

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI  
INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH**

**LAPORAN TUGAS AKHIR**

**SISTEM INFORMASI AKADEMIK BERBASIS WEB  
PADA SEKOLAH MENENGAH KEJURUAN  
UTAMA BAKTI PALEMBANG**



**Diajukan Oleh :  
MGS DJAHWA HARIDSYAH  
031200053**

**Untuk Memenuhi Sebagian dari Syarat  
Mencapai Gelar Ahli Madya**

**PALEMBANG  
2023**

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI  
INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH**

**LAPORAN TUGAS AKHIR**

**SISTEM INFORMASI AKADEMIK BERBASIS WEB  
PADA SEKOLAH MENENGAH KEJURUAN  
UTAMA BAKTI PALEMBANG**



**Diajukan Oleh :  
MGS DJAHWA HARIDSYAH  
031200053**

**Untuk Memenuhi Sebagian dari Syarat  
Mencapai Gelar Ahli Madya**

**PALEMBANG  
2023**

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI**

---

**INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH  
HALAMAN PENGESAHAN PEMBIMBING TUGAS AKHIR**

**NAMA : MGS DJAHWA HARIDSYAH  
NOMOR POKOK : 031200053  
PROGRAM STUDI : SISTEM INFORMASI  
JENJANG PENDIDIKAN : DIPLOMA TIGA  
JUDUL : SISTEM INFORMASI AKADEMIK BERBASIS  
WEB PADA SEKOLAH MENENGAH  
KEJURUAN UTAMA BAKTI PALEMBANG**

**Tanggal : 4 Agustus 2023  
Pembimbing**

**Mengetahui,  
Rektor**

**Jaka Purnama, S.Kom., M.Kom.**

**NIDN : 0219089401**

**Benedictus Effendi, S.T., M.T.**

**NIP : 09.PCT.13**

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI**

---

**INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH  
HALAMAN PENGESAHAN PENGUJI TUGAS AKHIR**

**NAMA : MGS DJAHWA HARIDSYAH**  
**NOMOR POKOK : 031200053**  
**PROGRAM STUDI : SISTEM INFORMASI**  
**JENJANG PENDIDIKAN : DIPLOMA TIGA**  
**JUDUL : SISTEM INFORMASI AKADEMIK BERBASIS  
WEB PADA SEKOLAH MENENGAH  
KEJURUAN UTAMA BAKTI PALEMBANG**

**Tanggal : 29 Agustus 2023**  
**Penguji 1**

**Tanggal: 28 Agustus 2023**  
**Penguji 2**

**Hendra Effendi, S.Kom., M.Kom.**  
**NIDN : 0217108001**

**Yayuk Ike Melani, S.Kom., M.Kom.**  
**NIDN : 0224059102**

**Menyetujui**  
**Rektor**

**Benedictus Effendi, S.T., M.T.**  
**NIP : 09.PCT.13**

**MOTO :**

*“Terus berusaha jangan sampai menyerah”*

**Kupersembahkan Kepada :**

- Ayahanda dan Ibunda Tercinta
- Saudara-saudaraku tersayang
- Para pendidik yang kuhormati
- Teman – teman terbaik

## KATA PENGANTAR

Segala puji dan syukur kami panjatkan kehadirat Allah SWT atas segala limpahan rahmat-nya. Shalawat serta salam semoga selalu tercurah limpahkan kepada junjungan kami Nabi Muhammad SAW sehingga penulis dapat menyelenggarakan kegiatan Tugas Akhir dan menyelesaikan laporan kegiatan Tugas Akhir dengan judul **“Sistem Informasi Akademik Berbasis Web Pada Sekolah Menengah Kejuruan Utama Bakti Palembang”**.

Sebagai rasa syukur dan hormat, melalui kesempatan ini penulis mengucapkan terima kasih banyak kepada semua pihak yang telah membantu, serta memberikan segala saran, motivasi dalam penulisan laporan Tugas Akhir ini. Untuk itu penulis mengucapkan terima kasih kepada:

1. Kepada Bapak Benedictus Effendi, S.T., M.T. Selaku Rektor Institut Teknologi dan Bisnis PalComTech.
2. Kepada Ibu Adelin, S.T., M.Kom., Selaku Wakil Rektor 1 Institut Teknologi dan Bisnis PalComTech
3. Kepada Ibu Dini Hari Pertiwi, S.Kom., M.Kom., Selaku Ketua Program Studi D3 Sistem Informasi.
4. Kepada Dosen Pembimbing LTA Jaka Purnama, M.Kom.

Demikian kata pengantar ini, dengan harapan besar semoga laporan Praktik Kerja Lapangan ini dapat bermanfaat dan berguna bagi para pembaca. Dengan segala kesadaran, penulis menyadari bahwa laporan Praktik Kerja Lapangan ini masih mempunyai banyak kekurangan sehingga harapan penulis mendapatkan kritik dan saran untuk menghasilkan sesuatu yang lebih baik.

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>HALAMAN PENGESAHAN PEMBIMBING TUGAS AKHIR</b> .....	<b>ii</b>
<b>HALAMAN PENGESAHAN PENGUJI TUGAS AKHIR</b> .....	<b>iii</b>
<b>KATA PENGANTAR</b> .....	<b>v</b>
<b>DAFTAR ISI</b> .....	<b>vi</b>
<b>DAFTAR GAMBAR</b> .....	<b>ix</b>
<b>DAFTAR TABEL</b> .....	<b>xi</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xii</b>
<b>ABSTRACT</b> .....	<b>xiii</b>
<b>ABSTRAK</b> .....	<b>xiv</b>
<b>BAB I     PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	5
1.5.1 Bagi Peneliti .....	5
1.5.2 Bagi Sekolah Menengah Kejuruan Utama Bakti .....	5
1.5.3 Bagi Akademik .....	5
1.6 Sistematika Penulisan .....	6
<b>BAB II    TINJAUAN PUSTAKA</b>	
2.1 Landasan Teori .....	8

2.1.1 Sekolah Menengah Kejuruan .....	8
2.1.2 Sistem Informasi Akademik .....	8
2.1.3 Website .....	9
2.1.4 Database (Basis Data) .....	9
2.1.5 PHP .....	9
2.1.6 Laravel .....	9
2.2 Penelitian Terdahulu .....	10
2.3 Kerangka Pemikiran .....	12
2.4 Objek Penelitian .....	13
2.4.1 Sejarah Perusahaan .....	13
2.4.2 Visi dan Misi .....	15
2.4.3 Struktur Organisasi .....	17
2.4.4 Uraian Kegiatan dan Wewenang .....	18

### **BAB III METODE PENELITIAN**

3.1 Lokasi dan Waktu Penelitian .....	23
3.1.1 Lokasi Penelitian .....	23
3.1.2 Waktu Penelitian .....	23
3.2 Jenis Data .....	23
3.2.1 Data Primer .....	23
3.2.2 Data Sekunder .....	24
3.3 Teknik Pengumpulan Data .....	24
3.3.1 Wawancara (interview) .....	24
3.3.2 Observasi (Pengamatan) .....	25
3.3.3 Studi Pustaka .....	25
3.4 Alat Pengembangan Sistem .....	25



3.4.1 Model Proses .....	26
3.5 Metode Pengembangan Sistem .....	34
<b>BAB IV HASIL DAN PEMBAHASAN</b>	
4.1 Hasil .....	37
4.1.1 Requirement .....	37
4.2 Design .....	50
4.2.1 Design Interface .....	50
4.2.2 Desain Database .....	61
4.3 Implementation .....	68
4.4 Pengujian Perangkat Lunak .....	79
4.5 Kesimpulan Hasil Pengujian .....	81
<b>BAB V KESIMPULAN DAN SARAN</b>	
5.1 Kesimpulan .....	83
5.2 Saran .....	83
<b>DAFTAR PUSTAKA .....</b>	<b>xv</b>
<b>HALAMAN LAMPIRAN .....</b>	<b>xvii</b>

## DAFTAR GAMBAR

Gambar 2.1 Kerangka Penelitian .....	12
Gambar 2. 2 Struktur Organisasi .....	17
Gambar 3.1 Metode <i>Waterfall</i> .....	34
Gambar 4. 1 <i>Use Case Diagram</i> Operator .....	38
Gambar 4. 2 <i>Use Case Diagram</i> Kepala Sekolah .....	39
Gambar 4. 3 <i>Use Case Diagram</i> Waka Kurikulum .....	40
Gambar 4. 4 <i>Use Case Diagram</i> Wali Kelas .....	40
Gambar 4. 5 <i>Use Case Diagram</i> Guru .....	41
Gambar 4. 6 <i>Use Case Diagram</i> Siswa .....	41
Gambar 4.7 <i>Activity Diagram</i> Operator .....	43
Gambar 4. 8 <i>Activity Diagram</i> Kepala Sekolah .....	44
Gambar 4. 9 <i>Activity Diagram</i> Waka Kurikulum .....	45
Gambar 4. 10 <i>Activity Diagram</i> Wali Kelas .....	46
Gambar 4. 11 <i>Activity Diagram</i> Guru .....	47
Gambar 4. 12 <i>Activity Diagram</i> Siswa .....	48
Gambar 4. 13 <i>Class Diagram</i> .....	49
Gambar 4. 14 Desain Halaman <i>Dashboard</i> Operator .....	50
Gambar 4. 15 Desain Halaman <i>Dashboard</i> Kepala Sekolah .....	51
Gambar 4. 16 Desain Halaman <i>Dashboard</i> Waka Kurikulum .....	52
Gambar 4. 17 Desain Halaman <i>Dashboard</i> Guru .....	52
Gambar 4. 18 Desain Halaman <i>Dashboard</i> Wali Kelas .....	53
Gambar 4. 19 Desain Halaman <i>Dashboard</i> Siswa .....	53
Gambar 4. 20 Desain Halaman Master-Jurusan .....	54
Gambar 4. 21 Desain Halaman Master-Kelas .....	54
Gambar 4. 22 Desain Halaman Master-Tahun Ajaran .....	55
Gambar 4. 23 Desain Halaman Master-Jam Pelajaran .....	55
Gambar 4. 24 Desain Halaman Master-Nama Pelajaran .....	56
Gambar 4. 25 Desain Halaman Master-Predikat .....	56

Gambar 4. 26 Desain Halaman Master-Data Mata Pelajaran .....	57
Gambar 4. 27 Desain Halaman Data Siswa .....	57
Gambar 4. 28 Desain Halaman Data Guru .....	58
Gambar 4. 29 Desain Halaman Data Akademik .....	58
Gambar 4. 30 Desain Halaman Data Siswa .....	59
Gambar 4. 31 Desain Halaman Jadwal Pelajaran .....	59
Gambar 4. 32 Desain Halaman Data Nilai .....	60
Gambar 4. 33 Desain Halaman Pengumuman .....	60
Gambar 4. 34 Desain Halaman <i>Management User</i> .....	61
Gambar 4. 35 Desain Halaman <i>Dashboard</i> Operator .....	68
Gambar 4. 36 Desain Halaman <i>Dashboard</i> Kepala Sekolah .....	69
Gambar 4. 37 Desain Halaman <i>Dashboard</i> Waka Kurikulum .....	69
Gambar 4. 38 Desain Halaman <i>Dashboard</i> Guru .....	70
Gambar 4. 39 Desain Halaman <i>Dashboard</i> Wali Kelas .....	71
Gambar 4. 40 Desain Halaman <i>Dashboard</i> Siswa .....	71
Gambar 4. 41 Desain Halaman Master-Jurusan .....	72
Gambar 4. 42 Desain Halaman Master-Kelas .....	72
Gambar 4. 43 Desain Halaman Master-Tahun Ajaran .....	73
Gambar 4. 44 Desain Halaman Master-Jam Pelajaran .....	73
Gambar 4. 45 Desain Halaman Master-Nama Pelajaran .....	74
Gambar 4. 46 Desain Halaman Master-Predikat .....	74
Gambar 4. 47 Desain Halaman Master-Data Mapel .....	75
Gambar 4. 48 Desain Halaman Data Siswa .....	75
Gambar 4. 49 Desain Halaman Data Guru .....	76
Gambar 4. 50 Desain Halaman Data Akademik .....	76
Gambar 4. 51 Desain Halaman Jadwal Pelajaran .....	77
Gambar 4. 52 Desain Halaman Data Nilai .....	77
Gambar 4. 53 Desain Halaman Pengumuman .....	78

## DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu .....	10
Tabel 3.1 Waktu Penelitian .....	23
Tabel 3. 2 <i>Use Case Diagram</i> .....	27
Tabel 3. 3 <i>Activity Diagram</i> .....	29
Tabel 3. 4 <i>Sequence Diagram</i> .....	31
Tabel 3. 5 <i>Class Diagram</i> .....	32
Tabel 4. 1 <i>User</i> .....	61
Tabel 4. 2 <i>Student</i> .....	62
Tabel 4. 3 <i>Teacher</i> .....	63
Tabel 4. 4 <i>Classes</i> .....	63
Tabel 4. 5 <i>Course</i> .....	64
Tabel 4. 6 <i>Classrooms</i> .....	64
Tabel 4. 7 <i>School Year</i> .....	64
Tabel 4. 8 <i>Course Has Teacher</i> .....	65
Tabel 4. 9 <i>Schedule</i> .....	65
Tabel 4. 10 <i>Courses</i> .....	66
Tabel 4. 11 <i>Course</i> .....	66
Tabel 4. 12 <i>Scores</i> .....	66
Tabel 4. 13 <i>Recab Scores</i> .....	67
Tabel 4. 14 <i>Announcement</i> .....	67
Tabel 4. 15 Tabel Pengujian Sistem .....	79

## DAFTAR LAMPIRAN

1. Lampiran 1. *Form* Topik dan Judul (*Fotocopy*)
2. Lampiran 2. Surat Balasan Riset dari Perusahaan (*Fotocopy*)
3. Lampiran 3. *Form* Konsultasi (*Fotocopy*)
4. Lampiran 4. Surat Pernyataan (*Fotocopy*)
5. Lampiran 5. *Form* Revisi Ujian Pra Sidang (*Fotocopy*)
6. Lampiran 6. *Form* Revisi Ujian LTA (Asli)
7. Lampiran 7. *Listing Code*

## **ABSTRACT**

*MGS DJAHWA HARIDSYAH . Web Based Academic Information System of Vocational Secondary School Utama Bakti Palembang.*

*The Academic Information System (AIS) is a system that manages academic data in educational institutions. It helps enhance education by providing evaluation data, curriculum planning, and features such as student data processing, scheduling, grades, and reports. SMK Utama Bakti, a technical school with 638 students, manages data using Microsoft Excel and Microsoft Word, but is limited to hardcopy. This restricts real-time data access and information delivery. To overcome this, an AIS is needed to automatically record grades, provide real-time access, and enable direct announcements. Using the waterfall method and UML, this system utilizes Use Case, Activity, Class, and Sequence diagrams. This system efficiently delivers academic information to teachers, students, and principals, facilitating data access and academic data integration. Suggestions include monitoring the system for security and additional features such as payment confirmation and automatic attendance.*

***Keywords: Academic Information System, Object-oriented Programming, Blackbox Testing.***

## ABSTRAK

MGS DJAHWA HARIDSYAH . Sistem Informasi Berbasis Web Pada Sekolah Menengah Kejuruan Utama Bakti Palembang.

Sistem Informasi Akademik (SIA) adalah sistem yang mengelola data akademik di lembaga pendidikan. Ini membantu meningkatkan pendidikan dengan menyediakan data evaluasi, perencanaan kurikulum, dan fitur seperti pengolahan data siswa, penjadwalan, nilai, serta laporan. SMK Utama Bakti, sekolah teknik dengan 638 siswa, mengelola data menggunakan *Microsoft Excel* dan *Microsoft Word*, tetapi terbatas pada *hardcopy*. Hal ini membatasi akses data *real-time* dan pengiriman informasi. Untuk mengatasi ini, diperlukan SIA yang otomatis merekam nilai, akses *real-time*, dan pengumuman langsung. Menggunakan metode *waterfall* dan UML, sistem ini memanfaatkan *diagram Use Case*, *Activity*, *Class*, dan *Sequence*. Sistem ini memberikan informasi akademik secara efisien kepada guru, siswa, dan kepala sekolah, memfasilitasi akses data, dan integrasi data akademik. Saran termasuk memantau sistem untuk keamanan dan fitur tambahan seperti konfirmasi pembayaran dan absensi otomatis.

**Kata Kunci : Sistem Informasi Akademik , Perancangan Sistem Berorientasi Objek, Pengujian *Black Box*.**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Menurut Widiati (2019), Sistem Informasi Akademik (SIA) adalah sebuah sistem informasi yang dirancang khusus untuk mengelola data akademik pada suatu institusi pendidikan seperti sekolah atau perguruan tinggi. Sistem ini dapat mengumpulkan, menyimpan, memproses dan mengelola data terkait siswa, guru mata pelajaran, nilai dan akademik lainnya. Sistem informasi akademik dapat membantu meningkatkan kualitas pendidikan dengan menyediakan data dan laporan yang diperlukan untuk melakukan evaluasi, perencanaan dan pengembangan kurikulum. Beberapa fitur umum yang dapat dimiliki oleh sistem informasi akademik antara lain pengolahan data siswa, penjadwalan mata pelajaran, pengolahan nilai, pembayaran, serta laporan dan analisis data akademik

Sekolah Menengah Kejuruan (SMK) Utama Bakti merupakan sekolah kejuruan yang bergerak dalam bidang keahlian teknik dan industri. Sekolah ini beralamat di Jl. STM Utama Bakti Lebong Siareng, Palembang. SMK Utama Bakti Palembang memiliki tiga program keahlian, yaitu Teknik Kendaraan Ringan (TKR), Teknik Instalasi Tenaga Listrik (TITL), dan Teknik Komputer Jaringan (TKJ). Pada tahun 2023, jumlah seluruh siswa SMK Utama Bakti Palembang adalah 638 siswa yang terdiri dari 516 siswa dan 122 siswi.



Pada SMK Utama Bakti Palembang, pengolahan data siswa, data nilai, data guru, data kelas, data mata pelajaran, informasi pengumuman dan jadwal mengajar mengandalkan penggunaan *Microsoft Excel* dan *Microsoft Word*. Namun, peran kedua aplikasi tersebut terbatas pada pembuatan surat dan form saja yang kemudian dicetak dalam bentuk *hardcopy* untuk keperluan pengarsipan. Ketergantungan pada *file* fisik ini mengakibatkan keterbatasan dalam akses data secara *real-time* dan kesulitan dalam pencarian data. Selain itu metode penyampaian informasi yang terbatas seperti penggunaan surat, pengumuman di papan pengumuman dan penyampaian informasi secara lisan. Hal menyebabkan penyampaian informasi menjadi lebih lambat, terabaikan dan tidak tersampaikan kepada semua pihak yang berkepentingan.

Untuk mengatasi masalah yang dihadapi oleh SMK Utama Bakti Palembang dalam pengelolaan data siswa, guru, perekapan nilai dan penyampaian informasi, diperlukan implementasi sebuah sistem informasi akademik yang mampu melakukan perekapan nilai secara otomatis, dapat di akses secara *real-time* dan menyampaikan pengumuman secara langsung.

Dengan menerapkan sistem informasi akademik yang terintegrasi, SMK Utama Bakti Palembang dapat meningkatkan efisiensi dan aksesibilitas dalam pengelolaan data akademik. Sistem ini akan memberikan kemudahan dalam pencarian data, perekapan nilai serta penyampaian informasi yang lebih efektif dan transparan bagi semua

pihak yang terlibat. Maka dari itu penulis mengambil judul “**Sistem Informasi Akademik Berbasis Web Pada Sekolah Menengah Kejuruan Utama Bakti**”.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana cara membuat proses pengolahan data akademik di SMK Utama Bakti Palembang menjadi lebih cepat dan dapat diakses secara *real-time*?
2. Bagaimana cara informasi pengumuman dapat tersampaikan secara cepat?

