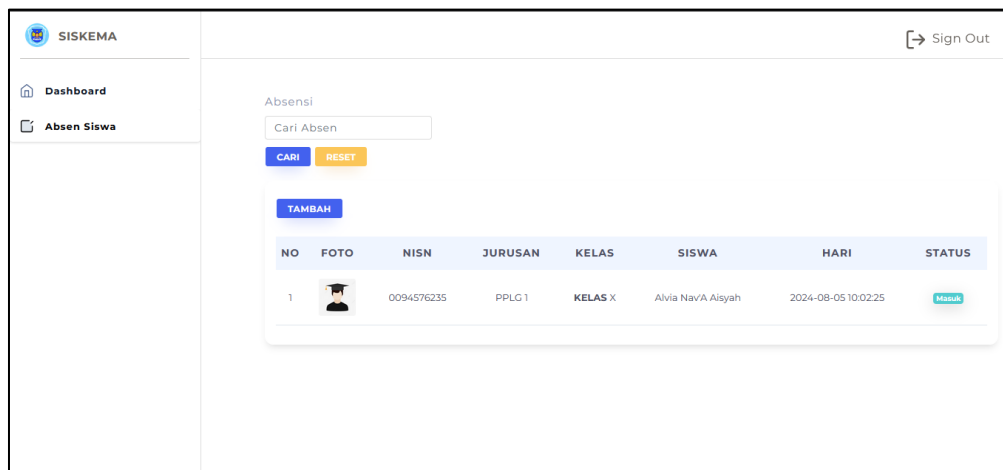


Jadwal Sekolah hari Rabu, 7 Agustus 2024					
RABU					
No	Mata kuliah	Jam	Jurusan	Guru	Ruang
1	Informatika	07:30-09:15	PPLG 1	Sanaji, S.Pd	A-2

Gambar 4. 77 Tampilan Dashboard Siswa

## 10. Absensi Siswa

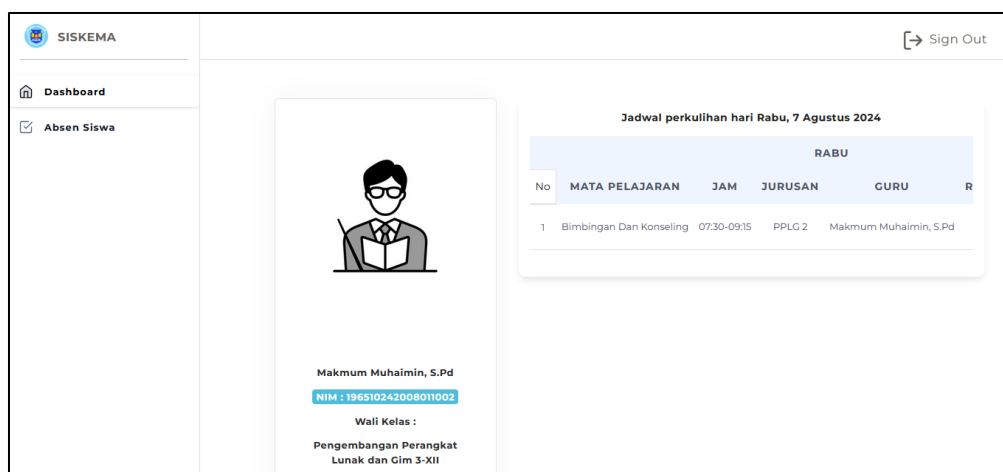
Menu absensi siswa memungkinkan siswa untuk mengelola data kehadiran setiap harinya. Hal ini membantu siswa untuk memantau kehadiran mereka sendiri. Tampilan absensi siswa dapat dilihat pada Gambar 4.78.



Gambar 4. 78 Tampilan Absen Siswa

## 11. Dashboard Guru

Dashboard guru memberikan tampilan khusus untuk guru, di mana guru dapat melihat jadwal mengajar setiap harinya dilengkapi dengan keterangan dan export. Tampilan ini dapat dilihat pada Gambar 4.79.

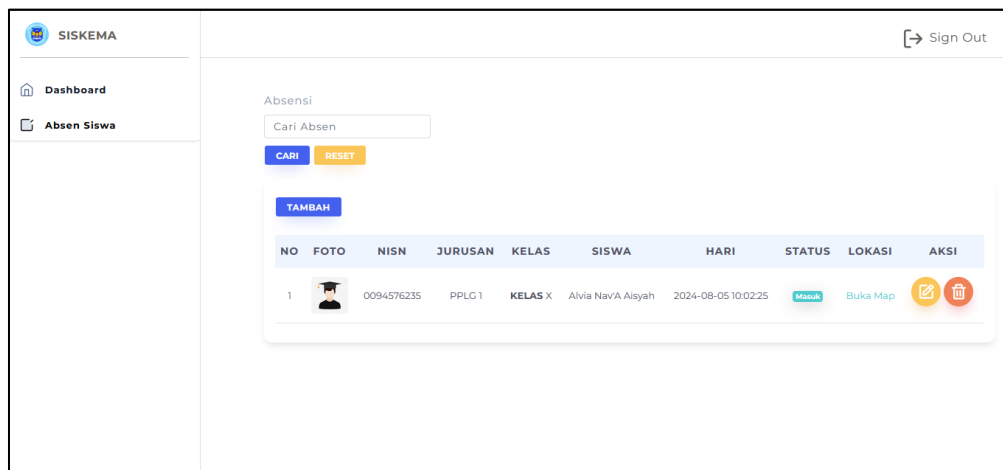


Gambar 4. 79 Tampilan Dashboard Guru



## 12. Melihat Absensi Siswa Sebagai Wali Kelas

Wali kelas memiliki fitur khusus untuk melihat absensi siswa di kelas mereka. Ini membantu wali kelas untuk memantau kehadiran dan memberikan perhatian lebih kepada siswa yang sering melakukan absen. Tampilan ini dapat dilihat pada Gambar 4.80.



Gambar 4. 80 Tampilan Wali Kelas Melihat Absen Siswa

## E. Pengujian Sistem

Pengujian dilakukan untuk memastikan bahwa kinerja aplikasi sudah berjalan dengan baik atau belum. Dalam penelitian ini, jenis pengujian yang digunakan adalah Blackbox Testing dan User Acceptance Testing (UAT).

### 1. Whitebox Testing

White Box Testing merupakan metode pengujian perangkat lunak di mana penguji memeriksa apakah ada kesalahan dalam kode yang dibuat serta mengevaluasi struktur internal dari aplikasi yang diuji. Hasil dari pengujian white box ini dapat dilihat pada Tabel 4.1.

Tabel 4. 2 Whitebox Testing

Node	Code	Pengertian
1.	<pre>public function optimizeSchedule(\$id) { \$timeSlots = WaktuModel::pluck('jam')- &gt;toArray(); \$rooms = RuangModel::pluck('nama', 'id')- &gt;toArray(); \$dosen = DosenModel::pluck('nama', 'id')- &gt;toArray(); \$classses = DaftarkelasModel::all();</pre>	<p>Menginisialisasi variabel yang diperlukan untuk penjadwalan.</p>
2.	<pre>public function generateSchedule(\$classses, \$timeSlots, \$rooms, \$dosen) { \$schedule = [];  foreach (\$classses as \$class) { \$scheduled = false; \$day = \$class-&gt;hari;  \$availableRooms = array_filter(\$rooms, function (\$roomId) use (\$class) { return \$roomId == \$class-&gt;ruang_id; }, ARRAY_FILTER_USE_KEY); foreach (\$timeSlots as \$timeSlot) { foreach (\$availableRooms as \$roomId =&gt; \$roomName) { if (\$this-&gt;isSlotAvailable(\$schedule, \$day, \$timeSlot, \$roomName, \$class-&gt;dosen_id)) { //script 5</pre>	<p>Memeriksa apakah ada kelas yang belum dijadwalkan. Dan Mengelola loop yang mengiterasi setiap kelas yang dijadwalkan</p>

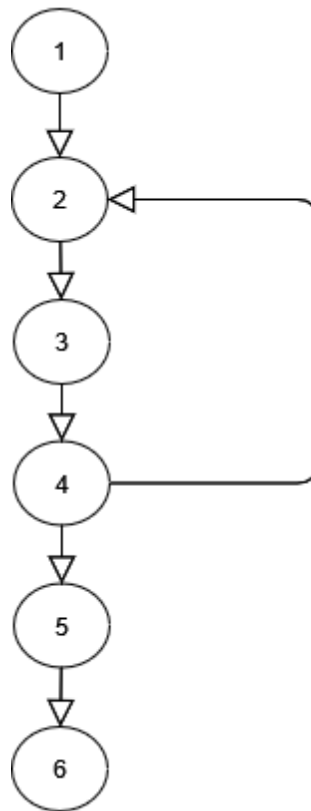
Node	Code	Pengertian
	}	
3.	<pre>private function isSlotAvailable(\$schedule, \$day, \$timeSlot, \$room, \$dosen) { foreach (\$schedule as \$entry) { if (\$entry['hari'] == \$day &amp;&amp; \$entry['waktu'] == \$timeSlot) { if (\$entry['ruang'] == \$room    \$entry['dosen_id'] == \$dosen) { return false; } } } return true; }</pre>	Memeriksa ketersediaan dari kondisi yang telah diterapkan..
4.	<pre>\$schedule[] = [ 'kode_kelas' =&gt; \$class-&gt;kode_kelas, 'makul_id' =&gt; \$class-&gt;makul_id, 'progdi_id' =&gt; \$class-&gt;progdi_id, 'ruang' =&gt; \$roomName, 'ruang_id' =&gt; \$roomId, 'dosen_id' =&gt; \$class-&gt;dosen_id, 'dosen_name' =&gt; \$dosen[\$class-&gt;dosen_id], 'waktu' =&gt; \$timeSlot, 'hari' =&gt; \$day, 'semester' =&gt; \$class-&gt;semester, ];</pre>	Menampilkan jadwal yang sudah di optimalkan dengan memasukan data waktu yang tersedia ke dalam kelas konflik.