## 1.3 Batasan Masalah

Batasan masalah pada penelitian ini akan dibahas sebagai berikut :

1. Membangun sebuah sistem informasi akademik yang terintegrasi untuk mengelola akademik seperti data siswa, data guru, penilaian siswa, jadwal mengajar dan penyampaian informasi pengumuman. Dengan adanya sistem informasi akademik di SMK Utama Bakti Palembang dapat dilakukan secara terintegrasi.
2. *Website* sistem informasi akademik ini dapat mengakses data akademik maupun informasi dengan mudah dan cepat.
3. Website ini dibuat dengan menggunakan bahasa pemrograman PHP, *Laravel Framework* 10 dan MySQL sebagai *database*.

4. Pengumpulan data yang digunakan pada penelitian ini menggunakan beberapa teknik pengumpulan data yaitu observasi (pengamatan), wawancara (*interview*) dan studi pustaka.
5. Model perancangan yang digunakan yaitu *Unified Modeling Language* (UML) dengan diagram seperti *Diagram Use Case*, *Sequence Diagram*, *Class Diagram* dan *Activity Diagram*.
6. Teknik pengujian yang digunakan dalam penelitian ini adalah *equivalence partitioning*, yang merupakan metode dalam pengujian *blackbox testing*.
7. Menu yang tersedia pada sistem informasi akademik yang akan dibangun yaitu:
  - a) Operator dapat mem~~an~~agement akun *user*.
  - b) Wakil kepala sekolah bidang kurikulum dapat mengelola data tahun akademik, penetapan wali kelas dan jadwal pelajaran.
  - c) Wali kelas dan guru dapat melihat data nilai dan data siswa serta dapat membuat jadwal pelajaran.
  - d) Siswa dapat melihat informasi nilai, informasi pengumuman dan jadwal pelajaran.
  - e) Orang tua dapat melihat informasi pengumuman dan diberitahukan melalui chatbot aplikasi *whatsapp*.

#### **1.4 Tujuan Penelitian**

Tujuan penelitian ini adalah untuk membuat aplikasi sistem informasi akademik yang dapat mempermudah proses pengolahan data

akademik yang dilakukan oleh wakil kepala sekolah, wali kelas, dan guru. Serta mempermudah penyampaian informasi menjadi lebih efektif dan efisien.

## **1.5 Manfaat Penelitian**

### **1.5.1 Bagi Peneliti**

1. Menambah pengalaman dan memperluas pengetahuan serta kemampuan dalam membangun sebuah sistem berbasis web dengan menggunakan bahasa pemrograman PHP dan MySQL.
2. Peneliti dapat menerapkan ilmu yang telah didapatkan selama belajar di Institut Teknologi dan Bisnis PalComTech.

### **1.5.2 Bagi Sekolah Menengah Kejuruan Utama Bakti**

1. Membantu pengolahan sistem akademik dengan lebih baik, mudah dan efisien, serta mampu memberikan penerapan aplikasi sistem informasi akademik pada SMK Utama Bakti Palembang.
2. Mempermudah aktivitas yang dilakukan oleh kepala sekolah, guru, dan tenaga kependidikan seperti kegiatan monitoring, merekap nilai, dan penjadwalan.

### **1.5.3 Bagi Akademik**

1. Mengetahui kemampuan yang dimiliki mahasiswa dalam menguasai ilmu yang didapatkan selama kuliah.
2. Memberikan gambaran tentang kesiapan mahasiswa dalam menghadapi dunia kerja yang sebenarnya.
3. Dapat menjadi bahan acuan dan pustaka untuk penelitian selanjutnya.

## **1.6 Sistematika Penulisan**

Demi mewujudkan suatu hasil yang baik dalam penyusunan laporan tugas akhir, penulisan menggunakan pembahasan yang sesuai dengan ketentuan yang diberikan, sistematika penulisan tersebut meliputi :

### **BAB I. PENDAHULUAN**

Pada bab ini berisi uraian latar belakang, rumusan masalah, ruang lingkup, tujuan penelitian, manfaat penelitian, dan sistematika penulisan

### **BAB II. GAMBARAN UMUM PERUSAHAAN**

Pada bab ini diuraikan mengenai gambaran umum perusahaan.

### **BAB III. TINJAUAN PUSTAKA**

Bab ini penulis akan membuat teori yang berdasarkan penulisan skripsi ini yang terdiri dari teori pendukung, hasil penelitian terdahulu dan kerangka penelitian.

### **BAB IV. METODE PENELITIAN**

Bab ini penulis membahas lokasi dan waktu penelitian, jenis data, teknik pengumpulan data, dan jenis penelitian dan alat serta teknik pengembangan sistem.

### **BAB V. HASIL DAN PEMBAHASAN**

Bab ini penulis membahas hasil-hasil yang diperoleh dalam penelitian dan pembahasan terhadap hasil yang telah dicapai

maupun masalah-masalah yang telah ditemukan selama penelitian serta pengujian sistem yang dibuat.

#### **BAB IV. PENUTUP**

Pada akhir bab ini hanya menguraikan beberapa simpulan dan saran dari pembahasan dalam penerapan metode *waterfall* dalam membangun sistem informasi akademik berbasis web pada SMK Utama Bakti Palembang menggunakan metode *waterfall* yang terdapat dalam bab-bab sebelumnya.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

##### **2.1.1 Sekolah Menengah Kejuruan**

Menurut Zahara (2021), Sekolah Menengah Kejuruan (SMK) merupakan lembaga pendidikan yang memiliki tujuan mempersiapkan siswa untuk memasuki dunia kerja. Siswa SMK adalah orang yang diharapkan menjadi tenaga kerja dan profesional siap kerja di dunia industri. Sebagai bagian dari sistem pendidikan nasional, Sekolah Menengah Kejuruan merupakan pendidikan yang mengutamakan pengembangan keterampilan siswa untuk bekerja dalam bidang tertentu, kemampuan menyesuaikan diri dengan lingkungan kerja, mengidentifikasi peluang kerja, dan berkembang untuk masa depan. SMK juga berupaya menghasilkan lulusan yang memiliki kompetensi yang sesuai dengan tuntutan dunia usaha dan industri, serta mampu bersaing secara global.

##### **2.1.2 Sistem Informasi Akademik**

Menurut Widiati (2019) Sistem Informasi Akademik (SIA) adalah sebuah sistem informasi yang dirancang khusus untuk mengelola data akademik pada suatu institusi pendidikan seperti sekolah atau perguruan tinggi. Sistem ini mengumpulkan, menyimpan, memproses, dan mengelola data terkait siswa, guru, mata pelajaran, jadwal pelajaran, nilai dan akademik lainnya.

### 2.1.3 Website

*Website* adalah halaman atau kumpulan halaman yang dapat diakses melalui internet dan dilihat pada perangkat seperti komputer, laptop, tablet atau *smartphone*. *Website* terdiri dari berbagai elemen seperti teks, gambar, video dan elemen interaktif lainnya yang digunakan untuk menyampaikan informasi, produk atau layanan kepada pengguna (Arif et al, 2023).

### 2.1.4 Database (Basis Data)

*Database* adalah kumpulan informasi yang disimpan secara sistematis dan terorganisir dalam sistem komputer. Informasi yang tersimpan dalam *database* terdiri dari informasi atau fakta yang penting dan dapat diolah sesuai dengan kebutuhan organisasi atau individu. Basis data digunakan untuk menyimpan, mengatur, dan memproses informasi secara efektif dan efisien (Al'asyari, 2023).

### 2.1.5 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman untuk mengembangkan aplikasi web dinamis. PHP adalah bahasa skrip sisi server yang mampu memproses dan membangun halaman web secara dinamis di sisi server sebelum mengirimkannya ke browser pengguna (Widia dan Asriningtias, 2021).

### 2.1.6 Laravel

Laravel adalah sebuah *framework* web berbasis PHP yang *open-source* dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola



MVC. Struktur pola MVC pada laravel sedikit berbeda pada struktur pola MVC pada umumnya. Di laravel terdapat *routing* yang menjembatani antara *request* dari *user* dan *controller*. Jadi *controller* tidak langsung menerima *request* tersebut (Sari et al, 2020).

## 2.2 Penelitian Terdahulu

Adapun rujukan jurnal dari penelitian terdahulu yang digunakan sebagai literatur dalam penelitian ini pada tabel 2.1.

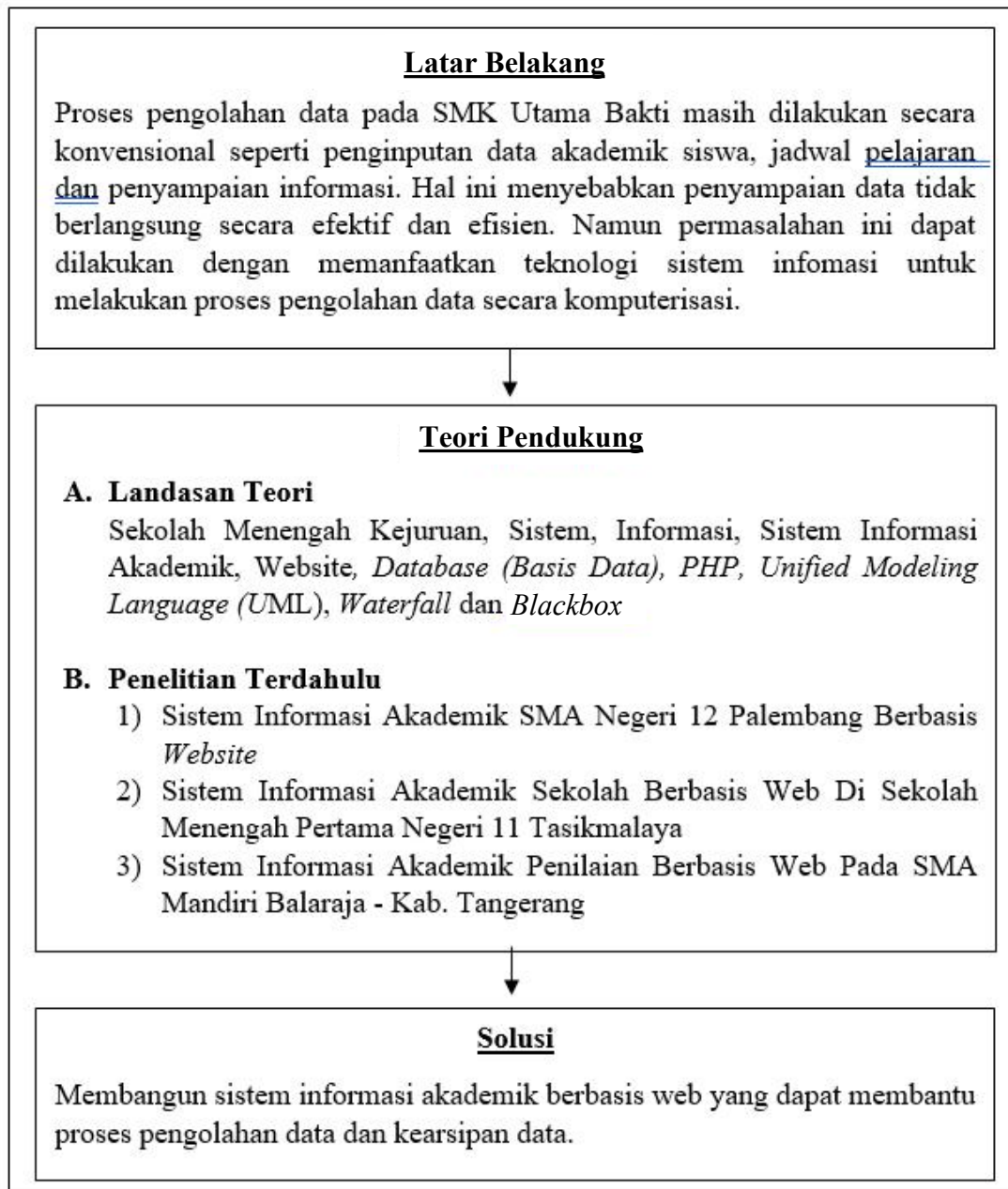
**Tabel 2.1 Penelitian Terdahulu**

No.	Nama Peneliti	Judul Penelitian	Hasil
1	Alfiati / 2020	Sistem Informasi Akademik SMA Negeri 12 Palembang Berbasis <i>Website</i> .	Hasil dari Sistem informasi akademik yang dibangun memberikan kemudahan dalam pengaksesan data dan informasi akademik.
2	Nugraha, Agus Ramdhani Pramukasari, Gati Sumaryana, Yusuf./ 2019.	Sistem Informasi Akademik Sekolah Berbasis Web Di Sekolah Menengah Pertama Negeri 11 Tasikmalaya. Jurnal Manajemen Informatika (JUMIKA). ISSN: 2541-6316.	Hasil dari penelitian ini menghasilkan sistem yang terkomputerisasi dan memudahkan proses pencarian data.
1	Yulianto / 2018.	Sistem Informasi Akademik Penilaian Berbasis	Hasil dari penelitian ini yaitu sistem yang mempermudah dalam

No.	Nama Peneliti	Judul Penelitian	Hasil
		Web Pada SMA Mandiri Balaraja - Kab. Tangerang. Techno Nusa Mandiri, ISSN: 2527-676X.	proses akademik dan laporan penilaian siswa. dengan sistem ini data dapat di akses tidak terbatas jarak dan waktu.

### 2.3 Kerangka Pemikiran

Kerangka pemikiran dalam sistem informasi akademik SMK Utama Bakti Palembang. Kerangka penelitian ini dapat dilihat pada gambar 2.1.



**Gambar 2.1 Kerangka Penelitian**

## **2.4 Objek Penelitian**

### **2.4.1 Sejarah Perusahaan**

Sejarah Yayasan SMK Utama Bakti Palembang dimulai pada tahun 1987, ketika Yayasan Pendidikan Utama Bakti Palembang didirikan. Pada tahun pelajaran 1986/1987, STM Utama Bakti didirikan di bawah naungan yayasan tersebut. Dalam rangka memimpin yayasan tersebut, pada awalnya, kepemimpinan dipegang oleh individu yang masih dalam dinas aktif. Namun, kemudian kepemimpinan yayasan diserahkan kepada Drs. Iskandar Gazali sebagai ketua lembaga. Peran individu ini terutama melibatkan memberikan pengarahan dan petunjuk terhadap pengembangan yayasan.

Pendirian STM Utama Bakti di tahun pelajaran 1986/1987 merupakan tonggak awal dalam perjalanan yayasan ini. Dalam kurun waktu ini, fokus utama adalah memastikan bahwa proses belajar mengajar di STM Utama Bakti berjalan baik, serta disiplin guru dan murid terjaga dengan baik. Sarana dan prasarana pendidikan juga menjadi perhatian, dengan usaha untuk memastikan kondisi yang memadai bagi kegiatan belajar mengajar.

Tidak lama setelah didirikan, STM Utama Bakti mulai menunjukkan perkembangan yang positif. Tiap tahun, dimulai sejak tahun ajaran 1988/1989, sekolah ini berhasil meluluskan siswanya dengan hasil yang memuaskan. Pensiunan pertama kali terjadi pada 1 Januari 1993, dan pada

saat itu, individu yang pensiun tersebut langsung terlibat secara aktif dalam pengelolaan dan pengembangan STM Utama Bakti.

Pengelolaan dan pengembangan STM Utama Bakti semakin ditingkatkan oleh individu yang pensiun tersebut setelah tahun 1993. Pada tahun 1996, sekolah ini mencapai pencapaian penting dengan memperoleh akreditasi "DISAMAKAN" dari Direktorat Jendral Pendidikan dan Kebudayaan. Hal ini memberi sekolah wewenang untuk menyelenggarakan ujian sendiri, yang menunjukkan kualitas pendidikan yang terus meningkat.

Sasaran pendidikan yang diinginkan oleh STM Utama Bakti melibatkan pengembangan siswa yang beriman dan bertaqwa (IMTAQ), serta memiliki ilmu pengetahuan dan keterampilan (IPTEK). Etos kerja yang tinggi juga menjadi salah satu nilai yang diupayakan dalam pendidikan di sekolah ini.

Melalui kerja keras dan komitmen dari kepala sekolah, staf, dan semua pihak terkait, tujuan pendidikan yang diinginkan secara bertahap tercapai. Pada tahun 2006, prestasi sekolah semakin diakui ketika SMK Utama Bakti Palembang berhasil meraih peringkat Akreditasi A+ (Amat Baik) dalam program keahlian teknik pemanfaatan tenaga listrik dan program teknik mekanik motor. Hal ini menegaskan kualitas pendidikan yang terus meningkat di sekolah ini.

Seiring berjalannya waktu, pengelolaan dan pengembangan SMK Utama Bakti Palembang terus berlanjut. Pada setiap tahap, tantangan baru

muncul, seperti mempertahankan peringkat Akreditasi A+ yang telah dicapai. Meskipun sudah pensiun, individu yang terlibat dalam pengelolaan sekolah tetap melibatkan diri dengan antusiasme dan komitmen. Setiap harinya, waktu dihabiskan untuk mengurus yayasan dan sekolah, dengan fokus pada pengembangan fasilitas, kesejahteraan guru, dan hal-hal lain yang berkontribusi pada peningkatan kualitas pendidikan.

Demikianlah, sejarah Yayasan SMK Utama Bakti Palembang menggambarkan perjalanan pendidikan yang dimulai dari pendirian STM Utama Bakti hingga pencapaian prestasi yang diakui dan berkelanjutan. Dengan tekad dan semangat untuk menghasilkan lulusan yang berkualitas, yayasan ini terus berkembang dalam rangka mencapai tujuan pendidikan yang diinginkan.

#### **2.4.2 Visi dan Misi**

##### **a. Visi**

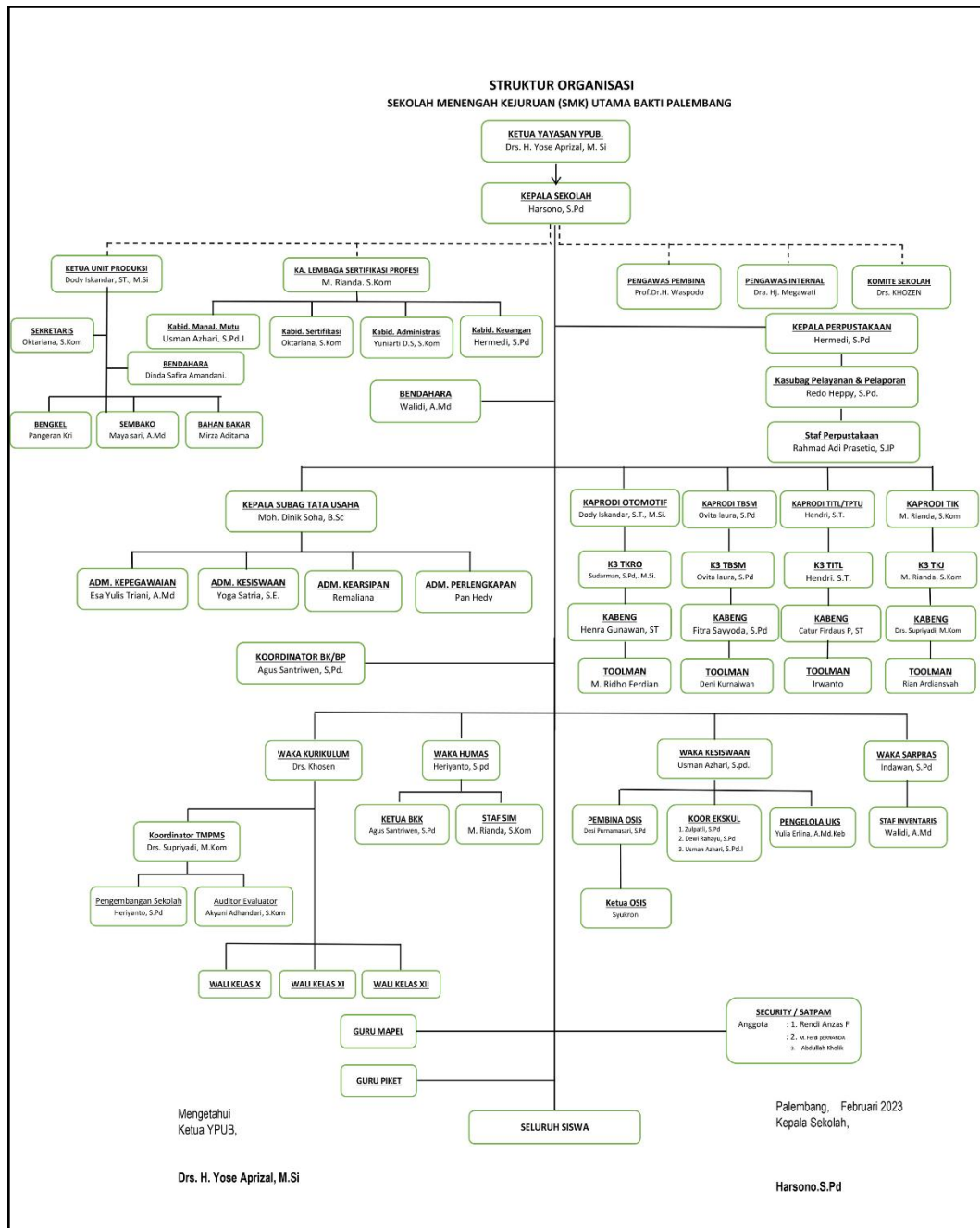
Terwujudnya sekolah menengah kejuruan (SMK) Utama Bakti Palembang berstandar nasional pada tahun 2020 dan terciptanya lulusan yang dapat bersaing di pasar global dengan memiliki iman dan taqwa, penguasaan teknologi dan berjiwa enterpreneur.