Node	Code	Pengertian
5.	<pre> public function saveChanges(Request \$request)     {     \$changes = json_decode(\$request-     &gt;input('changes'), true);      if (!is_array(\$changes)) {         return redirect()-&gt;back()-&gt;withErrors(['error'     =&gt; 'Invalid data format']);     }      \$timeSlots = WaktuModel::pluck('jam')-     &gt;toArray();     \$rooms = RuangModel::pluck('nama', 'id')-     &gt;toArray();     \$dosen = DosenModel::pluck('nama', 'id')-     &gt;toArray();     \$classes = DaftarkelasModel::all();      \$schedule = \$this-&gt;generateSchedule(\$classes,     \$timeSlots, \$rooms, \$dosen);      DaftarkelasModel::truncate();     foreach (\$schedule as \$change) {         DaftarkelasModel::create([             'uid' =&gt; Str::uuid(),             'kode_kelas' =&gt; \$change['kode_kelas'],             'progdi_id' =&gt; \$change['progdi_id'],             'makul_id' =&gt; \$change['makul_id'],             'dosen_id' =&gt; \$change['dosen_id'],             'ruang_id' =&gt; \$change['ruang_id'],             'semester' =&gt; \$change['semester'], </pre>	Menyimpan data yang sudah di optimalkan ke dalam database

Node	Code	Pengertian
	<pre>'hari' =&gt; \$change['hari'], 'start' =&gt; \$change['waktu'], ]); }  //script 7 }</pre>	
6.	<pre>return redirect('/daftar-kelas')-&gt;with('success', 'Data Berhasil Diubah');</pre>	Mengakhiri proses optimalisasi penjadwalan.

- a. Membuat Flowgraph dari program.

Berikut adalah Flowgraph dari kode penilaian diatas dapat dilihat pada Gambar 4.81.



Gambar 4. 81 Flowgraph

b. Menghitung Kompleksitas Siklomatik

Kompleksitas siklomatik adalah metrik yang digunakan untuk menilai kompleksitas struktural sebuah program. Metrik ini dihitung dengan rumus:

$$V(G) = E - N + 2P$$

dimana:

E adalah jumlah edge (garis alur) dalam flow graph,

N adalah jumlah node (titik keputusan) dalam flow graph.

P Adalah jumlah komponen yang terhubung ( untuk grafik yang terhubung tunggal, P=1)

Perhitungan kompleksitas siklomatik berikut ini:

- Jumlah Node (N): 6
- Jumlah Edge (E): 6

Dengan menggunakan rumus di atas, kompleksitas siklomatik dapat dihitung sebagai berikut:

$$V(G) = E - N + 2P = 6 - 6 + 2 \times 1$$

$$V(G) = 2$$

Basis path dihasilkan dari jalur independent secara linier menunjukkan 2 jalur independen dalam flow graph. Jalur-jalur independen tersebut adalah:

a. Path 1 : 1 → 2 → 3 → 4 → 5 → 6

b. Path 2 : 1 → 2 → 3 → 4 → 2 → 3 → 4 → 5

Dengan perhitungan ini, kita dapat melengkapi analisis white-box testing dengan menghitung cosine similarity. Ini membantu dalam menganalisis jalur eksekusi dan menentukan kesamaan antara jalur independen dalam program.

## 2. Blackbox Testing

Pengujian ini dilakukan oleh tiga dosen Informatika. Proses pengujian Blackbox dibagi menjadi tiga tahap, yaitu tujuan, skenario, dan hasil yang didapatkan. Setelah merancang pengujian Blackbox, rancangan tersebut kemudian dicetak menjadi lembar kuesioner dan dibagikan kepada ketiga dosen Informatika tersebut untuk melakukan pengujian. Hasil dari pengujian ini akan digunakan untuk mengevaluasi dan memastikan bahwa aplikasi telah berfungsi sesuai dengan spesifikasi yang diharapkan dan siap digunakan di lingkungan operasional.

### a. Hasil Perhitungan Pengujian Black Box

Hasil Perhitunagn Pengujian Black Box dapat dilihat pada tabel 4.2

Tabel 4. 3 Blackbox Testing

No	Nama Pengujian	Tujuan	Skenario	Hasil yang didapatkan	Hasil					
					Diterima			Ditolak		
1.	Menampilkan halaman login	User dapat melihat form login dan melakukan login	Ditampilkan saat user mengakses url	Sistem akan menampilkan halaman login	√	√	√			
2.	Menampilkan halaman dashboard admin	Admin dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard admin	√	√	√			
3.	Menampilkan halaman master data	Admin dapat melihat halaman master data	Admin menekan tombol navigasi dan masuk ke halamn master data	Sistem akan menampilkan halaman master data	√	√	√			
4.	Menambahka n data di master data	Admin dapat menambahkan data di master data	Admin menekan tombol tambah dan menambahka n data baru	Data di dalam master data bertambah	√	√	√			
5.	Mengedit data di master data	Admin dapat mengedit data di master data	Admin menekan tombol edit	Data di dalam master data berubah	√	√	√			



			dan memasukan data baru							
6.	Menghapus data di master data	Admin dapat menghapus data di dalam master data	Admin menekan tombol delete	Data di dalam master data terhapus sesuai data yang dipilih	√	√	√			
7.	Menampilkan halaman siswa	Admin dapat melihat halaman siswa	Admin menekan tombol navigasi dan masuk ke halamn siswa	Sistem akan menampilkan halaman siswa	√	√	√			
8.	Menambahka n data siswa	Admin dapat menambahkan data siswa	Admin menakan tombol tambah dan menambahka n data baru	Data siswa bertambah	√	√	√			
9.	Mengedit data siswa	Admin dapat mengedit data siswa	Admin menekan tombol edit dan memasukan data baru	Data siswa terubah	√	√	√			
10.	Menghapus data siswa	Admin dapat menghapus data siswa	Admin menekan tombol delete	Data siswa terhapus sesuai data yang dipilih	√	√	√			

11.	Menampilkan halaman guru	Admin dapat melihat halaman guru	Admin menekan tombol navigasi dan masuk ke halamn guru	Sistem akan menampilkan halaman guru	√	√	√			
12.	Menambahka n data guru	Admin dapat menambahkan data guru	Admin menakan tombol tambah dan menambahka n data baru	Data guru bertambah	√	√	√			
13.	Mengedit data guru	Admin dapat mengedit data guru	Admin menekan tombol edit dan memasukan data baru	Data guru berubah	√	√	√			
14.	Menghapus data guru	Admin dapat menghapus data guru	Admin menekan tombol delete	Data guru terhapus sesuai data yang dipilih	√	√	√			
15.	Menampilkan menu jadwal kelas	Admin dapat melihat halaman jadwal kelas	Admin menekan tombol navigasi dan masuk ke halamn jadwal kelas	Sistem akan menampilkan halaman jadwal kelas	√	√	√			

16.	Menambahkan data jadwal kelas	Admin dapat menambahkan data jadwal kelas	Admin menekan tombol tambah dan menambahkan data baru	Data jadwal kelas bertambah	√	√	√			
17.	Mengedit data jadwal kelas	Admin dapat mengedit data jadwal kelas	Admin menekan tombol edit dan memasukkan data baru	Data jadwal kelas berubah	√	√	√			
18.	Rekomendasi Jadwal Kelas	Admin dapat melihat rekomendasi jadwal kelas	Admin menekan tombol rekomendasi	Sistem akan menampilkan data jadwal kelas yang di rekomendasi	√	√	√			
19.	Menghapus data jadwal kelas	Admin dapat menghapus data jadwal kelas	Admin menekan tombol delete	Data jadwal kelas terhapus sesuai data yang dipilih	√	√	√			
20.	Menampilkan menu absensi siswa	Admin dapat melihat halaman absensi siswa	Admin menekan tombol navigasi dan masuk ke halaman absensis siswa	Sistem akan menampilkan halaman absensi siswa	√	√	√			

21.	Menambahkan data absensi siswa	Admin dapat menambahkan data absensi siswa	Admin menekan tombol tambah dan menambahkan data baru	Data absensi siswa bertambah	√	√	√			
22.	Mengedit data absensi siswa	Admin dapat mengedit data absensi siswa	Admin menekan tombol edit dan memasukan data baru	Data absensi siswa terubah	√	√	√			
23.	Menghapus data absensi siswa	Admin dapat menghapus absensi siswa	Admin menekan tombol delete	Data absensi siswa terhapus sesuai data yang dipilih	√	√	√			
24.	Menampilkan menu user data	Admin dapat melihat halaman user data	Admin menekan tombol navigasi dan masuk ke halaman user data	Sistem akan menampilkan halaman user data	√	√	√			
25.	Menambahkan data user data	Admin dapat menambahkan data user data	Admin menekan tombol tambah dan menambahkan data baru	Data user data bertambah	√	√	√			