##### **b. Misi**

1. Melaksanakan pendidikan dan pengajaran secara konsisten dan penuh tanggung jawab dengan berpedoman pada 8 Standar Nasional Pendidikan.

2. Melaksanakan pendidikan secara optimal, dibidang keagamaan dan kegiatan ekstrakurikuler.
3. Melaksanakan pendidikan dan pengajaran untuk menciptakan sumber daya manusia yang siap kerja, berkarakter bangsa, berwawasan lingkungan hidup dan berjiwa enterprenur.
4. Melaksanakan budaya mutu sekolah berpedoman disiplin, peduli lingkungan/kebersihan, melayani dengan tulus dan ikhlas, produktif, inovatif dan melaksanakan 5S (Senyum, Salam, Sapa, Sopan dan Santun).
5. Menerapkan pembelajaran berbasis teknologi informasi dan komunikasi.
6. Memberikan layanan pendidikan yang prima dan berkualitas.

### 2.4.3 Struktur Organisasi



Gambar 2. 2 Struktur Organisasi



#### **2.4.4 Uraian Kegiatan dan Wewenang**

##### **A. Kepala Sekolah**

- 1) Merancang dan menetapkan program kerja tahunan berdasarkan masukan setiap wakil kepala sub bagian tata usaha, kaprodi dan persetujuan dari pihak lembaga.
- 2) Membuat dan mengkoordinir pelaksanaan program peningkatan kualitas institusi dan SDM.
- 3) Mengkoordinir dan memonitor pengelolaan sekolah yang meliputi :
  - a. Sub Bagian Tata Usaha
  - b. Urusan Kurikulum
  - c. Urusan Sarana dan Prasarana
  - d. Urusan Kesiswaan
  - e. Urusan Hubungan Masyarakat
- 4) Penyusunan rencana strategis (Renstra).
- 5) Penyusunan rencana anggaran.
- 6) Meyelenggarakan rapat kerja tiap semester.
- 7) Meningkatkan kesejahteraan pegawai, karyawan dan tenaga fungsional.
- 8) Melaksanakan tugas yang ditetapkan atasan langsung (Dinas Pendidikan Provinsi Sumsel dan Kementrian Pendidikan dan Kebudayaan) dengan penuh tanggung jawab.
- 9) Melakukan supervisi terhadap pelaksanaan tugas pegawai, karyawan dan tenaga fungsional.

##### **B. Kepala Sub Bagian Tata Usaha**

C. Wakil Kepala Sekolah Bidang Hubungan Masyarakat dan Hubungan Industri

- 1) Mewakili kepala sekolah dalam bidang humas/hubin, jika kepala sekolah tidak berada ditempat atau berhalangan.
- 2) Menetapkan program kerja bidang humas/hubin.
- 3) Mengusulkan rancangan penempatan PSG siswa kelas XI kepada kepala sekolah.
- 4) Mengusulkan rancangan kunjungan industri kelas X dan XI.
- 5) Menyampaikan ide, gagasan, pertimbangan dan saran kepada kepala sekolah dalam rangka meningkatkan kualitas unit kerja.
- 6) Mengatur pendistribusian tugas staf bidang Humas/Hubin.
- 7) Memberikan/mengusulkan pemberian *reward* dan *punishment* terhadap staf dan siswa sesuai peraturan yang berlaku.
- 8) Mengajukan kandidat untuk mengisi suatu jabatan dalam rangka promosi.
- 9) Memberikan rekomendasi kepada lulusan untuk mengikuti seleksi tenaga kerja sesuai kebutuhan pelanggan.
- 10) Melaksanakan kerjasama dengan DU/DI dalam rangka peningkatan daya serap alumni.
- 11) Menyampaikan laporan kegiatan rutin dan tahunan kepada kepala sekolah.

D. Wakil Kepala Sekolah Bidang Sarana dan Prasarana

- 1) Mewakili kepala sekolah dan membantu tugas kepala sekolah baik kedalam maupun keluar, bila kepala sekolah berhalangan.
- 2) Menjabarkan instruksi kepala sekolah kepada karyawan maupun siswa agar dapat lebih dimengerti dan dilaksanakan.
- 3) Membantu Kepala Sekolah dalam hal membuat/menyusun program kerja
- 4) Mengawasi pelaksanaan program kerja sesuai bidangnya.
- 5) Menyusun program kerja tahunan bidang ketatausahaan (Administrasi, kerumahtanggaan, kepegawaian dan keuangan).
- 6) Menetapkan sasaran mutu bidang ketatausahaan.
- 7) Mengelola administrasi sekolah.
- 8) Mengelola admistrasi ketenagaan.
- 9) Mengelola administrasi keuangan.
- 10) Mengelola/melakukan pengawasn sasaran mutu terhadap staf tata usaha dalam menjalankan tugas sesuai bidang kerjanya untuk menjamin keterlaksanaan program.
- 11) Melaksanakan pelayanan prima terhadap *stakeholder*.
- 12) Melakukan pembinaan staf bidang tata usaha.
- 13) Membuat laporan pelaksanaan tugas setiap semester untuk disampaikan dan dipertanggungjawabkan kepada Kepala Sekolah.
- 14) Dalam menjalankan tugasnya berkoordinasi dengan semua wakil kepala sekolah.
- 15) Menjalankan tugas khusus yang diberikan kepala sekolah.

#### E. Wakil Kepala Sekolah Bidang Kurikulum

- 1) Mewakili kepala sekolah dalam bidang penyelenggaraan pendidikan selama kepala sekolah tidak berada ditempat tugas.
- 2) Mewakili kepala sekolah dan membantu tugas kepala sekolah baik kedalam maupun keluar, bila kepala sekolah berhalangan.
- 3) Menjabarkan instruksi kepala sekolah kepada guru dan karyawan maupun siswa agar dapat lebih dimengerti dan dilaksanakan.
- 4) Membantu Kepala Sekolah dalam hal membuat/menyusun program kerja.
- 5) Mengawasi pelaksanaan tugas guru piket, wali kelas dan petugas-petugas yang lain.
- 6) Mengawasi pelaksanaan program kerja sesuai bidangnya.
- 7) Membuat laporan pelaksanaan tugas setiap semester untuk disampaikan/dipertanggung jawabkan kepada kepala sekolah.
- 8) Bertanggung jawab terhadap pengembangan sesuai bidang tugasnya dalam menjalankan tugasnya berkoordinasi.

#### F. Wakil Kepala Sekolah Bidang Kesiswaan

- 1) Mewakili kepala sekolah dalam bidang kesiswaan selama kepala sekolah tidak berada di tempat/berhalangan hadir.
- 2) Mewakili kepala sekolah dan membantu tugas kepala sekolah baik ke dalam maupun ke luar, bila kepala sekolah berhalangan.
- 3) Menjabarkan instruksi kepala sekolah kepada guru dan karyawan maupun siswa agar lebih dimengerti dan dilaksanakan.

- 4) Membantu kepala sekolah dalam hal membuat/menyusun program kerja.
- 5) Mengawasi pelaksanaan tugas guru piket, wali kelas dan petugas-petugas yang lain.
- 6) Mengawasi pelaksanaan program kerja sesuai bidangnya.
- 7) Membuat laporan pelaksanaan tugas setiap semester untuk disampaikan/dipertanggungjawabkan terhadap pengembangan sesuai bidang tugasnya.
- 8) Bertanggung jawab terhadap pengembangan sesuai bidang tugasnya.
- 9) Melakukan kerjasama dengan kepala sub bagian tata usaha dan waka lain menjalankan tugas.
- 10) Menjalankan tugas khusus yang dibebankan oleh kepala sekolah.
- 11) Tugas khusus wakil kepala sekolah urusan kesiswaan.
- 12) Menyusun dan melaksanakan program di bidang kesiswaan sesuai persetujuan kepala sekolah.
- 13) Mengelola kegiatan OSIS dan Ekstrakurikuler sebagai wahana pengembangan minat, bakat dan pembinaan mental/kedisiplinan serta Imtaq-iptek.
- 14) Merencanakan dan melaksanakan penerimaan peseta didik baru, diklat kedisipilinan, *Outbound*, AMT, ESQ.
- 15) Membuat laporan *incidental* maupun rutin terhadap kegiatan di bidang kesiswaan.

## BAB III

### METODE PENELITIAN

#### 3.1 Lokasi dan Waktu Penelitian

##### 3.1.1 Lokasi Penelitian

Tempat penelitian dilaksanakan pada Sekolah Menengah Kejuruan (SMK) Utama Bakti Kota Palembang Provinsi Sumatera Selatan, yang beralamatkan di Jl. STM Utama Bakti Lebong Siareng Palembang Provinsi Sumatera Selatan.

##### 3.1.2 Waktu Penelitian

Jadwal dilakukanya penelitian ini dapat dilihat pada tabel 3.1.

**Tabel 3.1 Waktu Penelitian**

No	Kegiatan	April				Mei				Juni				Juli				Agustus				
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
1	<i>Requirement</i>																					
	Wawancara																					
	Observasi																					
	Studi Pustaka																					
2	<i>Design</i>																					
3	<i>Implementations</i>																					
4	<i>Verification</i>																					
5	<i>Maintenance</i>																					

#### 3.2 Jenis Data

##### 3.2.1 Data Primer

Menurut (Fauzi et al., 2022), data primer adalah data yang didapatkan dari hasil wawancara langsung dengan kepala sekolah SMK

Utama Bakti Palembang. Data primer yang peneliti gunakan adalah data dari hasil wawancara mengenai permasalahan dan kendala yang terjadi pada SMK Utama Bakti Palembang.

### **3.2.2 Data Sekunder**

Menurut (Asset et al., 2020), data sekunder adalah data yang diperoleh dari bahan-bahan yang tersedia di buku-buku, majalah, jurnal, bursa efek Indonesia (BEI) dan sumber lainnya yang secara tidak langsung berhubungan dengan SMK Utama Bakti Palembang berupa sejarah sekolah, struktur organisasi, data siswa dan data nilai.

## **3.3 Teknik Pengumpulan Data**

### **3.3.1 Wawancara (*interview*)**

Menurut (Herdayati et al. 2019), wawancara adalah suatu metode atau teknik komunikasi langsung antara peneliti atau pewawancara dengan responden atau subjek penelitian. Wawancara digunakan untuk memperoleh informasi, pemahaman, dan perspektif dari responden mengenai topik atau masalah yang diteliti. Wawancara dapat dilakukan dalam bentuk tanya jawab atau diskusi yang terstruktur atau tidak terstruktur, tergantung pada tujuan penelitian dan jenis data yang ingin diperoleh. Wawancara juga melibatkan penggunaan keterampilan komunikasi dan empati untuk menciptakan suasana yang nyaman bagi responden sehingga mereka dapat memberikan tanggapan yang jujur dan mendalam.

Dalam penelitian ini dilakukan wawancara bersama Bapak Drs. Khozen selaku wakil kepala sekolah bidang kurikulum SMK Utama Bakti Palembang. Dari hasil wawancara tersebut didapati belum adanya sistem yang membantu proses pengolahan data dan penyampaian informasi secara efisien dalam sistem akademik SMK Utama Bakti Palembang.

### **3.3.2 Observasi (Pengamatan)**

Observasi adalah aktivitas yang memiliki tujuan untuk mendapatkan informasi atau data dengan cara mengamati dan mencatat secara sistematis terhadap fenomena yang menjadi fokus pengamatan. (Jaelani et al., 2020).

Dalam hal ini penulis mengadakan pengamatan secara langsung mengenai alur sistem akademik jadwal pelajaran dan penilaian siswa.

### **3.3.3 Studi Pustaka**

Studi pustaka berkaitan langsung dengan kajian teoritis dan beberapa referensi yang tidak akan lepas dari literatur-literatur ilmiah. (Putri, 2019).

Dalam hal ini peneliti mencari referensi dari jurnal sebagai acuan dalam proses penelitian ini.

## **3.4 Alat Pengembangan Sistem**

Untuk mengembangkan sistem yang dibangun penulis untuk menggunakan metode *waterfall* dan *Unified Modeling Language (UML)* sebagai alat pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek UML. Pada penelitian ini *diagram* yang



digunakan terdiri dari *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*.

### 3.4.1 Model Proses

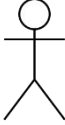
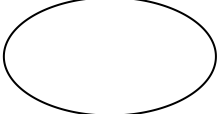
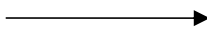

#### 3.4.1.1 *Unified Modeling Language (UML)*


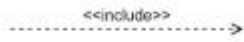
Menurut Prihandoyo (2018), *Unified Modeling Language (UML)* merupakan sebuah model perancangan sistem yang mempunyai kelebihan dapat memudahkan developer sistem dalam merancang sistem yang akan dibuat karena sifatnya yang berorientasikan pada objek. Terdapat beberapa *diagram UML* yang sering digunakan dalam pengembangan sebuah sistem, yaitu :

##### a. *Use Case Diagram*

*Use case diagram* merupakan alat pemodelan yang digunakan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. Dengan menggunakan *use case diagram*, kita dapat mengidentifikasi fungsi-fungsi yang ada di dalam sistem informasi dan mengidentifikasi siapa saja yang berhak atau memiliki akses untuk menggunakan fungsi-fungsi tersebut. Simbol – simbol *use case diagram* pada tabel 3.2.

Tabel 3. 2 *Use Case Diagram*



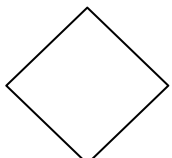
No	Simbol	Nama	Keterangan
1	 Actor	Aktor	Menggambarkan pengguna-pengguna yang akan menggunakan sistem.
2		<i>Use Case</i>	menggambarkan proses-proses yang terjadi pada sistem baru.
3		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) dalam <i>use case diagram</i> menggambarkan bahwa satu fungsi <i>use case</i> merupakan fungsi yang lebih umum daripada yang lain.
4		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> terjadi ketika aktor berpartisipasi dalam <i>use case</i> atau ketika <i>use case</i> memiliki interaksi dengan aktor

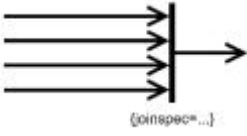

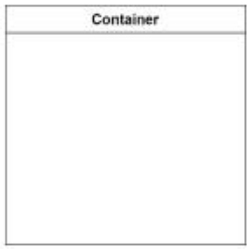
No	Simbol	Nama	Keterangan
5		<i>Extend</i>	<p>Relasi <i>use case</i> tambahan mengacu pada penambahan satu <i>use case</i> ke <i>use case</i> lainnya, di mana <i>use case</i> yang ditambahkan memiliki kemampuan untuk berdiri sendiri meskipun memiliki nama depan yang sama dengan <i>use case</i> yang ada.</p>
6		<i>Include</i>	<p>Relasi <i>use case</i> tambahan terjadi ketika sebuah <i>use case</i> ditambahkan ke <i>use case</i> lainnya, yang memungkinkan <i>use case</i> yang ditambahkan berfungsi secara mandiri meskipun memiliki nama depan yang sama dengan <i>use case</i> yang ada.</p>

b. *Activity Diagram*

*Activity diagram* adalah representasi visual yang menggambarkan aliran kerja atau aktivitas dari suatu sistem atau proses bisnis simbol-simbol yang digunakan ada pada tabel 3.3.

**Tabel 3. 3 *Activity Diagram***

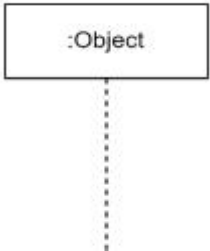

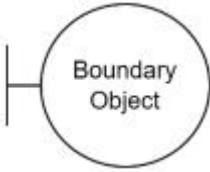
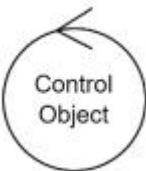
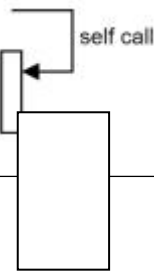

No	Simbol	Nama	Keterangan
1		<i>Initial</i>	Status awal aktivitas dalam sebuah sistem ditunjukkan dalam <i>diagram</i> aktivitas, dimana setiap aktivitas memiliki status awal yang spesifik.
2		<i>Activity</i>	Biasanya, aktivitas yang dilakukan oleh sistem dimulai dengan kata kerja dalam suatu proses
3		<i>Decision</i>	Percabangan asosiasi adalah saat dua atau lebih aktivitas digabungkan menjadi satu dalam suatu <i>diagram</i> .

No	Simbol	Nama	Keterangan
4		<i>Join</i>	Penggabungan asosiasi terjadi ketika terdapat lebih dari satu aktivitas yang digabungkan menjadi satu dalam suatu <i>diagram</i>
5		<i>Final</i>	<i>Diagram</i> aktivitas memiliki satu status akhir yang menunjukkan tindakan terakhir yang dilakukan oleh sistem
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

### c. *Sequence Diagram*

*Sequence diagram* adalah representasi visual yang menggambarkan perilaku objek dalam sebuah *use case* dengan menjelaskan waktu hidup objek dan pesan yang dikirimkan serta diterima antara objek. Tabel 3.4 berisi simbol-simbol yang digunakan dalam *sequence diagram*.


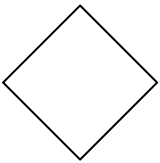
Tabel 3. 4 *Sequence Diagram*

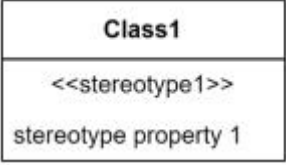
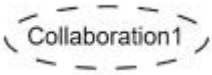


No	Simbol	Nama	Keterangan
1		<i>Life Line</i>	Komponen yang terhubung dengan objek digambarkan dengan garis putus.
2		<i>Entity Class</i>	Gambaran sistem sebagai landasan dalam menyusun basis data
3		<i>Boundary Class</i>	Menangani komunikasi antar lingkungan sistem
4		<i>Control Class</i>	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
5		<i>Recursive</i>	Pesan untuk dirinya
6		<i>A Focus of Control</i>	Mewakili proses durasi aktivasi sebuah proses

d. *Class Diagram*

*Class diagram* adalah representasi hubungan antara kelas-kelas dan penjelasan rinci dari setiap kelas dalam model desain suatu sistem. *Diagram* ini juga menunjukkan aturan-aturan dan tanggung jawab entitas yang mengatur perilaku sistem. Tabel 3.5 berisi simbol-simbol yang digunakan dalam *class diagram*.