26.	Mengedit data user data	Admin dapat mengedit data user data	Admin menekan tombol edit dan memasukkan data baru	Data user data berubah	√	√	√			
27.	Menghapus user data	Admin dapat menghapus user data	Admin menekan tombol delete	Data user data terhapus sesuai data yang dipilih	√	√	√			
Guru										
28.	Menampilkan halaman dashboard guru	Guru dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard guru	√	√	√			
29.	Menambah keterangan	Guru dapat menambah keterangan saat mengajar	Guru menekan tombol keterangan dan menambahkan data baru	Data keterangan bertambah	√	√	√			
30.	Export kegiatan mengajar	Guru dapat mencetak kegiatan mengajar	Guru menekan tombol export	Sistem mencetak sebuah file dengan data jadwal,	√	√	√			

				keterangan dan absensi						
31.	Menampilkan absen siswa	Guru dapat melihat absensi dari siswa	Admin menekan tombol navigasi dan masuk ke halaman user data	Sistem akan menampilkan halaman absensi siswa	√	√	√			
Siswa										
32.	Menampilkan halaman dashboard siswa	Siswa dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard siswa	√	√	√			
33.	Menampilkan menu absensi siswa	Admin dapat melihat halaman absensi siswa	Admin menekan tombol navigasi dan masuk ke halaman absensi siswa	Sistem akan menampilkan halaman absensi siswa	√	√	√			
34.	Menambahkan data absensi siswa	Admin dapat menambahkan data absensi siswa	Admin menekan tombol tambah dan menambahkan data baru	Data absensi siswa bertambah	√	√	√			

b. Kesimpulan

Berdasarkan pengujian black box yang dilakukan pada aplikasi "Tajwid Plus," sebanyak 34 pengujian telah dilakukan oleh 3 responden. Berikut adalah hasil pengujian tersebut:

1. Pengujian Pertama

$$\text{Tercapai } \frac{34}{34} \times 100\% = 100\%$$

$$\text{Gagal } \frac{0}{34} \times 100\% = 0\%$$

2. Pengujian Kedua

$$\text{Tercapai } \frac{34}{34} \times 100\% = 100\%$$

$$\text{Gagal } \frac{0}{34} \times 100\% = 0\%$$

3. Pengujian Ketiga

$$\text{Tercapai } \frac{34}{34} \times 100\% = 100\%$$

$$\text{Gagal } \frac{0}{34} \times 100\% = 0\%$$

$$\text{Jumlah presentase rata-rata tercapai } \frac{300\%}{3} = 100\%$$

$$\text{Jumlah presentase rata-rata gagal } \frac{0\%}{3} = 0\%$$

Berdasarkan analisis di atas, dari 34 pengujian yang dilakukan oleh 3 responden, hasilnya menunjukkan bahwa tingkat keberhasilan pengujian black box mencapai 100%. Tidak ditemukan kegagalan, sehingga persentase kegagalannya adalah 0%. Kesimpulannya, sistem optimalisasi penjadwalan berhasil berfungsi sesuai dengan yang diharapkan.

3. User Acceptance Testing (UAT)

Pengujian UAT ini dilakukan oleh 5 orang responden yang merupakan guru dan staf kurikulum dari SMK Negeri 5 Kendal. Melalui proses ini, penulis dan pengembang sistem dapat memastikan bahwa sistem tidak

hanya berfungsi secara teknis tetapi juga memenuhi kebutuhan dan ekspektasi pengguna.

a. Hasil Perhitungan Pengujian User Acceptance Testing (UAT)

Berikut merupakan hasil kuisioner pengujian User Acceptance Testing (UAT) yang telah diberikan kepada 5 orang responden. Hasil pengujian UAT dapat dilihat pada Tabel 4.3.

Tabel 4. 4 User Acceptance Testing

Pertanyaan	Hasil Pengujian				
	R1	R2	R3	R4	R5
1.	4	4	4	5	5
2.	5	4	5	5	4
3.	5	4	5	4	4
4.	5	5	5	4	4
5.	5	4	4	4	5
6.	4	4	4	5	5
7.	4	4	5	4	5
8.	5	5	4	5	4
9.	4	4	5	5	4
10.	4	4	5	5	4
Jumlah	45	42	46	46	44
Presentase	90%	84%	92%	92%	88%
Total	$\frac{446}{5} = 89,2\%$				

b. Kesimpulan Hasil Pengujian User Acceptance Testing (UAT)

Berdasarkan hasil pengujian User Acceptance Testing (UAT), evaluasi terhadap persentase untuk setiap pertanyaan yang mencakup kegunaan, kemudahan, dan User Interface (UI) telah dilakukan terhadap 5 responden. Nilai-nilai persentase tersebut



dijumlahkan dan kemudian dihitung rata-ratanya. Nilai rata-rata ini diperoleh dengan menggunakan rumus sebagai berikut:

$$\text{Presentase rata-rata} = \frac{\text{Jumlah total presentase}}{\text{Jumlah reesponden}}$$

$$\text{Presentase rata-rata} = \frac{446}{5} = 89,2\%$$

Dengan daftar kategori sebagai berikut:

0% - 20% = Sangat Kurang

21% - 40% = Kurang

41% - 60% = Cukup

61% - 80% = Baik

81% - 100% = Sangat Baik

Dari perhitungan di atas, diperoleh persentase rata-rata dari tiga aspek sebesar 89,2%. Dengan demikian, dapat disimpulkan bahwa hasil pengujian UAT pada aplikasi ini berada dalam kategori "Sangat Baik".

## F. Pembahasan

Dalam bagian ini, akan dilakukan analisis dan diskusi mengenai hasil pengembangan sistem optimalisasi penjadwalan. Pembahasan ini mencakup evaluasi terhadap temuan-temuan utama, perbandingan dengan hipotesis atau tujuan awal penelitian, serta hubungannya dengan teori yang telah dijelaskan sebelumnya. Selain itu, akan dibahas juga implikasi praktis dari temuan tersebut, keterbatasan yang dihadapi selama penelitian, dan rekomendasi untuk penelitian di masa depan.

### 1. Analisis Kebutuhan

Identifikasi kebutuhan merupakan langkah awal yang sangat penting dalam pengembangan sistem optimasi penjadwalan ini. Pada tahap ini, dilakukan analisis kebutuhan sistem dan kebutuhan fungsional untuk memastikan bahwa sistem dapat beroperasi secara optimal dan memenuhi kebutuhan pengguna dengan baik. Dalam analisis kebutuhan sistem, perangkat keras yang diperlukan meliputi server dengan

prosesor yang kuat, Processor Intel Core i5-8250U, Grafis Card NVIDIA GTX 950, RAM 4 DDR 4 dan penyimpanan SSD untuk memastikan kecepatan akses data yang tinggi. Perangkat lunak yang digunakan termasuk Laravel sebagai framework PHP untuk pengembangan back-end, MySQL sebagai database, serta HTML, CSS, dan JavaScript untuk front-end.

Kebutuhan fungsional yang diidentifikasi mencakup pengelolaan jadwal yang memungkinkan sistem untuk melakukan penjadwalan otomatis berdasarkan ketersediaan ruang, dosen, dan waktu, serta menghindari bentrok jadwal. Sistem juga harus mampu menyimpan dan mengelola data ruang kelas, termasuk kapasitas dan ketersediaan ruang, serta data dosen dan mata pelajaran yang diampu. Selain itu, sistem harus memiliki antarmuka yang intuitif dan mudah digunakan oleh admin mampu mengoptimasi jadwal, serta memastikan keamanan data dengan mencegah akses yang tidak sah. Dengan analisis kebutuhan yang komprehensif ini, pengembangan sistem dapat dilakukan dengan lebih terarah dan sesuai dengan harapan pengguna.

## 2. Desain Sistem

Penggunaan model perancangan seperti Unified Modeling Language (UML) sangat penting untuk desain sistem. Model ini termasuk diagram Use Case, Activity, Sequence, dan Class, yang membantu pengembang menemukan fungsi aplikasi dan alur kerja yang terjadi di dalamnya.

Untuk memahami bagaimana pengguna berinteraksi dengan sistem, diagram skenario penggunaan sangat penting. Selain itu, Activity Diagram menunjukkan urutan aktivitas dan alur kerja yang terlibat dan menjelaskan langkah-langkah proses. Sequence diagram menunjukkan interaksi antar objek dalam urutan waktu tertentu. Ini juga menunjukkan aliran pesan yang terjadi antar objek selama skenario berlangsung. Klasifikasi kelas dan hubungan antar kelas menunjukkan struktur sistem dan arsitektur aplikasi.

Penulis juga membuat wireframe sederhana untuk desain antarmuka pengguna menggunakan UML. Tujuan dari wireframe ini adalah untuk memastikan bahwa tata letak elemen antarmuka pengguna sesuai dengan kebutuhan fungsional dan mudah digunakan oleh pengguna. Wireframe juga memungkinkan visualisasi awal desain antarmuka menjadi lebih jelas, memastikan bahwa setiap elemen antarmuka pengguna ditempatkan secara intuitif, dan mendukung interaksi pengguna yang efektif dengan aplikasi.

Dalam proses desain wireframe, umpan balik dari pengguna atau pemangku kepentingan sangat penting untuk memastikan bahwa desain yang dihasilkan sesuai dengan kebutuhan dan harapan :

- a. Desain login : Mengganti desain halaman login sesuai arahan pembimbing dan penekanan lebih pada label untuk kolom email dan password agar lebih jelas.
- b. Tombol navigasi: Tata letak tombol navigasi tambah ditempatkan di bagian atas dan tombol edit dan delete di berikan kolom di sebelah data yang ingin di rubah atau di hapus.
- c. Keterbacaan dan Aksesibilitas: Memperbesar ukuran teks pada data saat melakukan input,update dan di bagian index untuk meningkatkan keterbacaan.
- d. Fitur : Membuat fitur menambahkan keterangan dan export oleh guru pada jadwal dan membuat menu absensi yang memenuhi kriteria untuk pengecekan lokasi,tanggal dan waktu.