**Tabel 3.5 *Class Diagram***

No	Simbol	Nama	Keterangan
1		<i>Generalization</i>	Dalam hubungan ini, objek anak ( <i>descendant</i> ) memperoleh dan mewarisi perilaku serta struktur data dari objek induk ( <i>ancestor</i> ) yang berada di atasnya.
2		<i>Nary Association</i>	Upaya dilakukan untuk menghindari asosiasi antara lebih dari dua objek.

No	Simbol	Nama	Keterangan
3		<i>Class</i>	Himpunan objek-objek yang memiliki atribut dan operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan tindakan yang ditampilkan oleh sistem yang menghasilkan hasil yang dapat diukur bagi seorang aktor.
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (independen) akan berdampak pada elemen yang bergantung

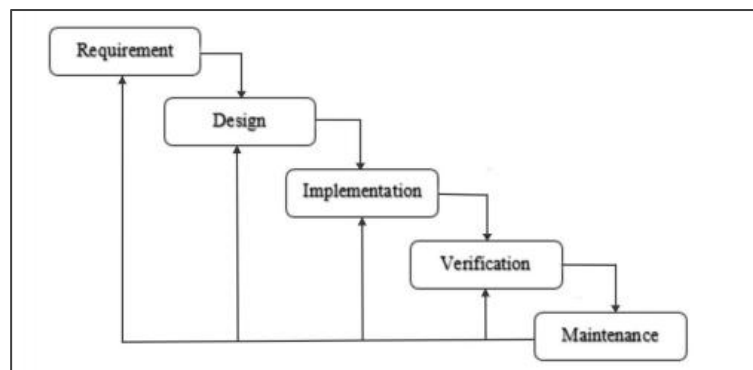


No	Simbol	Nama	Keterangan
			padanya, yaitu elemen yang tidak mandiri.
	_____	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

### 3.5 Metode Pengembangan Sistem

Metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “*Linear Sequential Model*”. Metode *waterfall* merupakan metode pengembangan yang dimulai dengan kebutuhan pengguna yang berlanjut melalui tahapan *requirement*, *design*, *implementations*, *verification*, dan *maitenance* (Wahid, 2020).

Tahapan dari metode *waterfall* dapat dilihat dari gambar di bawah ini



Sumber : (Wahid, 2020)

**Gambar 3.1 Metode *Waterfall***

Dalam hal ini terdapat beberapa tahapan pengembangan *waterfall* antara lain analisis kebutuhan, desain sistem, implementasi sistem, pengujian dan pemeliharaan.

### **1. Requirement**

Pada tahap ini diperlukannya komunikasi yang bertujuan untuk memahami sistem yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei secara langsung. Informasi dianalisis bertujuan untuk mendapatkan data-data yang dibutuhkan oleh pengguna. Fitur yang diperlukan pada sistem informasi akademik ini berupa monitoring data siswa, data guru, jadwal mengajar dan informasi pengumuman. Selain itu sistem yang dibangun juga dapat melakukan pengolahan data, nilai dan siswa.

### **2. Design**

Tahap ini merupakan proses pembuatan desain sistem yang berfungsi untuk memberikan gambaran lengkap mengenai apa saja yang harus di kerjakan dan tahap ini juga dapat membantu pengembang menyiapkan kebutuhan perangkat keras (*Hardware*) dalam pembuatan arsitektur sistem secara keseluruhan. Dalam tahapan ini akan dilakukan pembuatan *diagram* blok, *diagram* aliran data dan spesifikasi antarmuka pengguna (*user interface specification*).

### **3. *Implementation***

Tahapan *implementation* ini merupakan tahap pemrograman. Pembuatan perangkat lunak dikembangkan di program kecil yang biasa disebut sebagai unit. Setiap unit ini yang nantinya akan digabungkan dalam tahap berikutnya. Disamping itu unit akan diuji fungsionalitasnya yang disebut sebagai *unit testing*.

### **4. *Verification***

Dilakukannya verifikasi dan pengujian apakah perangkat lunak yang dikembangkan di program kecil sudah sesuai dengan kriteria sistem yang dibutuhkan.

### **5. *Maintenance***

Pada tahapan terakhir ini, perangkat lunak yang telah dibuat dan dioperasikan oleh pengguna akan dilakukan pemeliharaan. Pemeliharaan ini memungkinkan pengembang melakukan perbaikan yang tidak terdeteksi pada tahapan sebelumnya.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Hasil**

Pengembangan sistem akademik SMK Utama Bakti Palembang dengan menggunakan metode *waterfall* yang meliputi tahapan perancangan, desain, implementasi, dan pengujian, serta memanfaatkan metode *Unified Modeling Language* (UML) untuk membuat perancangan sistem merupakan langkah yang sistematis dan terstruktur. Metode *waterfall* memungkinkan penulis untuk mengevaluasi setiap tahapan dengan cermat sebelum melanjutkan ke tahapan berikutnya, sehingga meminimalkan risiko kesalahan yang mungkin terjadi dalam pengembangan sistem. Dengan pendekatan ini, diharapkan sistem akademik SMK Utama Bakti Palembang dapat ditingkatkan fungsionalitasnya, keandalannya, dan responsivitasnya terhadap kebutuhan sekolah dan pengguna serta menggunakan metode UML untuk membuat perancangan sistem.

##### **4.1.1 Requirement**

Pada tahap ini diperlukannya komunikasi yang bertujuan untuk memahami sistem yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei secara langsung. Informasi dianalisis bertujuan untuk mendapatkan data-data yang dibutuhkan oleh pengguna. Hasil dari tahapan ini yaitu

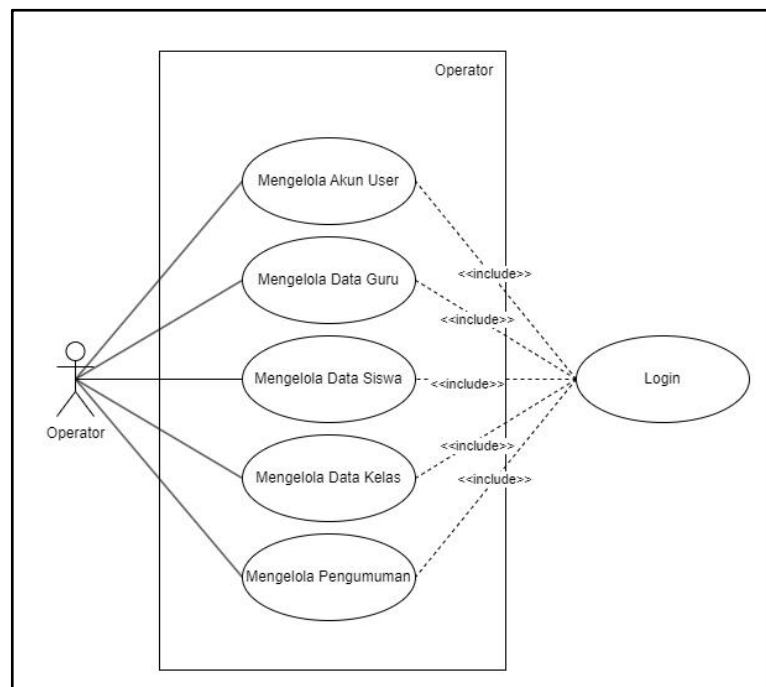
menetapkan fitur yang akan dibuat pada sistem informasi akademik berupa monitoring data siswa, data guru, jadwal mengajar, pengolahan data nilai dan informasi pengumuman. Selain itu didapatkan juga rancangan sistem yang akan dibuat berupa *use case diagram*, *activity diagram* dan *class diagram*. Penjelasan rancangan tersebut dapat dilihat berikut ini.

#### 4.1.1.1 Use Case Diagram

*Use case diagram* merupakan proses yang menjelaskan proses singkat yang terjadi pada sistem yang diusulkan, atau bisa dilihat sebagai berikut ini :

##### 1. Use Case Diagram Operator

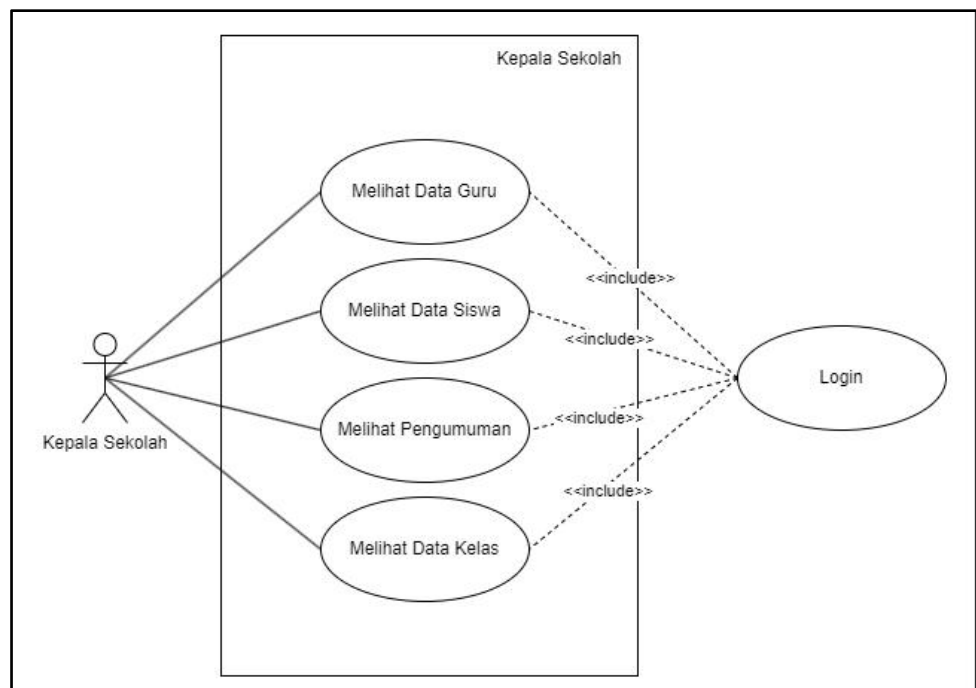
*Use case diagram* ini menjelaskan proses pada sistem yang diusulkan yaitu operator yang dapat mengelola akun *user*, data guru, data siswa, data nilai dan pengumuman yang dapat dilihat pada gambar 4.1.



**Gambar 4.1 Use Case Diagram Operator**

## 2. Use Case Diagram Kepala Sekolah

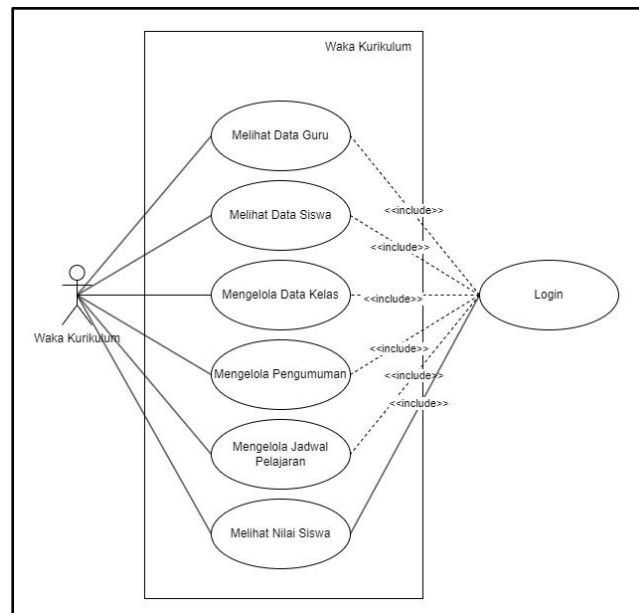
*Use case diagram* ini menjelaskan proses pada sistem yaitu kepala sekolah yang dapat melihat data siswa, guru, kelas dan pengumuman yang dapat dilihat pada gambar 4.2.



**Gambar 4. 2 Use Case Diagram Kepala Sekolah**

## 3. Use Case Diagram Waka Kurikulum

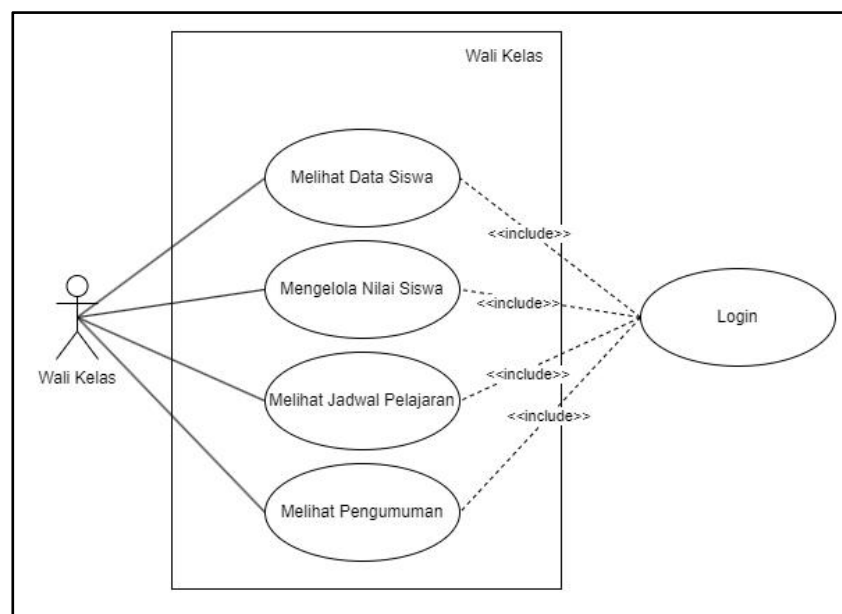
*Use case diagram* ini menjelaskan proses pada sistem yaitu wakil kepala sekolah bidang kurikulum yang dapat melihat data siswa, guru, mengelola kelas, mengelola pengumuman, mengelola jadwal pelajaran dan melihat nilai siswa yang dapat dilihat pada gambar 4.3.



**Gambar 4.3 Use Case Diagram Waka Kurikulum**

#### 4. Use Case Diagram Wali Kelas

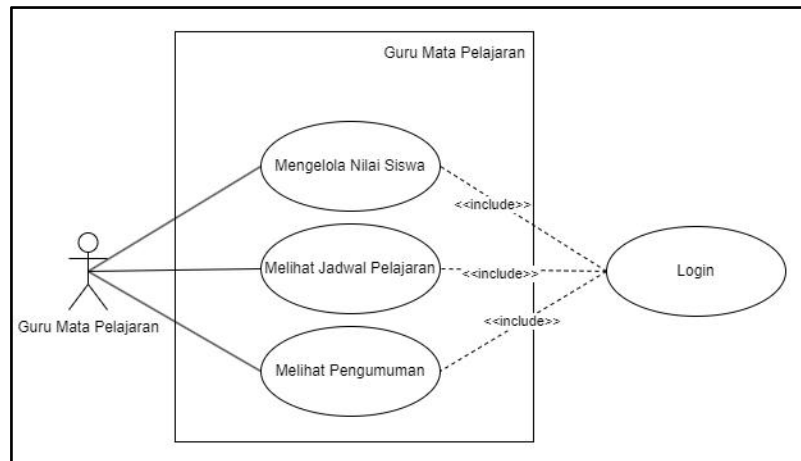
*Use case diagram* ini menjelaskan proses pada sistem yaitu wali kelas yang dapat melihat data siswa, mengelola nilai siswa, melihat jadwal pelajaran, *melihat* pengumuman yang dapat dilihat pada gambar 4.4.



**Gambar 4.4 Use Case Diagram Wali Kelas**

### 5. Use Case Diagram Guru

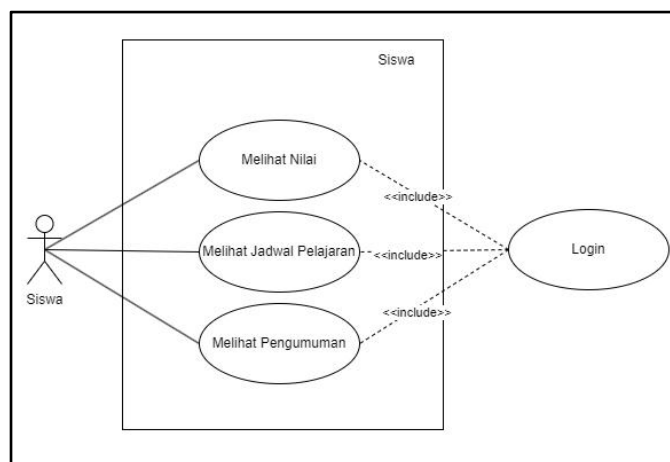
*Use case diagram* ini menjelaskan proses pada sistem yaitu guru yang dapat mengelola nilai siswa, melihat jadwal pelajaran dan melihat pengumuman yang dapat dilihat pada gambar 4.5.



**Gambar 4. 5 Use Case Diagram Guru**

### 6. Use Case Diagram Siswa

*Use case diagram* ini menjelaskan proses pada sistem yaitu siswa yang dapat melihat nilai, jadwal pelajaran dan pengumuman yang dapat dilihat pada gambar 4.6.