Setelh melakukan desain sistem maka akan dilakukan 3 tahap yaitu sebagai berikut :

- a. Tahap *Prototype*

Tahap ini adalah proses setelah dari *User Design* di membuat prototipe atau wireframe dari sistem yang akan dibangun. Prototipe ini digunakan untuk menunjukkan bagaimana sistem akan berfungsi dan memberikan gambaran awal kepada

pengguna. Tujuan dari tahap ini adalah untuk mengumpulkan umpan balik dari pengguna tentang kebutuhan dan ekspektasi mereka terhadap sistem.

b. Tahap *Refine*

Setelah mendapatkan umpan balik dari pengguna pada tahap prototipe, maka akan memperbaiki dan menyempurnakan sistem berdasarkan masukan tersebut. Tahap ini melibatkan penyesuaian desain, penambahan fitur yang diinginkan, dan perbaikan bug atau masalah yang ditemukan selama penggunaan prototipe.

c. Tahap Test

Setelah sistem yang disempurnakan selesai, sistem diuji secara menyeluruh untuk memastikan bahwa semua fungsi bekerja sesuai dengan kebutuhan dan spesifikasi yang diinginkan. Pengujian ini melibatkan pengguna akhir untuk memastikan bahwa sistem mudah digunakan dan memenuhi tujuan bisnis yang telah ditetapkan. Tetapi jika ditemukan masalah selama pengujian, tim pengembang kembali ke tahap refine untuk memperbaikinya, dan kemudian sistem diuji lagi.

Setelah ketiga langkah tersebut dalam proses desain sistem pada pengembangan RAD bersifat iteratif dan berulang sampai sistem memenuhi semua persyaratan pengguna dan siap untuk dilanjutkan ke tahap konstruksi

3. Kontruksi

Langkah pertama dalam implementasi sistem optimalisasi penjadwalan adalah membangun antarmuka pengguna (UI) berdasarkan desain yang telah dirancang sebelumnya. Setelah model antarmuka selesai dikembangkan, tahap berikutnya adalah implementasi fitur-fitur utama dan algoritma yang akan digunakan dalam proses optimalisasi.

Pada tahap implementasi fitur, beberapa langkah penting yang harus dilakukan meliputi:

- a) Pembuatan modul untuk pengelolaan data jadwal, termasuk kelas, dosen, ruang, dan waktu.
- b) Pengembangan modul untuk deteksi konflik penjadwalan, seperti bentrokan waktu atau ruangan.
- c) Implementasi fitur adalah integrasi modul optimasi yang akan menyesuaikan penjadwalan agar sesuai dengan kriteria yang telah ditetapkan.

Setelah fitur-fitur tersebut diimplementasikan, langkah selanjutnya adalah mengimplementasikan algoritma greedy untuk menghasilkan jadwal yang optimal berdasarkan input data yang tersedia.

Langkah terakhir adalah melakukan pengujian untuk memastikan bahwa sistem penjadwalan bekerja dengan baik. Pengujian ini mencakup pemeriksaan fungsionalitas seluruh fitur serta kemudahan penggunaan antarmuka dalam aplikasi. Setelah pengujian selesai dan hasilnya memuaskan, sistem penjadwalan dapat dianggap siap untuk digunakan oleh pengguna.

#### 4. Algoritma Greedy

Algoritma greedy adalah salah satu metode yang diterapkan dalam sistem optimalisasi penjadwalan ini. Prinsip dasar dari algoritma greedy adalah membuat keputusan terbaik di setiap langkah berdasarkan informasi yang tersedia saat itu, tanpa mempertimbangkan konsekuensi jangka panjang.

Dalam konteks penjadwalan, algoritma ini diterapkan dengan memilih slot waktu yang optimal untuk setiap kelas secara bertahap, berdasarkan ketersediaan ruangan, dosen, dan waktu. Sistem akan menghindari konflik seperti tabrakan hari, jam, ruangan, dan dosen dengan memeriksa ketersediaan slot yang sesuai sebelum menempatkan kelas pada jadwal.

Keunggulan dari pendekatan ini adalah kemampuannya untuk menghasilkan solusi dalam waktu singkat, karena tidak perlu meninjau semua kemungkinan solusi. Namun, karena algoritma greedy tidak mempertimbangkan konsekuensi jangka panjang, hasilnya mungkin bukan yang paling optimal secara keseluruhan, tetapi cukup baik untuk memenuhi kebutuhan praktis pengguna.

Langkah-langkah penerapan algoritma greedy dalam sistem penjadwalan ini meliputi:

- a) Memeriksa setiap kelas yang perlu dijadwalkan.
- b) Memilih slot waktu yang tersedia dan paling optimal berdasarkan kriteria yang telah ditetapkan.
- c) Menghindari konflik penjadwalan dengan memastikan tidak ada tabrakan pada waktu, ruangan, atau dosen.

## 5. Pengujian Aplikasi

Langkah terakhir dalam proses pengembangan adalah melakukan pengujian untuk memastikan kualitas sistem yang telah dikembangkan. Penulis melakukan tiga jenis pengujian: White Box Testing, Black Box Testing, dan User Acceptance Testing (UAT). Hasil pengujian dengan kompleksitas siklomatik  $V(G) = 3$  menunjukkan bahwa struktur kode cukup sederhana dan tidak terlalu kompleks, sehingga memudahkan perbaikan dan pengembangan sistem lebih lanjut. Jika semua jalur independen diuji secara menyeluruh, pengujian ini dapat dianggap memuaskan.

Pada Black Box Testing, hasilnya menunjukkan tingkat keberhasilan 100%, dengan kegagalan sebesar 0% dari 3 responden dan 34 pengujian yang telah disertai masukan dari para responden yang tercantum dalam Lampiran. Pengujian User Acceptance Testing (UAT) juga memberikan hasil yang sangat positif, dengan tingkat keberhasilan mencapai 89.2% dari 5 responden yang menjawab 10 pertanyaan.

Terakhir, pengujian lapangan menunjukkan bahwa sistem mampu memberikan rekomendasi jadwal Pelajaran yang optimal dengan baik dan memenuhi kondisi yang diberikan.

## **BAB V**

### **PENUTUP**

#### **A. Simpulan**

Penelitian ini telah menunjukkan bahwa sistem optimasi penjadwalan yang dikembangkan berhasil memenuhi tujuan utamanya, yaitu memberikan rekomendasi jadwal yang optimal sesuai dengan kondisi yang diberikan. Sistem ini telah diuji melalui tiga tahapan pengujian, yaitu White Box Testing, Black Box Testing, dan User Acceptance Testing (UAT), yang semuanya memberikan hasil yang memuaskan.

Hasil pengujian White Box Testing dengan kompleksitas siklomatik  $V(G) = 2$  menunjukkan bahwa struktur kode sistem cukup sederhana, yang memudahkan proses perbaikan dan pengembangan lebih lanjut. Pengujian Black Box Testing menghasilkan tingkat keberhasilan 100% tanpa adanya kegagalan dari 34 pengujian yang dilakukan oleh 3 responden. Sementara itu, pengujian User Acceptance Testing (UAT) juga memberikan hasil yang sangat baik, dengan tingkat keberhasilan 89.2% dari 5 responden yang menjawab 10 pertanyaan.

Implementasi algoritma Greedy dalam proses optimasi penjadwalan berperan penting dalam pencapaian hasil yang optimal. Algoritma Greedy bekerja dengan memilih solusi terbaik yang tersedia pada setiap langkah, dengan harapan bahwa pilihan lokal terbaik pada akhirnya akan menghasilkan solusi global yang optimal. Dalam konteks penjadwalan, algoritma ini digunakan untuk secara bertahap mengisi slot waktu yang tersedia dengan kelas, guru, dan ruangan yang paling sesuai berdasarkan kriteria tertentu, seperti menghindari bentrokan jadwal atau meminimalkan jarak waktu antar kelas. Proses ini memungkinkan tercapainya jadwal yang lebih efisien dan efektif dengan meminimalkan konflik waktu serta memaksimalkan penggunaan sumber daya yang ada.



Namun, perlu diperhatikan bahwa algoritma Greedy juga memiliki kelebihan dan kekurangan. Kelebihannya terletak pada kesederhanaan dan kecepatan dalam menghasilkan solusi. Algoritma ini mampu memberikan solusi yang cukup baik dalam waktu singkat, yang sangat berguna dalam situasi di mana keputusan cepat dibutuhkan. Selain itu, karena algoritma ini bekerja secara iteratif, proses penjadwalan dapat disesuaikan dengan mudah jika ada perubahan data atau kondisi.

Di sisi lain, kekurangan utama dari algoritma Greedy adalah kecenderungannya untuk menghasilkan solusi suboptimal dalam beberapa kasus. Karena algoritma ini hanya mempertimbangkan solusi terbaik pada setiap langkah tanpa memikirkan dampak jangka panjang, ada kemungkinan bahwa solusi yang dihasilkan bukanlah solusi global yang paling optimal. Hal ini terutama terjadi dalam situasi di mana pilihan lokal yang tampak optimal pada awalnya ternyata menghalangi tercapainya solusi yang lebih baik di kemudian hari.

Secara keseluruhan, penggunaan algoritma Greedy dalam sistem optimasi penjadwalan ini terbukti efektif untuk mencapai tujuan yang diinginkan, meskipun ada keterbatasan yang perlu diperhatikan. Dengan penyesuaian lebih lanjut dan pertimbangan tambahan, hasil yang lebih optimal dan efisien dapat dicapai di masa mendatang.