**Gambar 4. 6 Use Case Diagram Siswa**

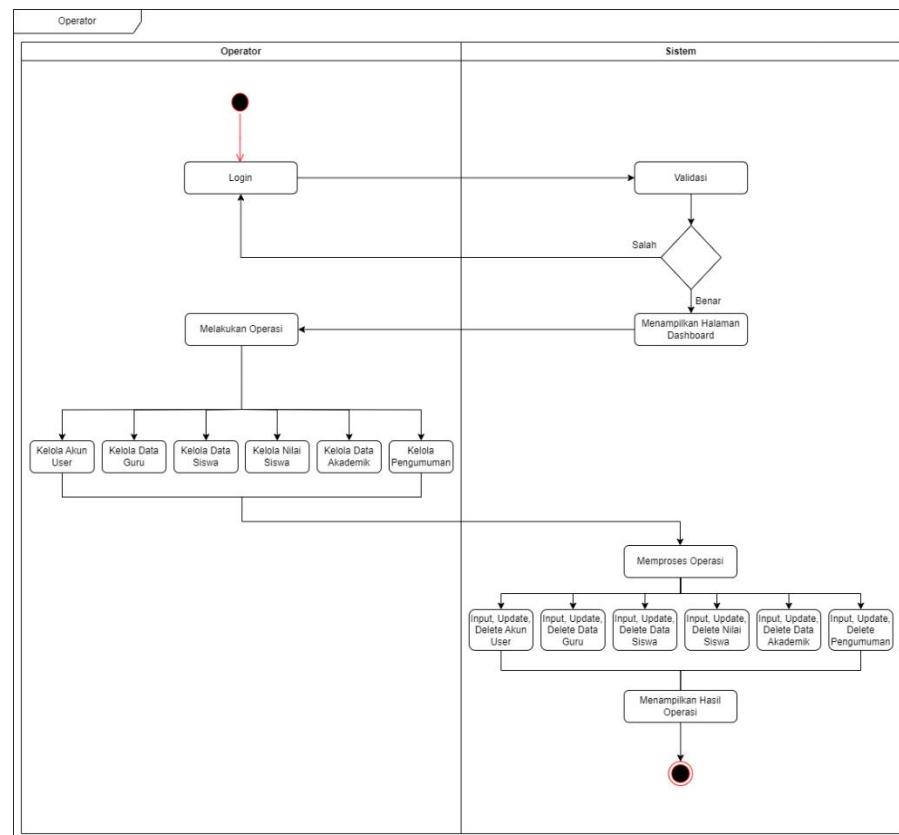


#### 4.1.1.2 *Activity Diagram*

Lalu selanjutnya *activity diagram* yang merupakan *diagram* aktivitas sebagai langkah-langkah yang dilakukan dalam suatu proses kerja atau bisa juga diartikan sebagai alur proses kerja, adapun *activity diagram* pada sistem akademik ini sebagai berikut :

1. *Activity Diagram* Operator

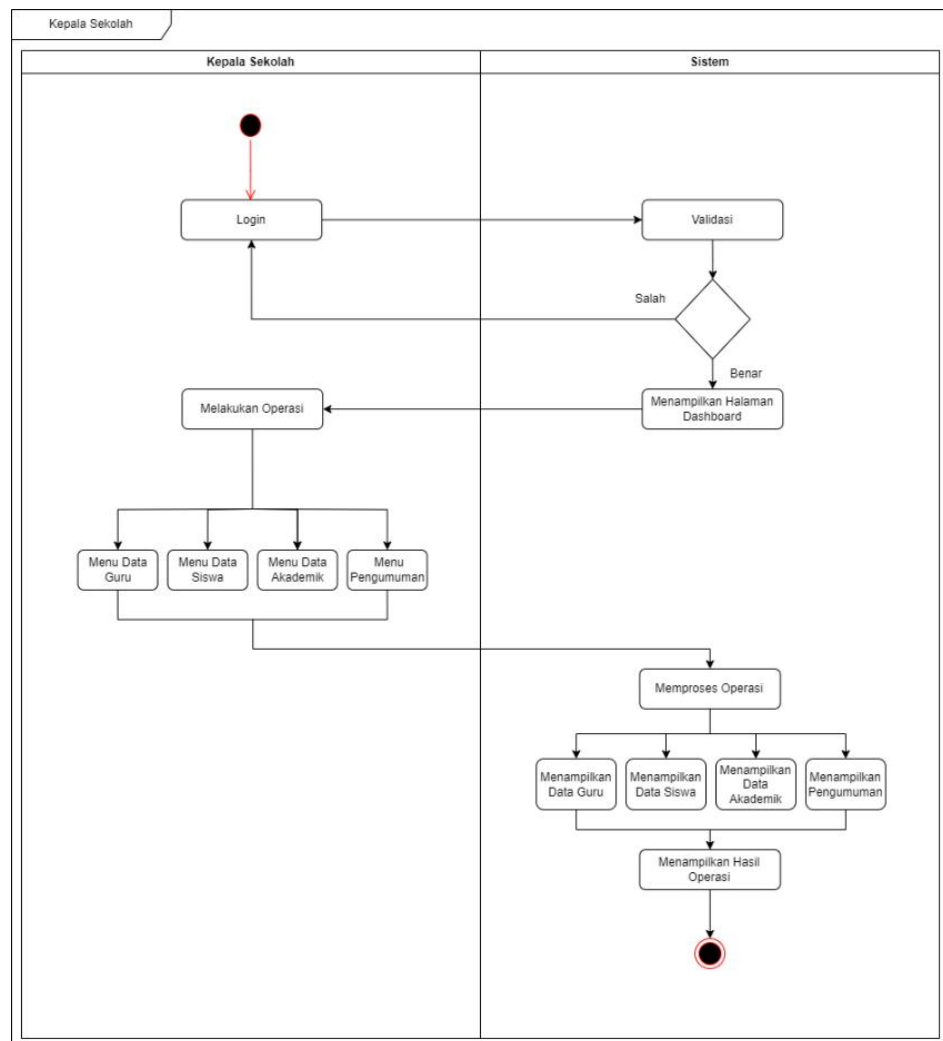
*Activity diagram* pada operator menggambarkan bahwa operator setelah melakukan *login*. Kemudian operator dapat melihat halaman *dashboard* dan menu berupa kelola akun *user*, data siswa, data guru, nilai siswa, data kelas, dan kelola pengumuman serta dapat menambahkan, mengedit dan menghapus seperti pada gambar gambar 4.7



**Gambar 4.7 Activity Diagram Operator**

## 2. Activity Diagram Kepala Sekolah

*Activity diagram* pada operator menggambarkan bahwa kepala sekolah setelah melakukan *login*. Kemudian kepala sekolah dapat melihat halaman *dashboard* dan menu berupa menu data guru, menu data siswa, menu data kelas dan menu pengumuman seperti pada gambar gambar 4.8.

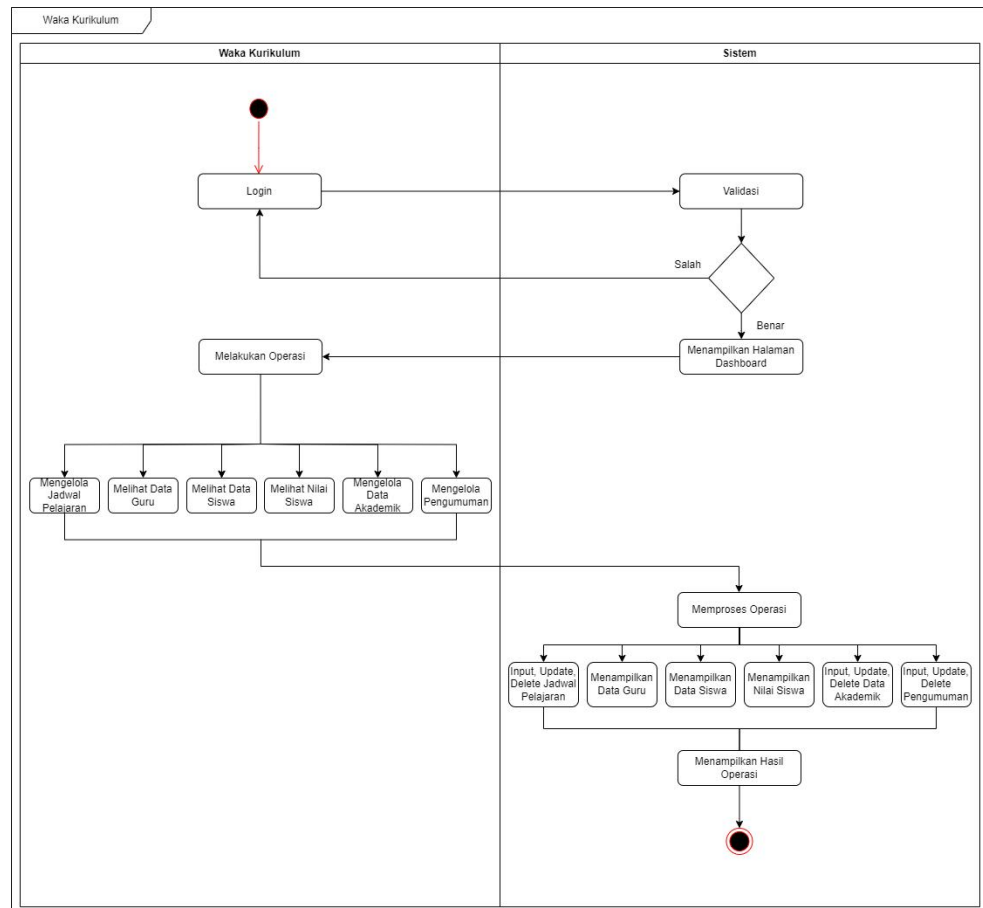


**Gambar 4. 8 Activity Diagram Kepala Sekolah**

### 3. Activity Diagram Waka Kurikulum

*Activity diagram* pada waka kurikulum menggambarkan bahwa waka kurikulum setelah melakukan *login*. Kemudian waka kurikulum dapat melihat halaman *dashboard* dan menu berupa mengelola jadwal Pelajaran melihat data guru, melihat data siswa, melihat nilai siswa, mengelola data

kelas, dan mengelola pengumuman serta dapat menambahkan, mengedit dan menghapus seperti pada gambar gambar 4.9

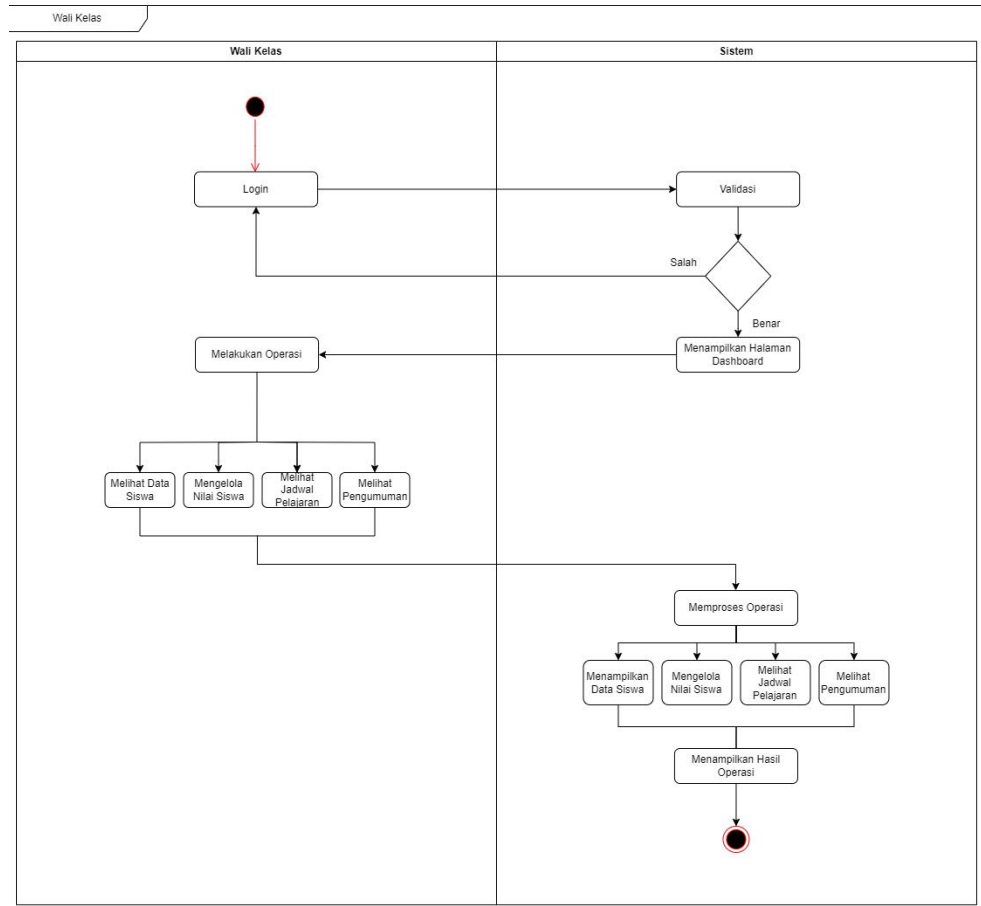


**Gambar 4.9 Activity Diagram Waka Kurikulum**

#### 4. Activity Diagram Wali Kelas

*Activity diagram* pada wali kelas menggambarkan bahwa wali kelas setelah melakukan *login*. Kemudian wali kelas dapat melihat halaman *dashboard* dan menu melihat data siswa, mengelola nilai siswa, melihat

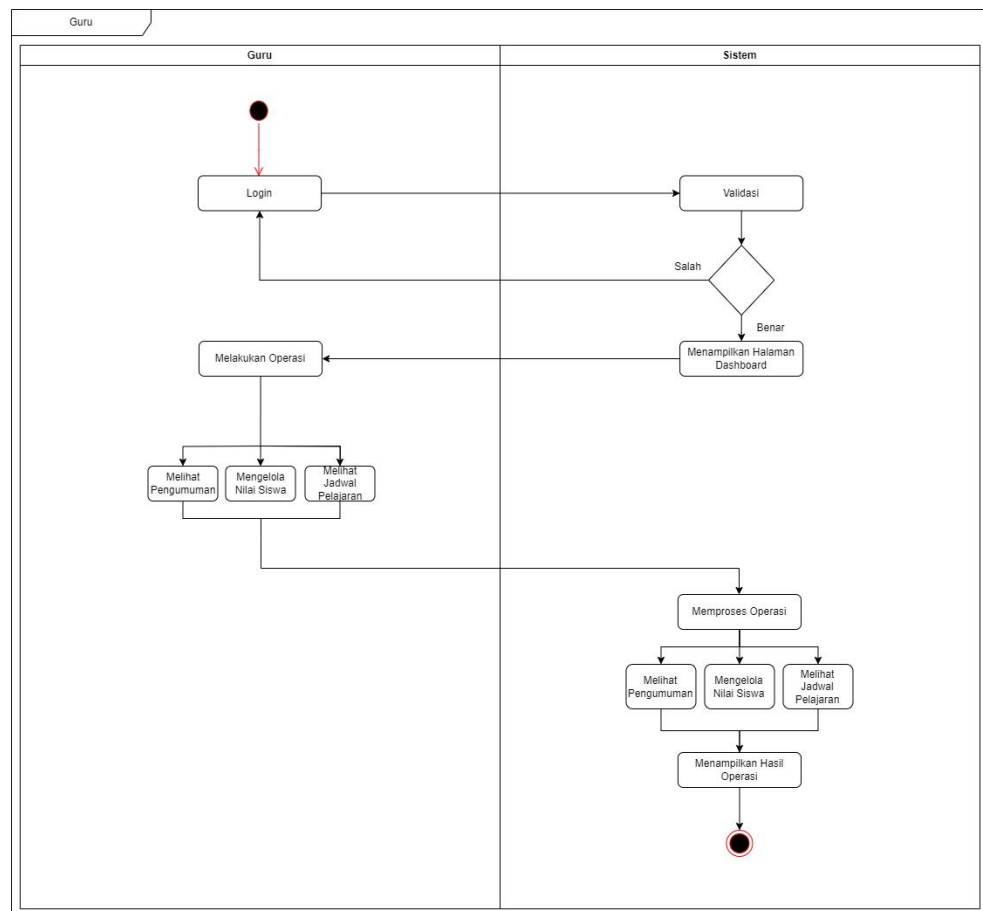
jadwal pelajaran, melihat pengumuman serta memvalidasi nilai seperti pada gambar gambar 4.10



**Gambar 4. 10 Activity Diagram Wali Kelas**

### 5. Activity Diagram Guru

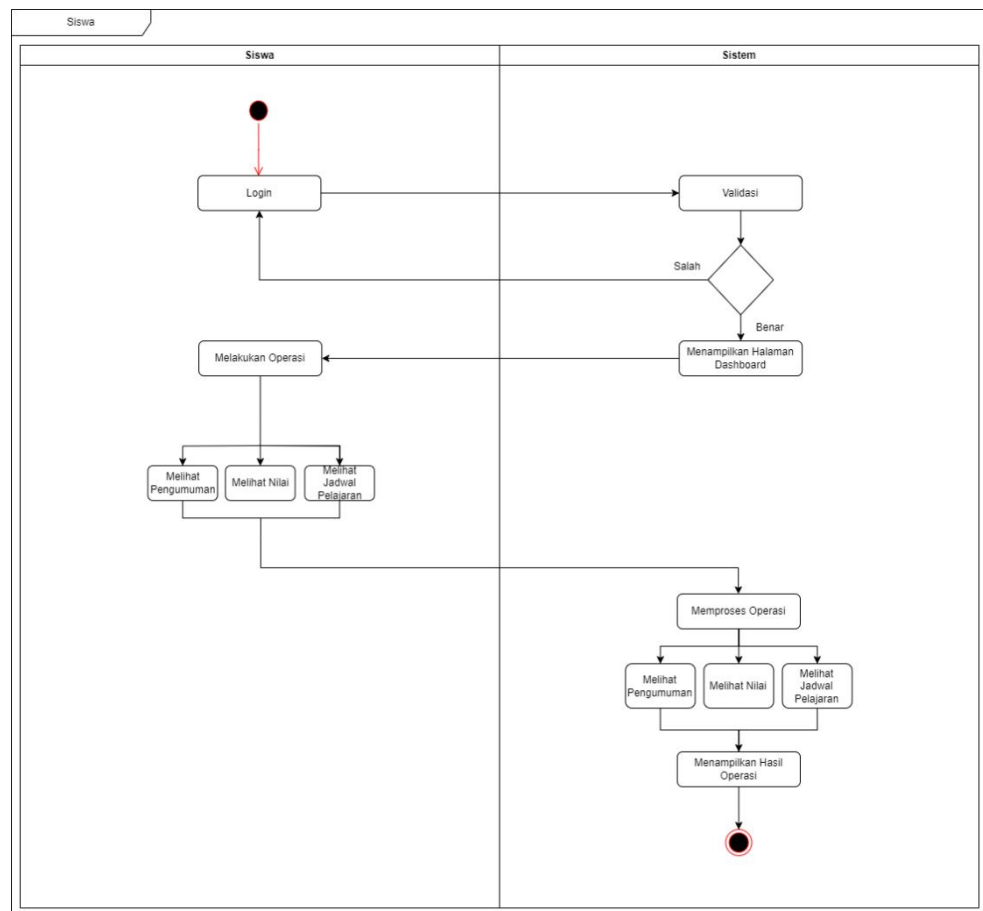
*Activity diagram* pada guru menggambarkan bahwa guru setelah melakukan *login*. Kemudian guru dapat melihat halaman *dashboard* dan menu melihat pengumuman, menginput nilai siswa dan melihat jadwal pelajaran seperti pada gambar gambar 4.11



**Gambar 4. 11 Activity Diagram Guru**

#### 6. Activity Diagram Siswa

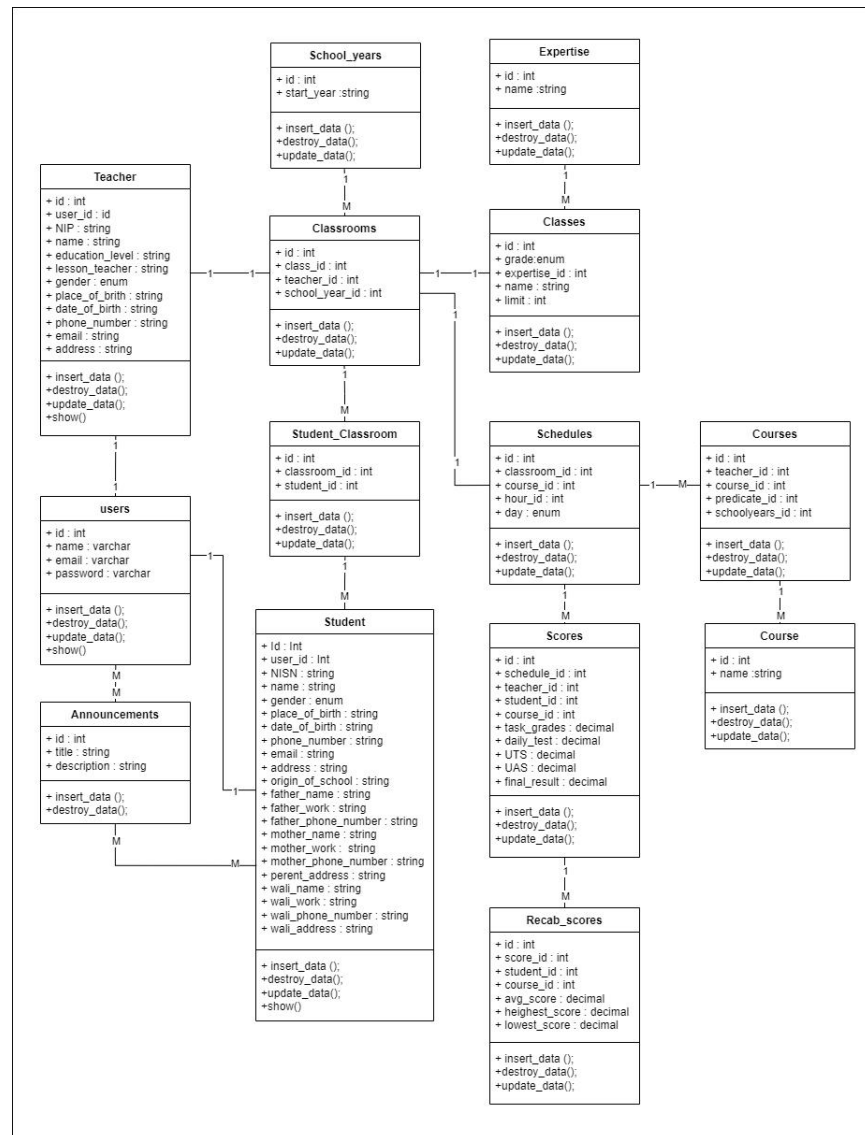
*Activity diagram* pada siswa menggambarkan bahwa siswa setelah melakukan *login*. Kemudian siswa dapat melihat halaman *dashboard* dan menu berupa menu nilai, jadwal pelajaran, dan pengumuman seperti pada gambar gambar 4.12



**Gambar 4. 12 Activity Diagram Siswa**

#### 4.1.1.3 Class Diagram

Selanjutnya merupakan *class diagram* dari Sistem Akademik SMK Utama Bakti Palembang, yang merupakan gambaran dari atribut beserta metode dari SIA untuk *class diagram* bisa dilihat pada Gambar 4.13



**Gambar 4. 13 Class Diagram**

Setelah tahapan perancangan menggunakan UML dimulai dari *use case*, *activity diagram* dan *class diagram*.



## 4.2 Design

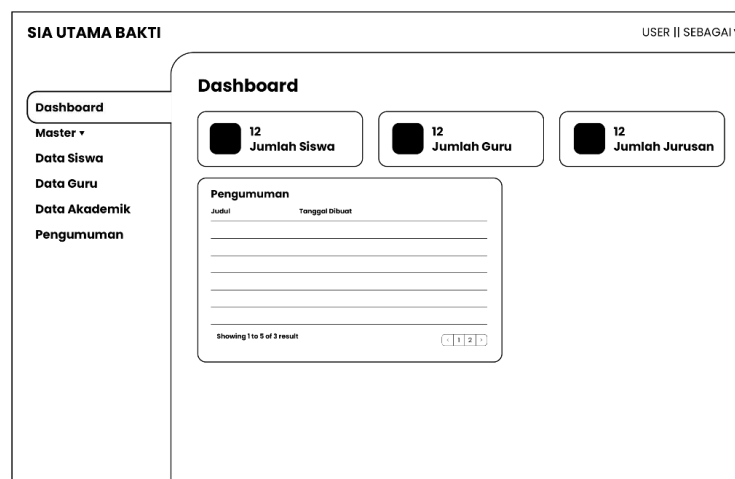
### 4.2.1 Design Interface

Tahapan ini merupakan proses pembuatan desain sistem yang berfungsi untuk memberikan gambaran sistem informasi akademik yang akan dibuat. Berikut tampilan desain yang telah dibuat.

#### 1. Desain Halaman *Dashboard* Operator

Berikut ini merupakan rancangan desain dari halaman operator, pada halaman ini operator memiliki hak akses berupa master yang terdiri dari jurusan kelas, tahun ajaran, jam pelajaran, nama pelajaran, predikat dan data mata pelajaran, data siswa, data guru, data akademik, pengumuman dan *management* akun yang dapat dilihat pada gambar 4.

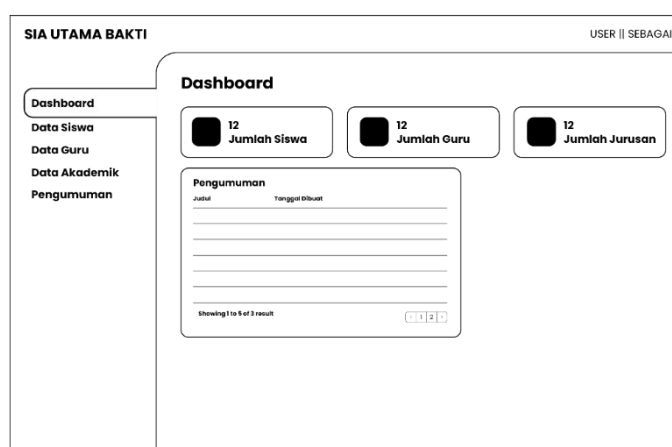
14



**Gambar 4. 14 Desain Halaman *Dashboard* Operator**

## 2. Desain Halaman *Dashboard* Kepala Sekolah

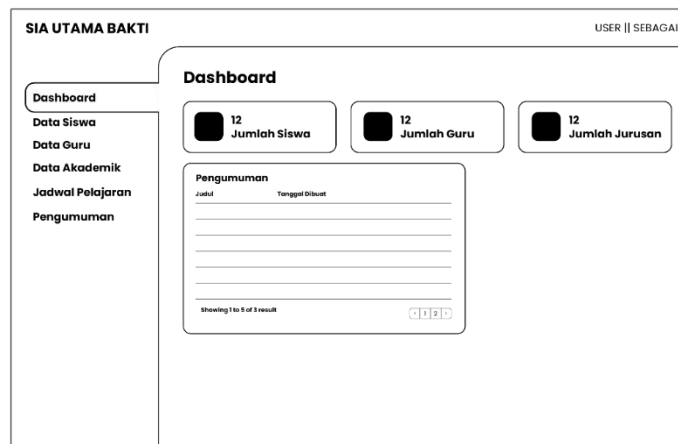
Berikut ini merupakan rancangan desain dari halaman *dashboard* kepala sekolah, pada halaman ini kepala sekolah memiliki akses melihat data siswa, melihat data guru, melihat data akademik dan melihat pengumuman yang dapat dilihat pada gambar 4. 15.



**Gambar 4. 15 Desain Halaman *Dashboard* Kepala Sekolah**

## 3. Desain Halaman *Dashboard* Waka Kurikulum

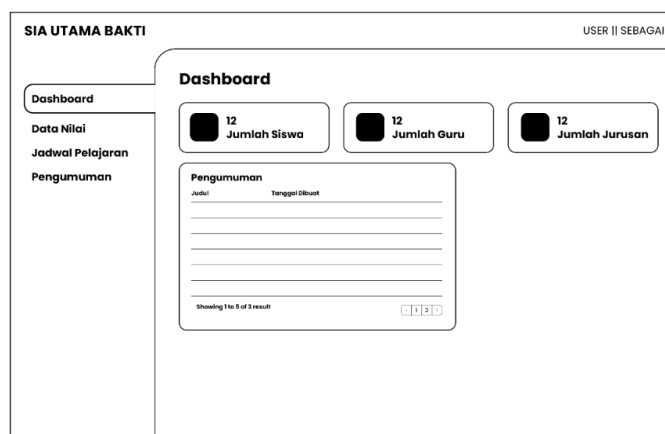
Berikut ini merupakan rancangan desain halaman *dashboard* dari waka kurikulum, pada halaman ini waka kurikulum memiliki akses melihat data siswa, melihat data guru, mengelola data akademik, mengelola jadwal pelajaran dan mengelola pengumuman yang dapat dilihat pada tabel 4. 16.



**Gambar 4. 16 Desain Halaman *Dashboard* Waka Kurikulum**

#### 4. Desain Halaman *Dashboard* Guru

Berikut ini merupakan rancangan desain halaman *dashboard* guru, pada halaman ini guru memiliki akses menambahkan nilai siswa, melihat jadwal pelajaran dan melihat pengumuman yang dapat dilihat pada gambar 4.17.

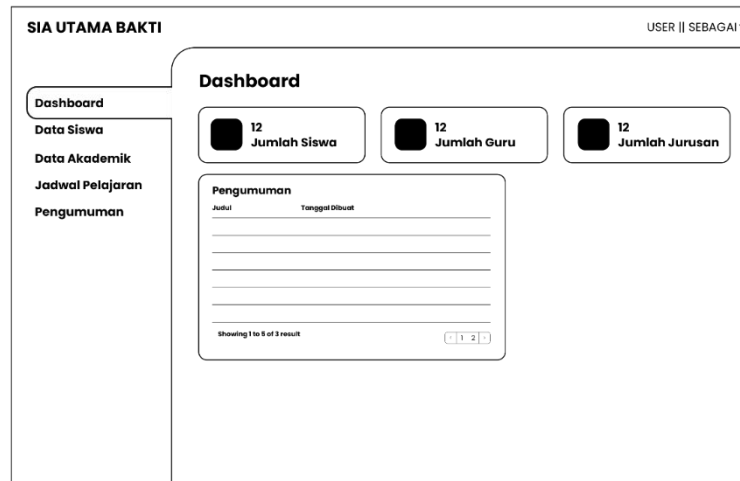


**Gambar 4. 17 Desain Halaman *Dashboard* Guru**

#### 5. Desain Halaman *Dashboard* Wali Kelas

Berikut ini merupakan rancangan desain halaman wali kelas, pada halaman ini wali kelas memiliki akses melihat data siswa mengelola

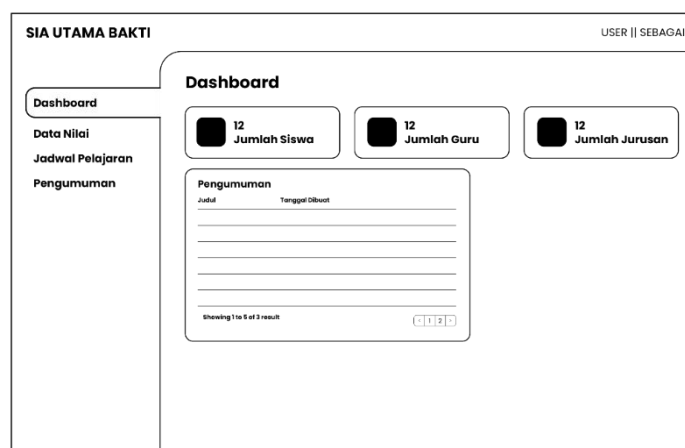
nilai siswa, melihat jadwal pelajaran dan melihat pengumuman yang dapat dilihat pada gambar 4.18.



**Gambar 4. 18 Desain Halaman *Dashboard* Wali Kelas**

#### 6. Desain Halaman *Dashboard* Siswa

Berikut ini merupakan rancangan desain halaman *dashboard* siswa, pada halaman ini siswa memiliki akses melihat nilai siswa itu sendiri, melihat jadwal pelajaran dan melihat pengumuman yang dapat dilihat pada gambar 4.19



**Gambar 4. 19 Desain Halaman *Dashboard* Siswa**

## 7. Desain Halaman Master-Jurusan

Desain halaman master-jurusan akan menampilkan data nama jurusan. Desain halaman master-jurusan ini dapat dilihat pada gambar 4. 20.

NO	Jurusan	Aksi
x	xxxxx	<input type="checkbox"/> <input type="checkbox"/>

**Gambar 4. 20 Desain Halaman Master-Jurusan**

## 8. Desain Halaman Master-Kelas

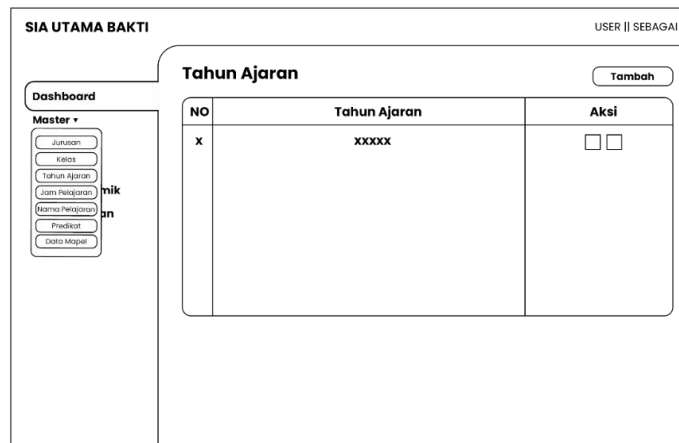
Desain halaman master-kelas akan menampilkan data kelas. Desain halaman master-kelas ini dapat dilihat pada gambar 4. 21.

NO	Tingkatan	Jurusan	Nama Kelas	Jumlah Siswa	Aksi
x	xxxxx	xxxxx	xxxxx	xxxxx	<input type="checkbox"/> <input type="checkbox"/>

**Gambar 4. 21 Desain Halaman Master-Kelas**

## 9. Desain Halaman Master-Tahun Ajaran

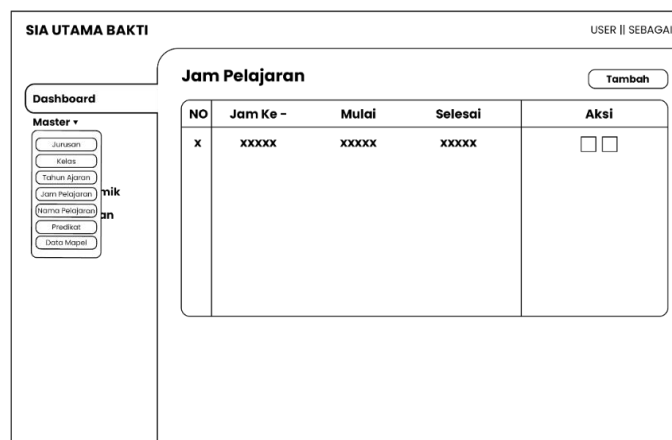
Desain halaman master-tahun ajaran akan menampilkan data tahun ajaran. Desain halaman master-tahun ajaran dapat dilihat pada gambar 4. 22.



**Gambar 4. 22 Desain Halaman Master-Tahun Ajaran**

## 10. Desain Halaman Master-Jam Pelajaran

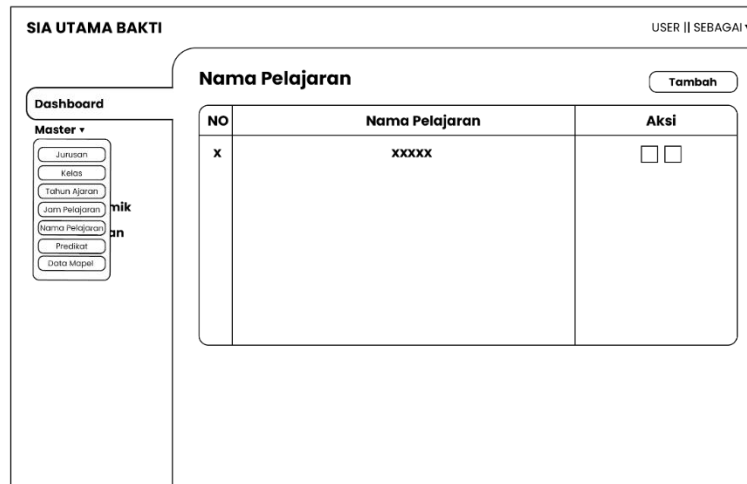
Desain halaman master-tahun ajaran akan menampilkan data tahun ajaran. Desain halaman master-tahun ajaran dapat dilihat pada gambar 4. 23.



**Gambar 4. 23 Desain Halaman Master-Jam Pelajaran**

## 11. Desain Halaman Master-Jam Pelajaran

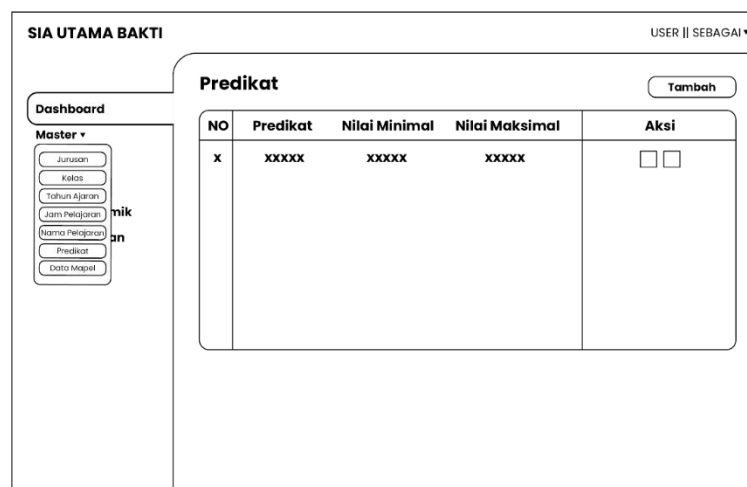
Desain halaman master-jam pelajaran akan menampilkan jam pelajaran. Desain halaman master-jam pelajaran dapat dilihat pada gambar 4. 24.



**Gambar 4. 24 Desain Halaman Master-Nama Pelajaran**

## 12. Desain Halaman Master-Predikat

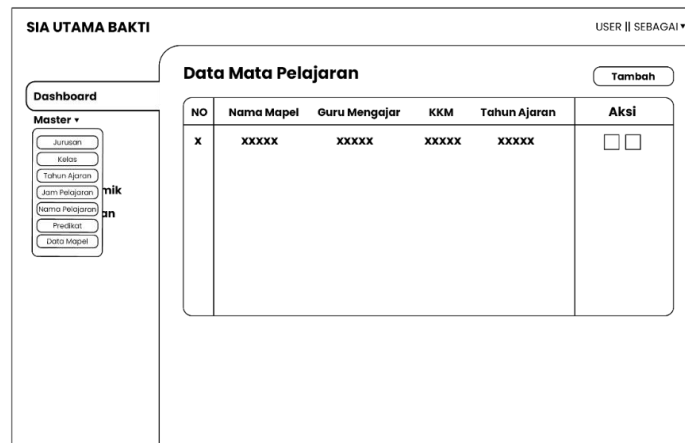
Desain halaman master-predikat akan menampilkan data predikat. Desain halaman master-predikat dapat dilihat pada gambar 4. 25.



**Gambar 4. 25 Desain Halaman Master-Predikat**

### 13. Desain Halaman Master-Data Mata Pelajaran

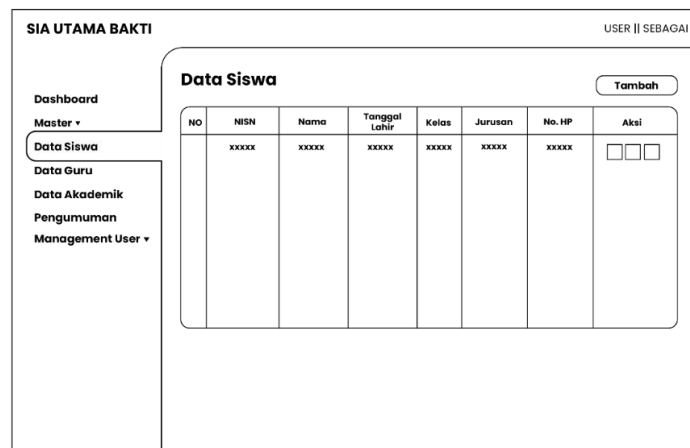
Desain halaman master-mata pelajaran akan menampilkan data mata pelajaran. Desain halaman master-mata pelajaran dapat dilihat pada gambar 4. 26.



**Gambar 4. 26 Desain Halaman Master-Data Mata Pelajaran**

### 14. Desain Halaman Data Siswa

Desain halaman data siswa akan menampilkan data siswa. Desain halaman data siswa dapat dilihat pada gambar 4. 27.



**Gambar 4. 27 Desain Halaman Data Siswa**



### 15. Desain Halaman Data Guru

Desain halaman data guru akan menampilkan data guru. Desain halaman data guru dapat dilihat pada gambar 4. 28.

SIA UTAMA BAKTI USER || SEBAGAI ▾

Dashboard  
Master ▾  
Data Siswa  
**Data Guru**  
Data Akademik  
Pengumuman  
Management User ▾

**Data Guru** Tambah

NO	NIP	Nama	Pendidikan	Status Kepegawaian	No. HP	Aksi
x	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

**Gambar 4. 28 Desain Halaman Data Guru**

### 16. Desain Halaman Data Akademik

Desain halaman data akademik akan menampilkan data akademik. Desain halaman data akademik dapat dilihat pada gambar 4. 29.