## **B. Saran**

Berdasarkan penelitian yang telah dilakukan, penulis memberikan beberapa saran yang dapat dipertimbangkan oleh peneliti selanjutnya dalam pengembangan dan optimasi sistem penjadwalan :

- 1) Menyediakan visualisasi yang lebih baik agar dapat mempermudah pengguna dalam memahami dan mengevaluasi jadwal yang dihasilkan. Hal ini juga memungkinkan pengguna untuk lebih mudah melakukan penyesuaian jika diperlukan.

- 2) Algoritma yang digunakan dalam optimasi penjadwalan dapat ditingkatkan fleksibilitasnya dengan memperkenalkan parameter penyesuaian yang dapat diatur oleh pengguna.
- 3) Peneliti selanjutnya dapat mempertimbangkan penggunaan algoritma lain, seperti algoritma Genetika atau Artificial Bee Colony, untuk meningkatkan efisiensi dan hasil optimal.

## DAFTAR PUSTAKA

- [1] B. W. Aulia, M. Rizki, P. Prindiyana, and S. Surgana, "Peran Krusial Jaringan Komputer dan Basis Data dalam Era Digital," JUSTINFO | Jurnal Sistem Informasi dan Teknologi Informasi, vol. 1, no. 1, pp. 9–20, Dec. 2023, doi: 10.33197/justinfo.vol1.iss1.2023.1253.
- [2] DESI SEPTRIANI, "Hambatan Dalam Menerapkan Sistem Informasi Manajemen Dalam Dunia Pendidikan Dan Cara Mengatasinya," 2019.
- [3] F. Sahrul, S. Kom, M. Eng, M. A. Safi'ie, S. Si, and O. Decroly, "Implementasi Sistem Informasi Akademik Berbasis Web Menggunakan Framework Laravel," 2016.
- [4] B. Willson, "Penerapan Algoritma Greedy pada Perancangan Jadwal Kuliah yang Optimal dengan Memperhatikan Preferensi Dosen," 2023. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy->
- [5] M. Fitriawati, M. Dwi Rahmatya, S. Mauluddin, and A. Nursikuwagus, "Analisis Dan Pemodelan Prototipe Sistem Informasi Jadwal Kuliah Dengan Algoritma ABC (Artificial Bee Colony)," 2018.
- [6] A. S. U. M. Restie Maya Puspita, "Pengembangan Aplikasi Penjadwalan Kegiatan Pelatihan Teknologi Informasi Dan Komunikasi Dengan Algoritma Genetika," 2016.
- [7] M. Muhardeny, M. Haviz Irfani, and J. Alie, "Penjadwalan Mata Pelajaran Menggunakan Algoritma Particle Swarm Optimization (PSO) Pada SMPIT Mufidatul Ilmi," 2023.
- [8] A. Roihan, K. Nasution, and Mhd. Z. Siambaton, "Implementasi Algoritma Greedy Kombinasi dengan Perulangan pada Aplikasi Penjadwalan Praktikum," sudo Jurnal Teknik Informatika, vol. 1, no. 2, pp. 42–50, Jun. 2022, doi: 10.56211/sudo.v1i2.8.

- [9] E. S. Laela, W. Gata, and J. J. Purnama, "Optimalisasi Algoritma Greedy dalam Penyusunan Jadwal Pelajaran pada SMK Nurul Islam Cianjur," vol. 12, no. 7, 2022, doi: 10.36418/syntax.
- [10] M. Zen, R. B. Utomo, and N. Hamdi, "Perancangan Sistem Informasi Penjadwalan Mata Pelajaran Pada Smk Negeri 9 Medan," 2023.
- [11] H. Agustin, "Sistem Informasi Manajemen Menurut Prespektif Islam," *Jurnal Tabarru' : Islamic Banking and Finance*, vol. 1, no. 1, 2018.
- [12] S. Informasi Akademik Berbasis Web Di SMP Negeri, K. Gaung Anak Serka, and M. Rizki Alpiandi, "Sistem Informasi Akademik Berbasis Web Di SMP Negeri 2 Kecamatan Gaung Anak Serka," 2016.
- [13] A. R. Nugraha and G. Pramukasari, "urnal Manajemen Informatika Sistem Informasi Akademik Sekolah Berbasis Web Di Sekolah Menengah Pertama Negeri 11 Tasikmalaya," *JUMIKA*, vol. 4, no. 2, 2017.
- [14] Imam Saepul Azmi and Ina Najiyah, "Sistem Informasi Penjadwalan Mata Pelajaran Menggunakan Algoritma Artificial Bee Colony Berbasis Web," *Jurnal Ilmiah Teknik Informatika dan Komunikasi*, vol. 3, no. 3, pp. 35–48, Sep. 2023, doi: 10.55606/juitik.v3i3.612.
- [15] F. Mone and J. E. Simarmata, "Aplikasi Algoritma Genetika Dalam Penjadwalan Mata Kuliah," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 15, no. 4, pp. 615–628, Dec. 2021, doi: 10.30598/barekengvol15iss4pp615-628.
- [16] L. Santoso and J. Amanullah, "Pengembangan Sistem Informasi Akademik Berbasis Website Menggunakan Metode Rapid Application Development (RAD)," vol. 15, no. 2, pp. 250–259, 2022, [Online]. Available: <http://journal.stekom.ac.id/index.php/elkom>page250
- [17] N. Hidayat and K. Hati, "Penerapan Metode Rapid Application Development (RAD) dalam Rancang Bangun Sistem Informasi Rapor Online (SIRALINE)."

- [18] Z. Hakim and P. Meilina, "Sistem Informasi Akademik Berbasis Webiste (Studi Kasus: Smpit Avicenna)," 2022. [Online]. Available: <https://jurnal.umj.ac.id/index.php/just-it/index>
- [19] A. Rahman and D. Redaksi, "Informatika Dan Teknologi (Intech) Rapid Application Development Sistem Pembelajaran Daring Berbasis Android," JURNAL INTECH, vol. 1, no. 2, pp. 20–25, 2020.
- [20] V. Marudut and M. Siregar, "Sistem Informasi Pendataan Logistik Aktiva Tetap PT. Bank Central Asia, tbk Kantor Cabang Pematangsiantar," 2018.
- [21] A. B. Putra and S. Nita, "Perancangan dan Pembangunan Sistem Informasi E-Learning Berbasis Web (Studi Kasus Pada Madrasah Aliyah Kare Madiun)," 2019.
- [22] J. Firnando, B. Franko, S. Pratama Tanzil, N. Wilyanto, H. Christianto Tan, and E. M. Hartati Kom, "Pembuatan Website Menggunakan Visual Studio Code di SMA Xaverius 3 Palembang," 2023.
- [23] E. Ferdiana Sari, "Penerapan Github Sebagai Media E-Learning Untuk Mengetahui Keefektifan Kolaborasi Project Pada Mata Pelajaran Pemrograman Web Dan Perangkat Bergerak Di SMK Negeri 2 Surabaya," 2021.
- [24] T. Hidayat, H. D. Putri, and A. Uk, "Pengujian Portal Mahasiswa pada Sistem Informasi Akademik (SINA) menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis CORE View metadata, citation and similar papers at core," JUTIS, vol. 7, no. 1, 2019, [Online]. Available: [www.ccsenet.org/cis](http://www.ccsenet.org/cis)
- [25] E. L. Hady, K. Haryono, and N. W. Rahayu, "User Acceptance Testing (UAT) pada Purwarupa Sistem Tabungan Santri (Studi Kasus: Pondok Pesantren Al-Mawaddah) User Acceptance Testing (UAT) of the Prototype of Students' Savings Information System (Case Study: Al-Mawaddah Islamic Boarding School)," 2020.

## LAMPIRAN

### d. Black Box Testing

**KUESIONER PENGUJIAN BLACK BOX PADA "PENERAPAN ALGORITMA GREEDY UNTUK OPTIMALISASI PENJADWALAN MATA PELAJARAN PADA SISTEM INFORMASI AKADEMIK SMK NEGERI 5 KENDAL BERBASIS WEB"**

Nama Penguji : Khoiriya Latifah . S.kom . M.kom  
 Tanggal Pengujian : 8 Agustus 2024.

No	Nama Pengujian	Tujuan	Skenario	Hasil yang didapatkan	Pengujian	
					Diterima	Ditolak
<b>Admin</b>						
1.	Menampilkan halaman login	User dapat melihat form login dan melakukan login	Ditampilkan saat user mengakses url	Sistem akan menampilkan halaman login	✓	
2.	Menampilkan halaman dashboard admin	Admin dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard admin	✓	
3.	Menampilkan halaman master data	Admin dapat melihat halaman master data	Admin menekan tombol navigasi dan masuk ke halaman master data	Sistem akan menampilkan halaman master data	✓	
4.	Menambahkan data di master data	Admin dapat menambahkan data di master data	Admin menekan tombol tambah dan menambahkan data baru	Data di dalam master data bertambah	✓	
5.	Mengedit data di master data	Admin dapat mengedit data di master data	Admin menekan tombol edit dan memasukan data baru	Data di dalam master data terubah	✓	
6.	Menghapus data di master data	Admin dapat menghapus data di dalam master data	Admin menekan tombol delete	Data di dalam master data terhapus sesuai data yang dipilih	✓	

7.	Menampilkan halaman siswa	Admin dapat melihat halaman siswa	Admin menekan tombol navigasi dan masuk ke halamn siswa	Sistem akan menampilkan halaman siswa	✓	
8.	Menambahkan data siswa	Admin dapat menambahkan data siswa	Admin menekan tombol tambah dan menambahkan data baru	Data siswa bertambah	✓	
9.	Mengedit data siswa	Admin dapat mengedit data siswa	Admin menekan tombol edit dan memasukan data baru	Data siswa terubah	✓	
10.	Menghapus data siswa	Admin dapat menghapus data siswa	Admin menekan tombol delete	Data siswa terhapus sesuai data yang dipilih	✓	
11.	Menampilkan halaman guru	Admin dapat melihat halaman guru	Admin menekan tombol navigasi dan masuk ke halamn guru	Sistem akan menampilkan halaman guru	✓	
12.	Menambahkan data guru	Admin dapat menambahkan data guru	Admin menekan tombol tambah dan menambahkan data baru	Data guru bertambah	✓	
13.	Mengedit data guru	Admin dapat mengedit data guru	Admin menekan tombol edit dan memasukan data baru	Data guru terubah	✓	
14.	Menghapus data guru	Admin dapat menghapus data guru	Admin menekan tombol delete	Data guru terhapus sesuai data yang dipilih	✓	
15.	Menampilkan menu jadwal kelas	Admin dapat melihat halaman jadwal kelas	Admin menekan tombol navigasi dan masuk ke	Sistem akan menampilkan halaman jadwal kelas	✓	

			halamn jadwal kelas			
16.	Menambahkan data jadwal kelas	Admin dapat menambahkan data jadwal kelas	Admin menekan tombol tambah dan menambahkan data baru	Data jadwal kelas bertambah	✓	
17.	Mengedit data jadwal kelas	Admin dapat mengedit data jadwal kelas	Admin menekan tombol edit dan memasukan data baru	Data jadwal kelas berubah	✓	
18.	Rekomendasi Jadwal Kelas	Admin dapat melihat rekomendasi jadwal kelas	Admin menekan tombol rekomendasi	Sistem akan menampilkan data jadwl kelas yang di rekomendasi	✓	
19.	Menghapus data jadwal kelas	Admin dapat menghapus data jadwal kelas	Admin menekan tombol delete	Data jadwal kelas terhapus sesuai data yang dipilih	✓	
20.	Menampilkan menu absensi siswa	Admin dapat melihat halaman absensi siswa	Admin menekan tombol navigasi dan masuk ke halaman absensis siswa	Sistem akan menampilkan halaman absensi siswa	✓	
21.	Menambahkan data absensi siswa	Admin dapat menambahkan data absensi siswa	Admin menekan tombol tambah dan menambahkan data baru	Data absensi siswa bertambah	✓	
22.	Mengedit data absensi siswa	Admin dapat mengedit data absensi siswa	Admin menekan tombol edit dan memasukan data baru	Data absensi siswa berubah	✓	
23.	Menghapus data absensi siswa	Admin dapat menghapus absensi siswa	Admin menekan tombol delete	Data absensi siswa terhapus sesuai data yang dipilih	✓	



24.	Menampilkan menu user data	Admin dapat melihat halaman user data	Admin menekan tombol navigasi dan masuk ke halaman user data	Sistem akan menampilkan halaman user data	✓	
25.	Menambahkan data user data	Admin dapat menambahkan data user data	Admin menekan tombol tambah dan menambahkan data baru	Data user data bertambah	✓	
26.	Mengedit data user data	Admin dapat mengedit data user data	Admin menekan tombol edit dan memasukan data baru	Data user data berubah	✓	
27.	Menghapus user data	Admin dapat menghapus user data	Admin menekan tombol delete	Data user data terhapus sesuai data yang dipilih	✓	
<b>Guru</b>						
28.	Menampilkan halaman dashboard guru	Guru dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard guru	✓	
29.	Menambah keterangan	Guru dapat menambah keterangan saat mengajar	Guru menekan tombol keterangan dan menambahkan data baru	Data keterangan bertambah	✓	
30.	Export kegiatan mengajar	Guru dapat mencetak kegiatan mengajar	Guru menekan tombol export	Sistem mencetak sebuah file dengan data jadwal, keterangan dan absensi	✓	
31.	Menampilkan absen siswa	Guru dapat melihat absensi dari siswa	Admin menekan tombol navigasi dan masuk ke	Sistem akan menampilkan halaman absensi siswa	✓	

			halaman user data			
Siswa						
32.	Menampilkan halaman dashboard siswa	Siswa dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard siswa	✓	
33.	Menampilkan menu absensi siswa	Admin dapat melihat halaman absensi siswa	Admin menekan tombol navigasi dan masuk ke halaman absensi siswa	Sistem akan menampilkan halaman absensi siswa	✓	
34.	Menambahkan data absensi siswa	Admin dapat menambahkan data absensi siswa	Admin menekan tombol tambah dan menambahkan data baru	Data absensi siswa bertambah	✓	

Saran dari penguji :

.....  
 .....  
 .....

Semarang,



.....

**KUESIONER PENGUJIAN BLACK BOX PADA "PENERAPAN ALGORITMA GREEDY UNTUK OPTIMALISASI PENJADWALAN MATA PELAJARAN PADA SISTEM INFORMASI AKADEMIK SMK NEGERI 5 KENDAL BERBASIS WEB"**

Nama Penguji : Ramadhan Renaldy, S.kom, m.kom

Tanggal Pengujian : 8 Agustus 2024

No	Nama Pengujian	Tujuan	Skenario	Hasil yang didapatkan	Pengujian	
					Diterima	Ditolak
<b>Admin</b>						
1.	Menampilkan halaman login	User dapat melihat form login dan melakukan login	Ditampilkan saat user mengakses url	Sistem akan menampilkan halaman login	✓	
2.	Menampilkan halaman dashboard admin	Admin dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard admin	✓	
3.	Menampilkan halaman master data	Admin dapat melihat halaman master data	Admin menekan tombol navigasi dan masuk ke halaman master data	Sistem akan menampilkan halaman master data	✓	
4.	Menambahkan data di master data	Admin dapat menambahkan data di master data	Admin menekan tombol tambah dan menambahkan data baru	Data di dalam master data bertambah	✓	
5.	Mengedit data di master data	Admin dapat mengedit data di master data	Admin menekan tombol edit dan memasukan data baru	Data di dalam master data berubah	✓	
6.	Menghapus data di master data	Admin dapat menghapus data di dalam master data	Admin menekan tombol delete	Data di dalam master data terhapus sesuai data yang dipilih	✓	

7.	Menampilkan halaman siswa	Admin dapat melihat halaman siswa	Admin menekan tombol navigasi dan masuk ke halamn siswa	Sistem akan menampilkan halaman siswa	✓	
8.	Menambahkan data siswa	Admin dapat menambahkan data siswa	Admin menakan tombol tambah dan menambahkan data baru	Data siswa bertambah	✓	
9.	Mengedit data siswa	Admin dapat mengedit data siswa	Admin menekan tombol edit dan memasukan data baru	Data siswa terubah	✓	
10.	Menghapus data siswa	Admin dapat menghapus data siswa	Admin menekan tombol delete	Data siswa terhapus sesuai data yang dipilih	✓	
11.	Menampilkan halaman guru	Admin dapat melihat halaman guru	Admin menekan tombol navigasi dan masuk ke halamn guru	Sistem akan menampilkan halaman guru	✓	
12.	Menambahkan data guru	Admin dapat menambahkan data guru	Admin menakan tombol tambah dan menambahkan data baru	Data guru bertambah	✓	
13.	Mengedit data guru	Admin dapat mengedit data guru	Admin menekan tombol edit dan memasukan data baru	Data guru terubah	✓	
14.	Menghapus data guru	Admin dapat menghapus data guru	Admin menekan tombol delete	Data guru terhapus sesuai data yang dipilih	✓	
15.	Menampilkan menu jadwal kelas	Admin dapat melihat halaman jadwal kelas	Admin menekan tombol navigasi dan masuk ke	Sistem akan menampilkan halaman jadwal kelas	✓	



			halamn jadwal kelas			
16.	Menambahkan data jadwal kelas	Admin dapat menambahkan data jadwal kelas	Admin menekan tombol tambah dan menambahkan data baru	Data jadwal kelas bertambah	✓	
17.	Mengedit data jadwal kelas	Admin dapat mengedit data jadwal kelas	Admin menekan tombol edit dan memasukan data baru	Data jadwal kelas berubah	✓	
18.	Rekomendasi Jadwal Kelas	Admin dapat melihat rekomendasi jadwal kelas	Admin menekan tombol rekomendasi	Sistem akan menampilkan data jadwl kelas yang di rekomendasi	✓	
19.	Menghapus data jadwal kelas	Admin dapat menghapus data jadwal kelas	Admin menekan tombol delete	Data jadwal kelas terhapus sesuai data yang dipilih	✓	
20.	Menampilkan menu absensi siswa	Admin dapat melihat halaman absensi siswa	Admin menekan tombol navigasi dan masuk ke halaman absensis siswa	Sistem akan menampilkan halaman absensi siswa	✓	
21.	Menambahkan data absensi siswa	Admin dapat menambahkan data absensi siswa	Admin menekan tombol tambah dan menambahkan data baru	Data absensi siswa bertambah	✓	
22.	Mengedit data absensi siswa	Admin dapat mengedit data absensi siswa	Admin menekan tombol edit dan memasukan data baru	Data absensi siswa berubah	✓	
23.	Menghapus data absensi siswa	Admin dapat menghapus absensi siswa	Admin menekan tombol delete	Data absensi siswa terhapus sesuai data yang dipilih	✓	