SIA UTAMA BAKTI USER || SEBAGAI ▾

Dashboard  
Master ▾  
Data Siswa  
Data Guru  
**Data Akademik**  
Pengumuman  
Management User ▾

**Data Kelas xxxx** Tambah

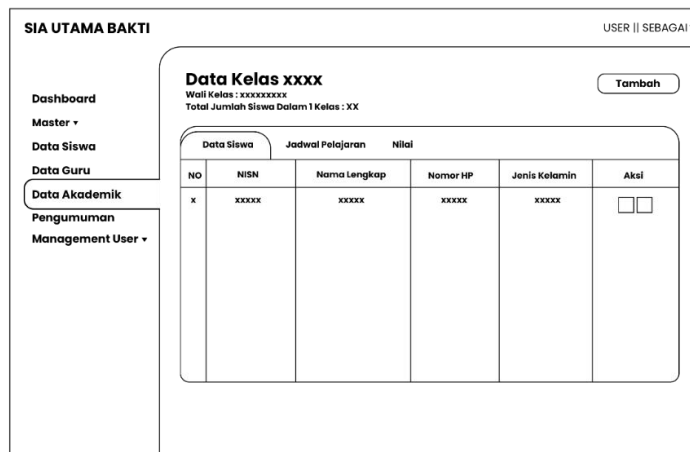
Semua Tahun ▾

NO	Nama Kelas	Wali Kelas	Tahun Ajaran	Aksi
x	xxxxx	xxxxx	xxxxx	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

**Gambar 4. 29 Desain Halaman Data Akademik**

## 17. Desain Halaman Data Kelas

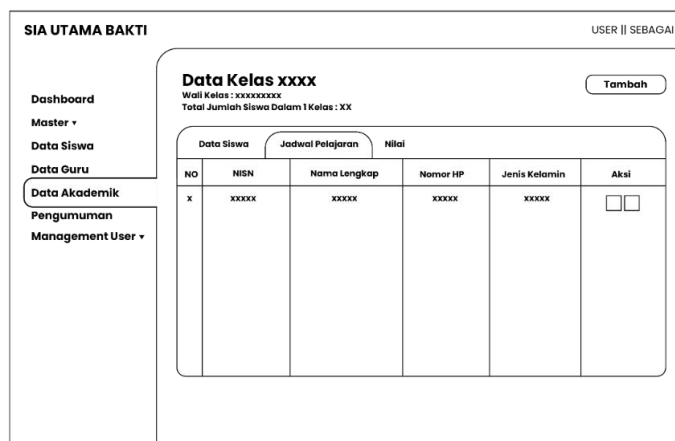
Desain halaman data kelas akan menampilkan data kelas. Desain halaman data kelas dapat dilihat pada gambar 4. 30.



**Gambar 4. 30 Desain Halaman Data Siswa**

## 18. Desain Halaman Jadwal Pelajaran

Desain halaman jadwal pelajaran akan menampilkan jadwal pelajaran berdasarjan kelas Desain halaman jadwal pelajaran dapat dilihat pada gambar 4. 31.



**Gambar 4. 31 Desain Halaman Jadwal Pelajaran**

### 19. Desain Halaman Data Nilai

Desain halaman data nilai akan menampilkan data nilai siswa .

Desain halaman data nilai dapat dilihat pada gambar 4. 32.

SIA UTAMA BAKTI								USER    SEBAGAI ▾	
<b>Nama Siswa : xxxxx</b> NISN : xxxxx Kelas : xxxxx								<input type="button" value="Tambah"/>	
Data Siswa		Jadwal Pelajaran		Nilai					
NO	Nama Siswa	Nilai Tugas	Nilai Ulangan Harian	Nilai UTS	Nilai UAS	Nilai Akhir	Aksi		
x	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	<input type="checkbox"/> <input type="checkbox"/>		

**Gambar 4. 32 Desain Halaman Data Nilai**

### 20. Desain Halaman Pengumuman

Desain halaman pengumuman akan menampilkan semua pengumuman. Desain halaman pengumuman dapat dilihat pada gambar 4. 33.

SIA UTAMA BAKTI					USER    SEBAGAI ▾	
<b>Pengumuman</b>					<input type="button" value="Tambah"/>	
NO	Judul Pengumuman	Isi Pengumuman	Tanggal Dibuat	Aksi		
x	xxxxx	xxxxx	xxxxx	<input type="checkbox"/> <input type="checkbox"/>		

**Gambar 4. 33 Desain Halaman Pengumuman**

## 21. Desain Halaman *Management User*

Desain halaman *management user* akan menampilkan data *user* .

Desain halaman *management user* dapat dilihat pada gambar 4. 34.

NO	Nama	Email	Roles	Aksi
1	xxxx	xxxx	xxxx	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

**Gambar 4. 34 Desain Halaman *Management User***

### 4.2.2 Desain *Database*

Sistem basis data yang dirancang bertujuan untuk menyimpan dan mengelola semua informasi yang terkait dengan sistem akademik. Berikut adalah perencanaan desain basis data untuk sistem tersebut.

#### 1. Tabel *User*

Tabel *user* merupakan tabel yang menampung data *user* yang dapat mengakses sistem. Adapun struktur tabel *user* dapat dilihat pada tabel 4.1

**Tabel 4. 1 *User***

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Name	Varchar	-
3	<i>Username</i>	Varchar	-
4	Password	Varchar	-

## 2. Tabel *Student*

Tabel *student* merupakan tabel yang menampung data informasi siswa, Adapun struktur tabel *student* dapat dilihat pada tabel 4.2

**Tabel 4. 2 *Student***

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	id	Bigint	Primary key
2	<i>user_id</i>	Bigint	Foreign Key
3	<i>Classroom_id</i>	Bigint	
4	NISN	Varchar	-
5	Name	Varchar	-
6	Gender	Enum	-
7	Place_of_birth	Varchar	-
8	Date_of_birth	Varchar	-
9	Phone_number	Varchar	-
10	Email	Varchar	-
11	Address	Varchar	-
12	Origin_of_school	Varchar	-
13	Father_name	Varchar	-
14	Father_work	Varchar	-
15	Father_phone_number	Varchar	-
16	Mother_name	Varchar	-
17	Mother_work	Varchar	-
18	Mother_phone_number	Varchar	-
19	Perent_address	Varchar	-
20	Wali_name	Varchar	-
21	Wali_work	Varchar	-
22	Wali_phone_number	Varchar	-
23	Wali_Addres	Varchar	-

### 3. Tabel *Teacher*

Tabel *teacher* merupakan tabel yang menampung data informasi guru

Adapun struktur *teacher* guru dapat dilihat pada tabel 4.3

**Tabel 4. 3 *Teacher***

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	<i>User_id</i>	Bigint	Foreign Key
3	NIP	Varchar	-
4	Name	Varchar	-
5	Education_level	Varchar	-
6	Lesson_ <i>teacher</i>	Varchar	-
7	Gender	Enum	-
8	Place_of_birth	Varchar	-
9	Phone_number	Varchar	-
10	Email	Varchar	-
11	Address	Varchar	-

### 4. Tabel *Classes*

Tabel *classes* merupakan tabel yang menampung data informasi kelas

Adapun struktur tabel *class* dapat dilihat pada tabel 4.4

**Tabel 4. 4 *Classes***

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Grade	Enum	-
3	<i>Expertise_id</i>	Bigint	Foreign Key
4	Name	Varchar	-
5	Limit	Bigint	-

5. Tabel *Expertise*

Tabel *expertise* merupakan tabel yang menampung nama jurusan.

Adapun struktur tabel pelajaran dapat dilihat pada tabel 4.5

**Tabel 4. 5 Course**

No	Nama Field	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Name	Varchar	-

6. Tabel *Classrooms*

Tabel *classroom* merupakan tabel yang menampung data informasi kelas,

Adapun struktur tabel *classroom* dapat dilihat pada tabel 4.6

**Tabel 4. 6 Classrooms**

No	Nama Field	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	<i>Class_id</i>	Bigint	Foreign Key
3	<i>Teacher_id</i>	Bigint	Foreign Key
4	<i>School_year_id</i>	Bigint	Foreign Key

7. Tabel *School Year*

Tabel *school year* merupakan tabel yang menampung data tahun ajaran,

Adapun struktur tabel *school year* dapat dilihat pada tabel 4.7

**Tabel 4. 7 School Year**

No	Nama Field	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Start_year	Bigint	-

8. Tabel *Course Has Teacher*

Tabel *course has teacher* merupakan tabel yang menghubungkan antara mata pelajaran dan guru, adapun struktur tabel *course has teacher* dapat dilihat pada tabel 4.8

**Tabel 4. 8 Course Has Teacher**

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Course_id	Bigint	Foreign Key
3	Teacher_id	Bigint	Foreign Key

9. Tabel *Schedule*

Tabel *schedule* merupakan tabel yang menampung data informasi mata pelajaran, Adapun struktur tabel *schedule* dapat dilihat pada tabel 4.9

**Tabel 4. 9 Schedule**

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Classroom_id	Bigint	Foreign Key
3	Course_id	Bigint	Foreign Key
4	Hour_id	Bigint	Foreign Key
5	Day	Enum	-

10. Tabel *Courses*

Tabel *course* merupakan tabel yang menampung data informasi pelajaran dan guru mengajar, Adapun struktur tabel *course* dapat dilihat pada Tabel 4.10



**Tabel 4. 10 Courses**

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Course_id	Bigint	Foreign Key
3	<i>Teacher_id</i>	Bigint	Foreign Key
4	Predicate_id	Bigint	Foreign Key
5	Schoolyears_id	Bigint	Foreign Key
6	<i>Classroom_id</i>	Bigint	Foreign Key

11. Tabel *Course*

Tabel *course* merupakan tabel yang menampung data informasi nama pelajaran, Adapun struktur tabel *course* dapat dilihat pada tabel 4.11

**Tabel 4. 11 Course**

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Name	Varchar	-

12. Tabel *Scores*

Tabel *scores* merupakan Tabel yang menampung data informasi nilai siswa, Adapun struktur tabel *scores* dapat dilihat pada tabel 4.12

**Tabel 4. 12 Scores**

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Schedule_id	Bigint	Foreign Key
3	<i>Teacher_id</i>	Bigint	Foreign Key
4	<i>Student_id</i>	Bigint	Foreign Key
5	Course_id	Bigint	Foreign Key

No	Nama <i>Field</i>	Tipe Data	Keterangan
6	Task_grades	Decimal	-
7	Daily_test	Decimal	-
8	UTS	Decimal	-
9	UAS	Decimal	-
10	Final_result	Decimal	-

### 13. Tabel *Recab Scores*

Tabel *recab scores* merupakan tabel yang menampung data rekapitulasi nilai siswa, Adapun struktur tabel *recab scores* dapat dilihat pada tabel 4.13

**Tabel 4. 13 *Recab Scores***

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Score_id	Bigint	Foreign Key
3	<i>Student_id</i>	Bigint	Foreign Key
4	Course_id	Bigint	Foreign Key
5	Avg_score	Decimal	-
6	Heighest_score	Decimal	-
7	Lowest_Score	Decimal	-

### 14. Tabel *Announcement*

Tabel *announcement* merupakan tabel yang menampung informasi pengumuman, adapun struktur tabel *announcement* dapat dilihat pada Tabel 4.14

**Tabel 4. 14 *Announcement***

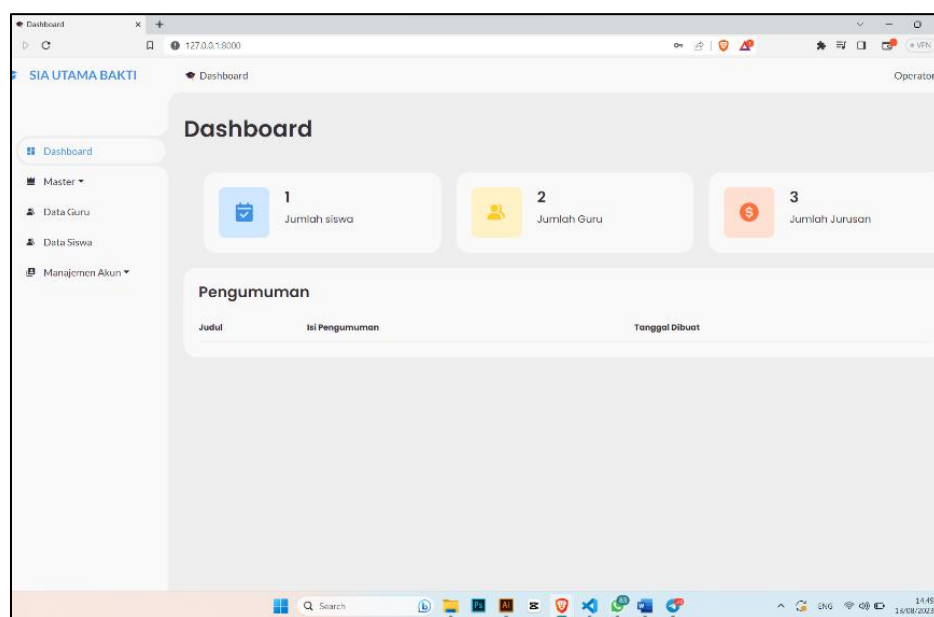
No	Nama <i>Field</i>	Tipe Data	Keterangan
1	Id	Bigint	Primary key
2	Title	Varchar	-
3	Description	Varchar	-

### 4.3 *Implementation*

Setelah tahapan perancangan desain *interface*, selanjutnya merupakan tahapan implementasi. Adapun tampilan hasil implementasi sebagai berikut :

#### 1. Desain Halaman *Dashboard* Operator

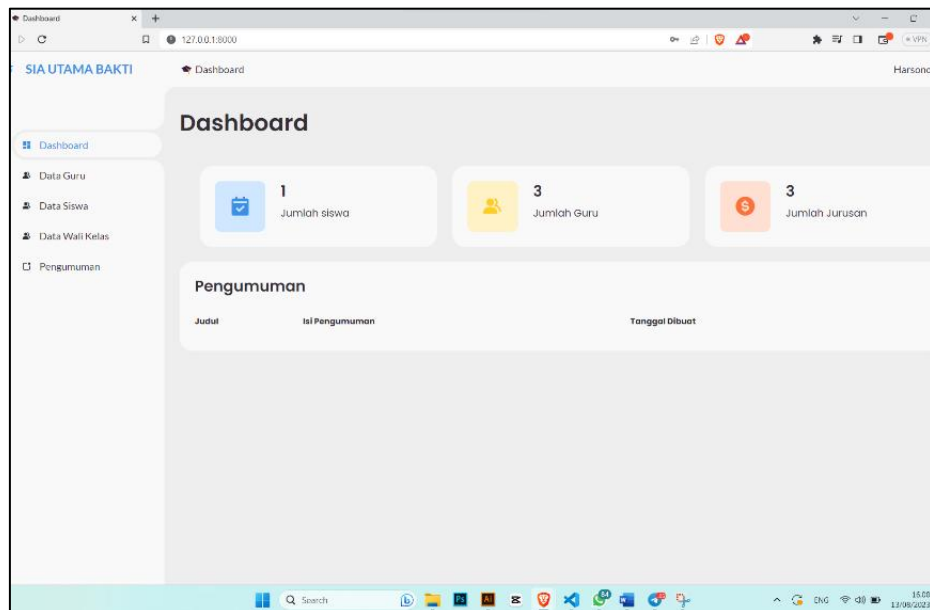
Berikut ini merupakan rancangan desain dari operator, operator merupakan *user* yang mengelola berbagai data dari SIA.



**Gambar 4. 35 Desain Halaman *Dashboard* Operator**

#### 2. Desain Halaman *Dashboard* Kepala Sekolah

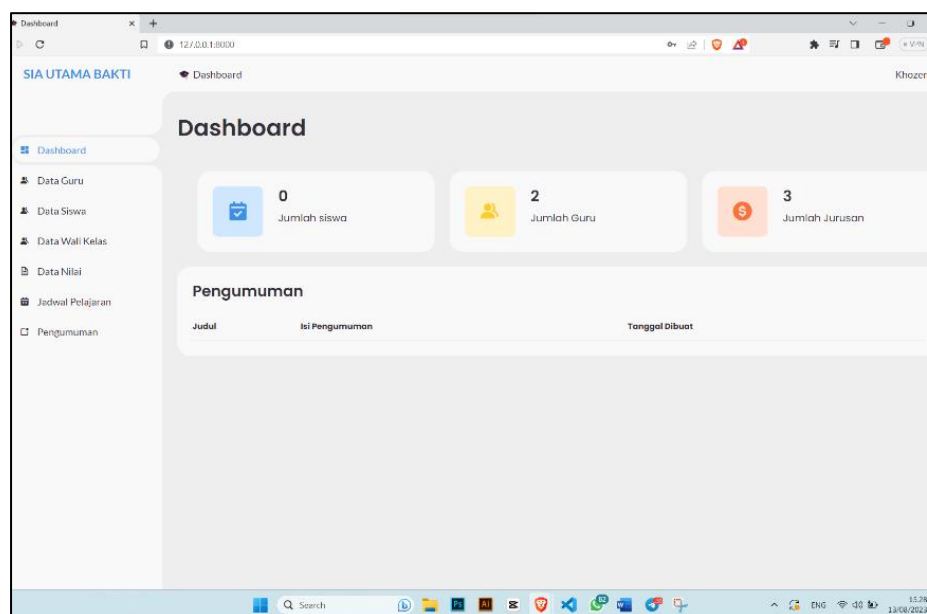
Pada gambar 4.36 merupakan hasil desain *interface dashboard* kepala sekolah. Jika kepala sekolah berhasil masuk pada menu *login*, maka halaman yang pertama kali akan muncul adalah *dashboard* .



**Gambar 4. 36 Desain Halaman *Dashboard* Kepala Sekolah**

### 3. Desain Halaman *Dashboard* Waka Kurikulum

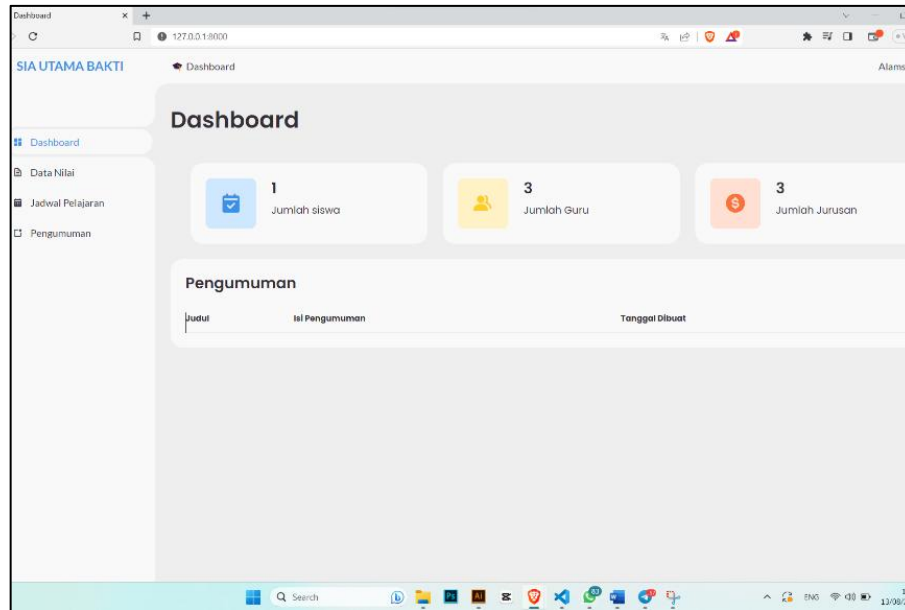
Pada gambar 4.37 merupakan hasil desain *interface dashboard* waka kurikulum. Jika waka kurikulum berhasil masuk pada menu *login*, maka halaman yang pertama kali akan muncul adalah *dashboard* .