24.	Menampilkan menu user data	Admin dapat melihat halaman user data	Admin menekan tombol navigasi dan masuk ke halaman user data	Sistem akan menampilkan halaman user data	✓	
25.	Menambahkan data user data	Admin dapat menambahkan data user data	Admin menekan tombol tambah dan menambahkan data baru	Data user data bertambah	✓	
26.	Mengedit data user data	Admin dapat mengedit data user data	Admin menekan tombol edit dan memasukan data baru	Data user data berubah	✓	
27.	Menghapus user data	Admin dapat menghapus user data	Admin menekan tombol delete	Data user data terhapus sesuai data yang dipilih	✓	
<b>Guru</b>						
28.	Menampilkan halaman dashboard guru	Guru dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard guru	✓	
29.	Menambah keterangan	Guru dapat menambah keterangan saat mengajar	Guru menekan tombol keterangan dan menambahkan data baru	Data keterangan bertambah	✓	
30.	Export kegiatan mengajar	Guru dapat mencetak kegiatan mengajar	Guru menekan tombol export	Sistem mencetak sebuah file dengan data jadwal, keterangan dan absensi	✓	
31.	Menampilkan absen siswa	Guru dapat melihat absensi dari siswa	Admin menekan tombol navigasi dan masuk ke	Sistem akan menampilkan halaman absensi siswa	✓	

			halaman user data			
Siswa						
32.	Menampilkan halaman dashboard siswa	Siswa dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard siswa	✓	
33.	Menampilkan menu absensi siswa	Admin dapat melihat halaman absensi siswa	Admin menekan tombol navigasi dan masuk ke halaman absensi siswa	Sistem akan menampilkan halaman absensi siswa	✓	
34.	Menambahkan data absensi siswa	Admin dapat menambahkan data absensi siswa	Admin menekan tombol tambah dan menambahkan data baru	Data absensi siswa bertambah	✓	

Saran dari penguji :

- \* tombol simpan ditaruh atas saja (UI/UX nya diperbagus / diperbaiki)
- Absensi siswa jangan upload foto tapi langsung foto pakai kamera

Semarang, 8 Agustus 2024



Pamadnan Renaldi

**KUESIONER PENGUJIAN BLACK BOX PADA "PENERAPAN ALGORITMA GREEDY UNTUK OPTIMALISASI PENJADWALAN MATA PELAJARAN PADA SISTEM INFORMASI AKADEMIK SMK NEGERI 5 KENDAL BERBASIS WEB"**

Nama Penguji : Her Latifah PWI MS, Mkom.

Tanggal Pengujian : 8 Agustus 2024

No	Nama Pengujian	Tujuan	Skenario	Hasil yang didapatkan	Pengujian	
					Diterima	Ditolak
<b>Admin</b>						
1.	Menampilkan halaman login	User dapat melihat form login dan melakukan login	Ditampilkan saat user mengakses url	Sistem akan menampilkan halaman login	✓	
2.	Menampilkan halaman dashboard admin	Admin dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard admin	✓	
3.	Menampilkan halaman master data	Admin dapat melihat halaman master data	Admin menekan tombol navigasi dan masuk ke halaman master data	Sistem akan menampilkan halaman master data	✓	
4.	Menambahkan data di master data	Admin dapat menambahkan data di master data	Admin menekan tombol tambah dan menambahkan data baru	Data di dalam master data bertambah	✓	
5.	Mengedit data di master data	Admin dapat mengedit data di master data	Admin menekan tombol edit dan memasukan data baru	Data di dalam master data terubah	✓	
6.	Menghapus data di master data	Admin dapat menghapus data di dalam master data	Admin menekan tombol delete	Data di dalam master data terhapus sesuai data yang dipilih	✓	



7.	Menampilkan halaman siswa	Admin dapat melihat halaman siswa	Admin menekan tombol navigasi dan masuk ke halamn siswa	Sistem akan menampilkan halaman siswa	✓	
8.	Menambahkan data siswa	Admin dapat menambahkan data siswa	Admin menekan tombol tambah dan menambahkan data baru	Data siswa bertambah	✓	
9.	Mengedit data siswa	Admin dapat mengedit data siswa	Admin menekan tombol edit dan memasukan data baru	Data siswa terubah	✓	
10.	Menghapus data siswa	Admin dapat menghapus data siswa	Admin menekan tombol delete	Data siswa terhapus sesuai data yang dipilih	✓	
11.	Menampilkan halaman guru	Admin dapat melihat halaman guru	Admin menekan tombol navigasi dan masuk ke halamn guru	Sistem akan menampilkan halaman guru	✓	
12.	Menambahkan data guru	Admin dapat menambahkan data guru	Admin menakan tombol tambah dan menambahkan data baru	Data guru bertambah	✓	
13.	Mengedit data guru	Admin dapat mengedit data guru	Admin menekan tombol edit dan memasukan data baru	Data guru terubah	✓	
14.	Menghapus data guru	Admin dapat menghapus data guru	Admin menekan tombol delete	Data guru terhapus sesuai data yang dipilih	✓	
15.	Menampilkan menu jadwal kelas	Admin dapat melihat halaman jadwal kelas	Admin menekan tombol navigasi dan masuk ke	Sistem akan menampilkan halaman jadwal kelas	✓	

			halamn jadwal kelas			
16.	Menambahkan data jadwal kelas	Admin dapat menambahkan data jadwal kelas	Admin menekan tombol tambah dan menambahkan data baru	Data jadwal kelas bertambah	✓	
17.	Mengedit data jadwal kelas	Admin dapat mengedit data jadwal kelas	Admin menekan tombol edit dan memasukan data baru	Data jadwal kelas berubah	✓	
18.	Rekomendasi Jadwal Kelas	Admin dapat melihat rekomendasi jadwal kelas	Admin menekan tombol rekomendasi	Sistem akan menampilkan data jadwl kelas yang di rekomendasi	✓	
19.	Menghapus data jadwal kelas	Admin dapat menghapus data jadwal kelas	Admin menekan tombol delete	Data jadwal kelas terhapus sesuai data yang dipilih	✓	
20.	Menampilkan menu absensi siswa	Admin dapat melihat halaman absensi siswa	Admin menekan tombol navigasi dan masuk ke halaman absensis siswa	Sistem akan menampilkan halaman absensi siswa	✓	
21.	Menambahkan data absensi siswa	Admin dapat menambahkan data absensi siswa	Admin menekan tombol tambah dan menambahkan data baru	Data absensi siswa bertambah	✓	
22.	Mengedit data absensi siswa	Admin dapat mengedit data absensi siswa	Admin menekan tombol edit dan memasukan data baru	Data absensi siswa berubah	✓	
23.	Menghapus data absensi siswa	Admin dapat menghapus absensi siswa	Admin menekan tombol delete	Data absensi siswa terhapus sesuai data yang dipilih	✓	

24.	Menampilkan menu user data	Admin dapat melihat halaman user data	Admin menekan tombol navigasi dan masuk ke halaman user data	Sistem akan menampilkan halaman user data	✓	
25.	Menambahkan data user data	Admin dapat menambahkan data user data	Admin menekan tombol tambah dan menambahkan data baru	Data user data bertambah	✓	
26.	Mengedit data user data	Admin dapat mengedit data user data	Admin menekan tombol edit dan memasukan data baru	Data user data berubah	✓	
27.	Menghapus user data	Admin dapat menghapus user data	Admin menekan tombol delete	Data user data terhapus sesuai data yang dipilih	✓	
<b>Guru</b>						
28.	Menampilkan halaman dashboard guru	Guru dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard guru	✓	
29.	Menambah keterangan	Guru dapat menambah keterangan saat mengajar	Guru menekan tombol keterangan dan menambahkan data baru	Data keterangan bertambah	✓	
30.	Export kegiatan mengajar	Guru dapat mencetak kegiatan mengajar	Guru menekan tombol export	Sistem mencetak sebuah file dengan data jadwal, keterangan dan absensi	✓	
31.	Menampilkan absen siswa	Guru dapat melihat absensi dari siswa	Admin menekan tombol navigasi dan masuk ke	Sistem akan menampilkan halaman absensi siswa	✓	

			halaman user data			
Siswa						
32.	Menampilkan halaman dashboard siswa	Siswa dapat melihat halaman dashboard	Halaman dashboard dapat diakses setelah melakukan login	Sistem akan menampilkan halaman dashboard siswa	✓	
33.	Menampilkan menu absensi siswa	Admin dapat melihat halaman absensi siswa	Admin menekan tombol navigasi dan masuk ke halaman absensi siswa	Sistem akan menampilkan halaman absensi siswa	✓	
34.	Menambahkan data absensi siswa	Admin dapat menambahkan data absensi siswa	Admin menekan tombol tambah dan menambahkan data baru	Data absensi siswa bertambah	✓	

Saran dari penguji :

*cxpant lokal*  
*wali kelas → data siswa*

Semarang,

  
 Nur. Lestari, S.Pd, M.Pd, M.Kom



## e. User Acceptance Testing

**KUESIONER PENGUJIAN USER ACCEPTANCE TESTING (UAT) PADA  
"PENERAPAN ALGORITMA *GREEDY* UNTUK OPTIMALISASI PENJADWALAN  
MATA PELAJARAN PADA SISTEM INFORMASI AKADEMIK SMK NEGERI 5  
KENDAL BERBASIS WEB"**

Nama Penguji : Defi Purwanti, S.Pd.

Tanggal Pengujian : 5 Agustus 2024

No	Pertanyaan	Skor				
		Tidak setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat setuju
<b>Aspek kegunaan</b>						
1.	Apakah Anda dapat dengan mudah menemukan dan mengakses jadwal mata pelajaran di aplikasi?					✓
2.	Apakah tampilan jadwal mata pelajaran jelas dan mudah dibaca?				✓	
3.	Apakah fitur optimalisasi penjadwalan dengan algoritma Greedy berfungsi dan mudah digunakan?				✓	
4	Apakah informasi mengenai data master, siswa, guru, daftar kelas dan user data yang tersedia ditampilkan dengan jelas?				✓	
<b>Aspek kemudahan pengguna (user)</b>						
5.	Apakah Anda merasa informasi yang ditampilkan pada aplikasi mudah untuk dipahami dan akurat sesuai dengan jadwal yang sebenarnya?					✓

6.	Apakah fitur penjadwalan mata pelajaran ditampilkan dengan jelas dan mudah diakses di aplikasi?					✓
<b>Aspek tampilan (user interface)</b>						
7.	Seberapa mudah Anda membaca dan memahami informasi yang ditampilkan?					✓
8.	Apakah ukuran dari text yang ditampilkan sudah sesuai?				✓	
9.	Apakah penggunaan ikon pada aplikasi membantu dalam navigasi dan pemahaman fungsi-fungsi yang ada?				✓	
10.	Apakah tata letak menu dan tombol navigasi pada aplikasi memudahkan Anda dalam mengakses berbagai fitur?				✓	

**Keterangan :**

1 = Tidak setuju

2 = Kurang setuju

3 = Cukup setuju

4 = Setuju

5 = Sangat setuju

Kendal, 5 - 8 - 2024


  
Defi Purwanti

**KUESIONER PENGUJIAN USER ACCEPTANCE TESTING (UAT) PADA  
 “PENERAPAN ALGORITMA *GREEDY* UNTUK OPTIMALISASI PENJADWALAN  
 MATA PELAJARAN PADA SISTEM INFORMASI AKADEMIK SMK NEGERI 5  
 KENDAL BERBASIS WEB”**

Nama Penguji : *Nidya Setiora*  
 Tanggal Pengujian : *5 Agustus 2024*

No	Pertanyaan	Skor				
		Tidak setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat setuju
<b>Aspek kegunaan</b>						
1.	Apakah Anda dapat dengan mudah menemukan dan mengakses jadwal mata pelajaran di aplikasi?					✓
2.	Apakah tampilan jadwal mata pelajaran jelas dan mudah dibaca?					✓
3.	Apakah fitur optimalisasi penjadwalan dengan algoritma Greedy berfungsi dan mudah digunakan?				✓	
4	Apakah informasi mengenai data master, siswa, guru, daftar kelas dan user data yang tersedia ditampilkan dengan jelas?				✓	
<b>Aspek kemudahan pengguna (<i>user</i>)</b>						
5.	Apakah Anda merasa informasi yang ditampilkan pada aplikasi mudah untuk dipahami dan akurat sesuai dengan jadwal yang sebenarnya?				✓	

6.	Apakah fitur penjadwalan mata pelajaran ditampilkan dengan jelas dan mudah diakses di aplikasi?					✓
<b>Aspek tampilan (user interface)</b>						
7.	Seberapa mudah Anda membaca dan memahami informasi yang ditampilkan?					✓
8.	Apakah ukuran dari text yang ditampilkan sudah sesuai?					✓
9.	Apakah penggunaan ikon pada aplikasi membantu dalam navigasi dan pemahaman fungsi-fungsi yang ada?					✓
10.	Apakah tata letak menu dan tombol navigasi pada aplikasi memudahkan Anda dalam mengakses berbagai fitur?					✓

**Keterangan :**

1 = Tidak setuju

2 = Kurang setuju

3 = Cukup setuju

4 = Setuju

5 = Sangat setuju

  
 Nidy Seioriz



**KUESIONER PENGUJIAN USER ACCEPTANCE TESTING (UAT) PADA  
"PENERAPAN ALGORITMA *GREEDY* UNTUK OPTIMALISASI PENJADWALAN  
MATA PELAJARAN PADA SISTEM INFORMASI AKADEMIK SMK NEGERI 5  
KENDAL BERBASIS WEB"**

Nama Penguji : Hari Sutanto

Tanggal Pengujian : 5 Agustus 2024

No	Pertanyaan	Skor				
		Tidak setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat setuju
<b>Aspek kegunaan</b>						
1.	Apakah Anda dapat dengan mudah menemukan dan mengakses jadwal mata pelajaran di aplikasi?				✓	
2.	Apakah tampilan jadwal mata pelajaran jelas dan mudah dibaca?					✓
3.	Apakah fitur optimalisasi penjadwalan dengan algoritma Greedy berfungsi dan mudah digunakan?					✓
4	Apakah informasi mengenai data master, siswa, guru, daftar kelas dan user data yang tersedia ditampilkan dengan jelas?					✓
<b>Aspek kemudahan pengguna (user)</b>						
5.	Apakah Anda merasa informasi yang ditampilkan pada aplikasi mudah untuk dipahami dan akurat sesuai dengan jadwal yang sebenarnya?				✓	

6.	Apakah fitur penjadwalan mata pelajaran ditampilkan dengan jelas dan mudah diakses di aplikasi?				✓	
<b>Aspek tampilan (user interface)</b>						
7.	Seberapa mudah Anda membaca dan memahami informasi yang ditampilkan?					✓
8.	Apakah ukuran dari text yang ditampilkan sudah sesuai?				✓	
9.	Apakah penggunaan ikon pada aplikasi membantu dalam navigasi dan pemahaman fungsi-fungsi yang ada?					✓
10.	Apakah tata letak menu dan tombol navigasi pada aplikasi memudahkan Anda dalam mengakses berbagai fitur?					✓

**Keterangan :**

**1 = Tidak setuju**

**2 = Kurang setuju**

**3 = Cukup setuju**

**4 = Setuju**

**5 = Sangat setuju**

.....  
  
 .....

**KUESIONER PENGUJIAN USER ACCEPTANCE TESTING (UAT) PADA  
"PENERAPAN ALGORITMA *GREEDY* UNTUK OPTIMALISASI PENJADWALAN  
MATA PELAJARAN PADA SISTEM INFORMASI AKADEMIK SMK NEGERI 5  
KENDAL BERBASIS WEB"**

Nama Penguji : Diah Mesayu P.

Tanggal Pengujian : 5 Agustus 2024

No	Pertanyaan	Skor				
		Tidak setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat setuju
<b>Aspek kegunaan</b>						
1.	Apakah Anda dapat dengan mudah menemukan dan mengakses jadwal mata pelajaran di aplikasi?				✓	
2.	Apakah tampilan jadwal mata pelajaran jelas dan mudah dibaca?				✓	
3.	Apakah fitur optimalisasi penjadwalan dengan algoritma Greedy berfungsi dan mudah digunakan?				✓	
4	Apakah informasi mengenai data master, siswa, guru, daftar kelas dan user data yang tersedia ditampilkan dengan jelas?					✓
<b>Aspek kemudahan pengguna (user)</b>						
5.	Apakah Anda merasa informasi yang ditampilkan pada aplikasi mudah untuk dipahami dan akurat sesuai dengan jadwal yang sebenarnya?				✓	

6.	Apakah fitur penjadwalan mata pelajaran ditampilkan dengan jelas dan mudah diakses di aplikasi?				✓	
<b>Aspek tampilan (user interface)</b>						
7.	Seberapa mudah Anda membaca dan memahami informasi yang ditampilkan?				✓	
8.	Apakah ukuran dari text yang ditampilkan sudah sesuai?					✓
9.	Apakah penggunaan ikon pada aplikasi membantu dalam navigasi dan pemahaman fungsi-fungsi yang ada?				✓	
10.	Apakah tata letak menu dan tombol navigasi pada aplikasi memudahkan Anda dalam mengakses berbagai fitur?				✓	

**Keterangan :**

1 = Tidak setuju

2 = Kurang setuju

3 = Cukup setuju

4 = Setuju

5 = Sangat setuju

Penda, 5 Agustus 2024

Diah Mesayu P.

**KUESIONER PENGUJIAN USER ACCEPTANCE TESTING (UAT) PADA  
"PENERAPAN ALGORITMA GREEDY UNTUK OPTIMALISASI PENJADWALAN  
MATA PELAJARAN PADA SISTEM INFORMASI AKADEMIK SMK NEGERI 5  
KENDAL BERBASIS WEB"**

Nama Penguji : *Itmi Nugroho, S.Pd*

Tanggal Pengujian : *5 Agustus 2024*

No	Pertanyaan	Skor				
		Tidak setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat setuju
<b>Aspek kegunaan</b>						
1.	Apakah Anda dapat dengan mudah menemukan dan mengakses jadwal mata pelajaran di aplikasi?				✓	
2.	Apakah tampilan jadwal mata pelajaran jelas dan mudah dibaca?					✓
3.	Apakah fitur optimalisasi penjadwalan dengan algoritma Greedy berfungsi dan mudah digunakan?					✓
4	Apakah informasi mengenai data master, siswa, guru, daftar kelas dan user data yang tersedia ditampilkan dengan jelas?					✓
<b>Aspek kemudahan pengguna (user)</b>						
5.	Apakah Anda merasa informasi yang ditampilkan pada aplikasi mudah untuk dipahami dan akurat sesuai dengan jadwal yang sebenarnya?				✓	



6.	Apakah fitur penjadwalan mata pelajaran ditampilkan dengan jelas dan mudah diakses di aplikasi?				✓	
<b>Aspek tampilan (user interface)</b>						
7.	Seberapa mudah Anda membaca dan memahami informasi yang ditampilkan?				✓	
8.	Apakah ukuran dari text yang ditampilkan sudah sesuai?					✓
9.	Apakah penggunaan ikon pada aplikasi membantu dalam navigasi dan pemahaman fungsi-fungsi yang ada?				✓	
10.	Apakah tata letak menu dan tombol navigasi pada aplikasi memudahkan Anda dalam mengakses berbagai fitur?				✓	

**Keterangan :**

1 = Tidak setuju

2 = Kurang setuju

3 = Cukup setuju

4 = Setuju

5 = Sangat setuju

Kendal, 5 Agustus 2024


  
Lina Nugroho, S.Pd