**Gambar 4. 37 Desain Halaman *Dashboard* Waka Kurikulum**

#### 4. Desain Halaman *Dashboard* Guru

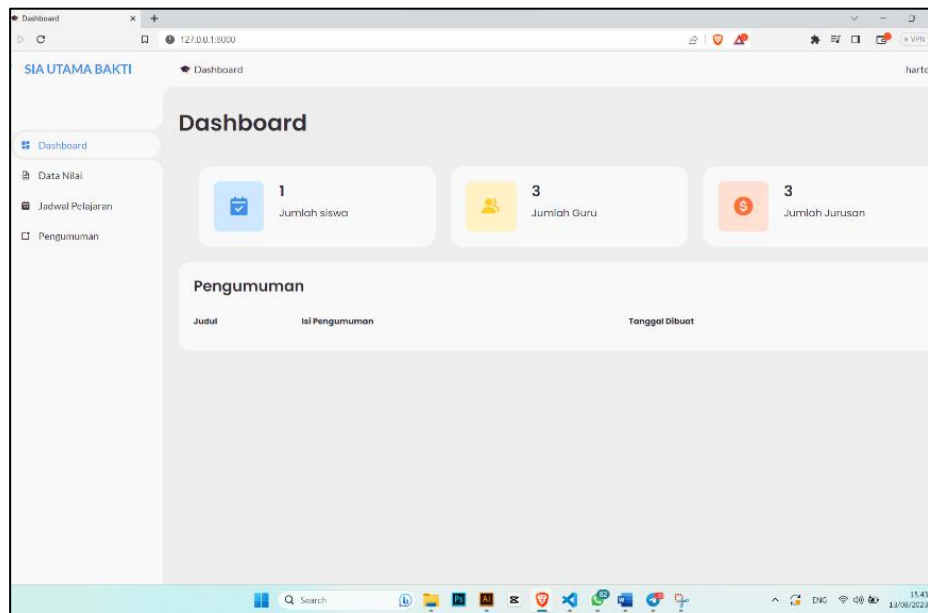
Pada gambar 4.38 merupakan hasil desain *interface dashboard* guru. Jika guru berhasil masuk pada menu *login*, maka halaman yang pertama kali akan muncul adalah *dashboard*.



**Gambar 4. 38 Desain Halaman *Dashboard* Guru**

#### 5. Desain Halaman *Dashboard* Wali Kelas

Pada gambar 4.39 merupakan hasil desain *interface dashboard* wali kelas. Jika wali kelas berhasil masuk pada menu *login*, maka halaman yang pertama kali akan muncul adalah *dashboard*.

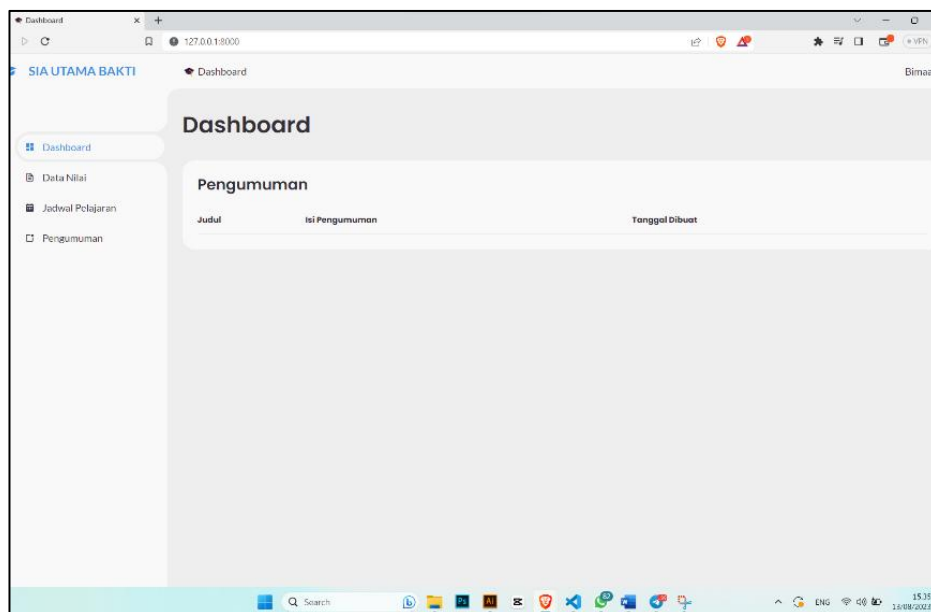


**Gambar 4. 39 Desain Halaman *Dashboard* Wali Kelas**

#### 6. Desain Halaman *Dashboard* Siswa

Pada gambar 4.40 merupakan hasil desain *interface dashboard* siswa.

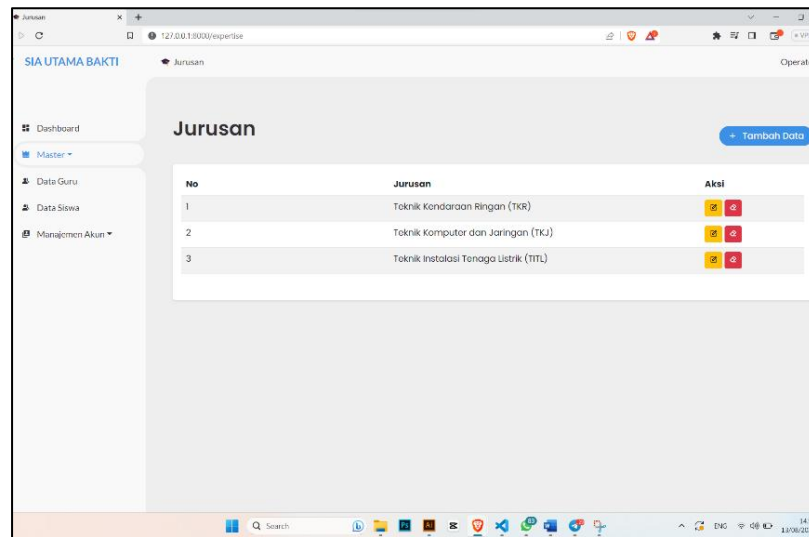
Jika siswa berhasil masuk pada menu *login*, maka halaman yang pertama kali akan muncul adalah *dashboard* .



**Gambar 4. 40 Desain Halaman *Dashboard* Siswa**

## 7. Desain Halaman Master-Jurusan

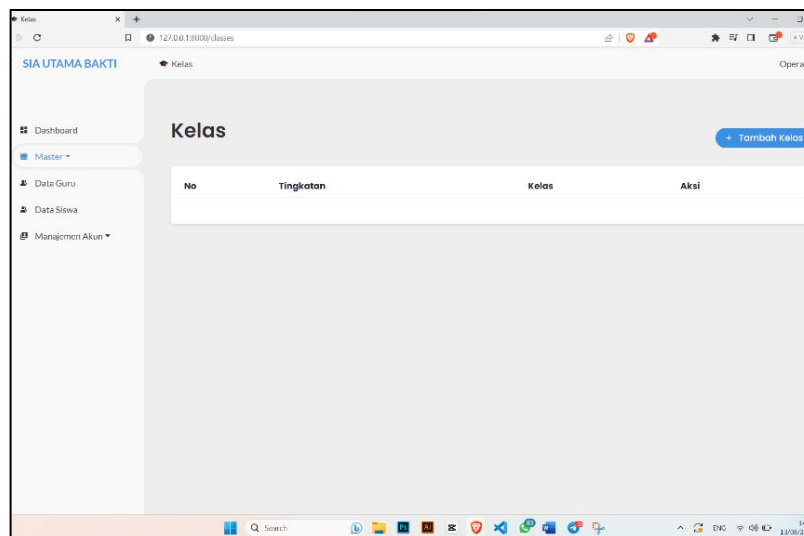
Pada gambar 4.41 merupakan hasil desain nama jurusan. Pada halaman ini yang bisa mengakses adalah operator.



**Gambar 4. 41 Desain Halaman Master-Jurusan**

## 8. Desain Halaman Master-Kelas

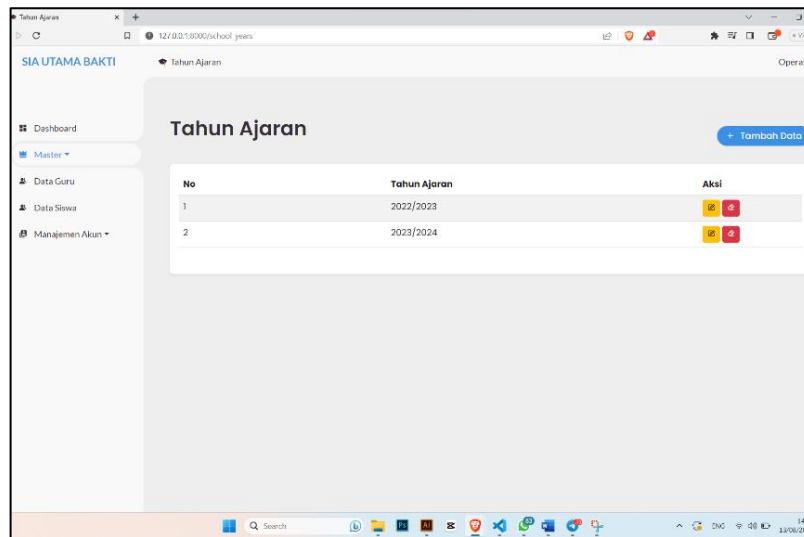
Pada gambar 4.42 merupakan hasil desain kelas. Pada halaman ini yang bisa mengakses adalah operator.



**Gambar 4. 42 Desain Halaman Master-Kelas**

## 9. Desain Halaman Master-Tahun Ajaran

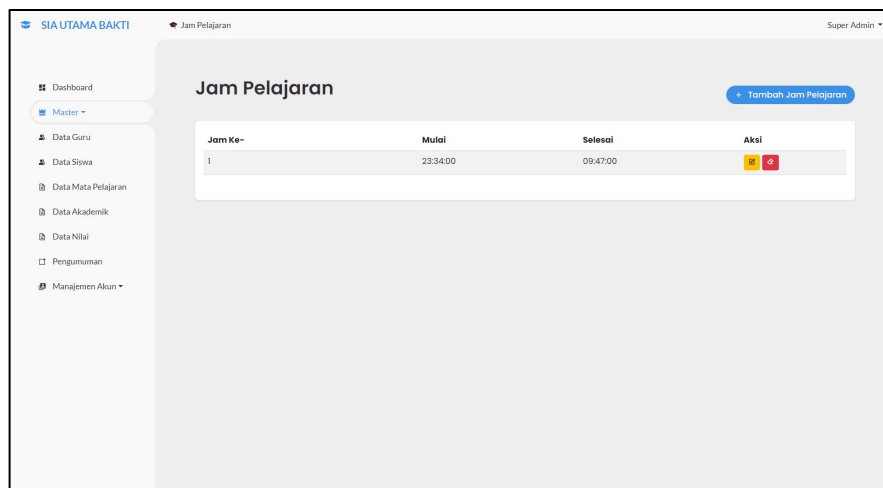
Pada gambar 4.43 merupakan hasil desain tahun ajaran. Pada halaman ini yang bisa mengakses adalah operator.



**Gambar 4. 43 Desain Halaman Master-Tahun Ajaran**

## 10. Desain Halaman Master-Jam Pelajaran

Pada gambar 4.44 merupakan hasil desain jam pelajaran. Pada halaman ini yang bisa mengakses adalah operator.

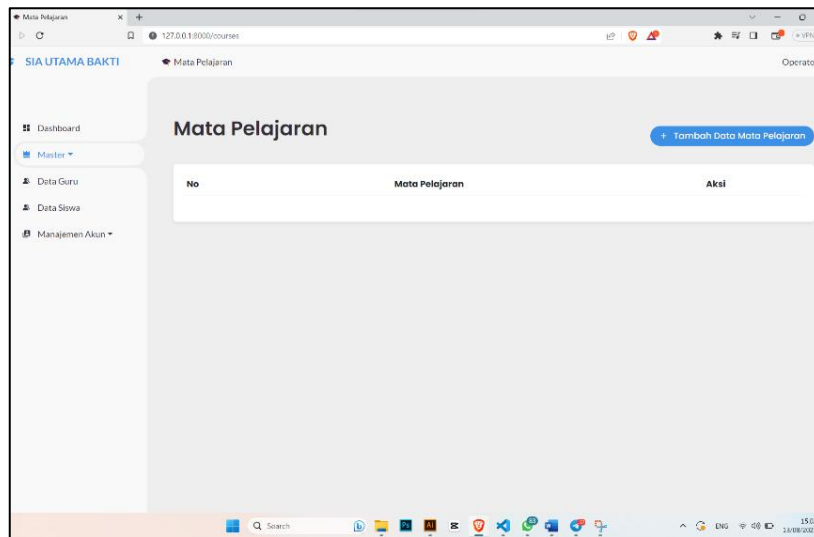


**Gambar 4. 44 Desain Halaman Master-Jam Pelajaran**



## 11. Desain Halaman Master-Nama Pelajaran

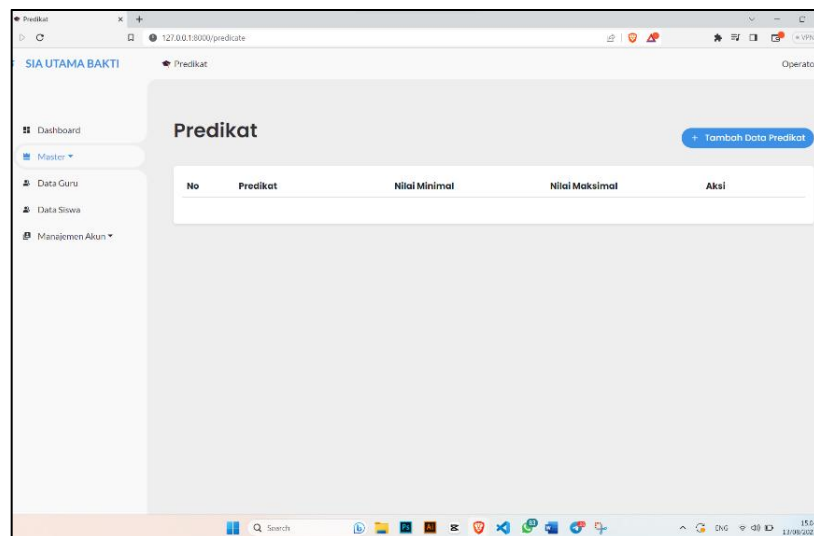
Pada gambar 4.45 merupakan hasil desain nama pelajaran. Pada halaman ini yang bisa mengakses adalah operator.



**Gambar 4. 45 Desain Halaman Master-Nama Pelajaran**

## 12. Desain Halaman Master-Predikat

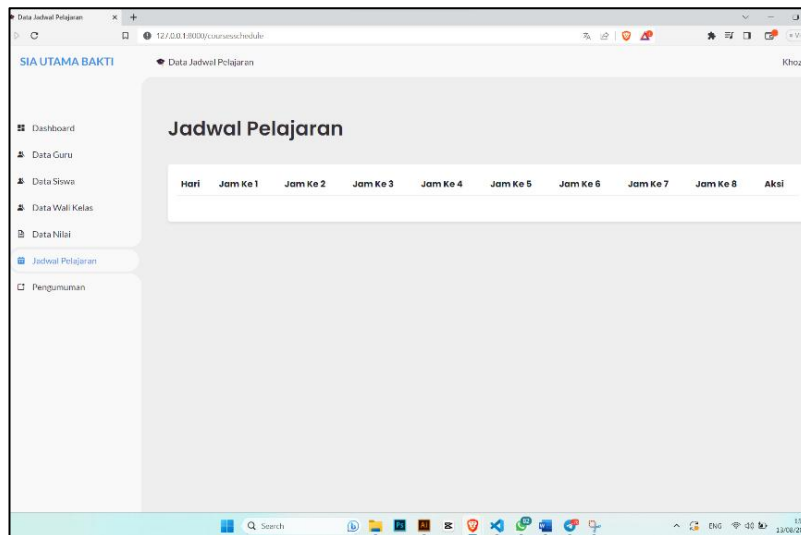
Pada gambar 4.46 merupakan hasil desain predikat. Pada halaman ini yang bisa mengakses adalah operator.



**Gambar 4. 46 Desain Halaman Master-Predikat**

### 13. Desain Halaman Master-Data Mapel

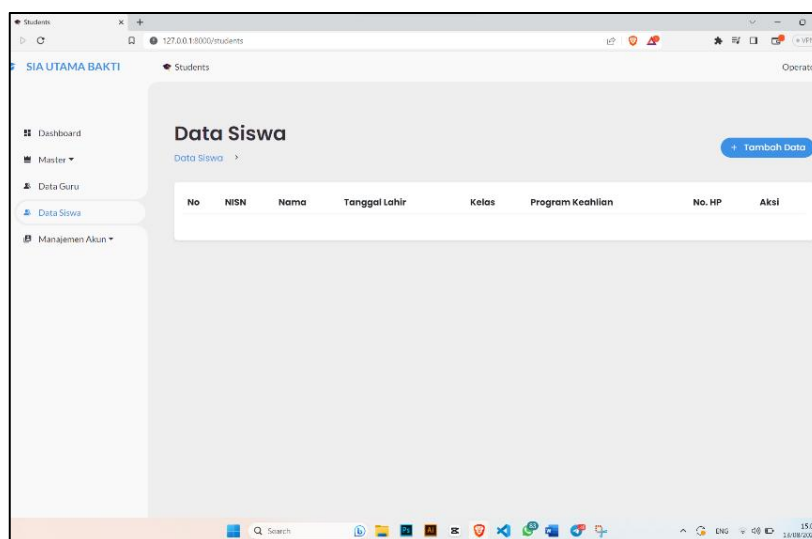
Pada gambar 4.47 merupakan hasil desain data mapel. Pada halaman ini yang bisa mengakses adalah operator.



**Gambar 4. 47 Desain Halaman Master-Data Mapel**

### 14. Desain Halaman Data Siswa

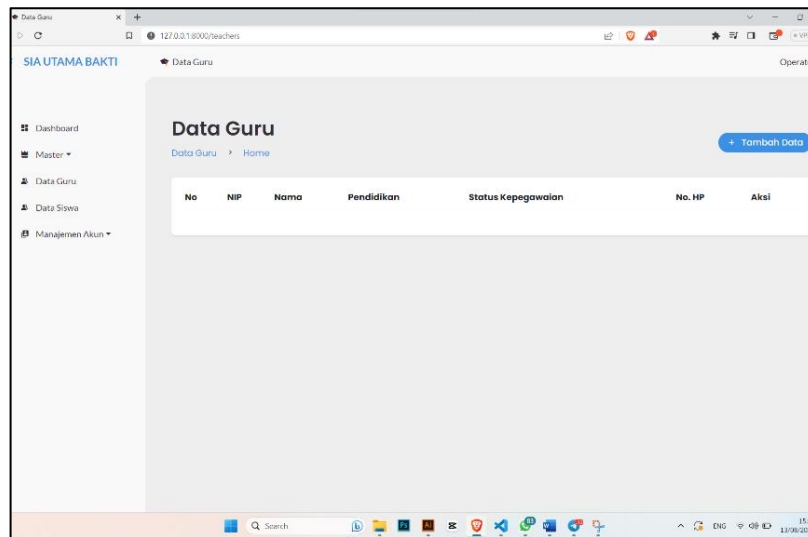
Pada gambar 4.48 merupakan hasil desain data siswa. Pada halaman ini yang bisa mengakses adalah operator dan waka kurikulum.



**Gambar 4. 48 Desain Halaman Data Siswa**

## 15. Desain Halaman Data Guru

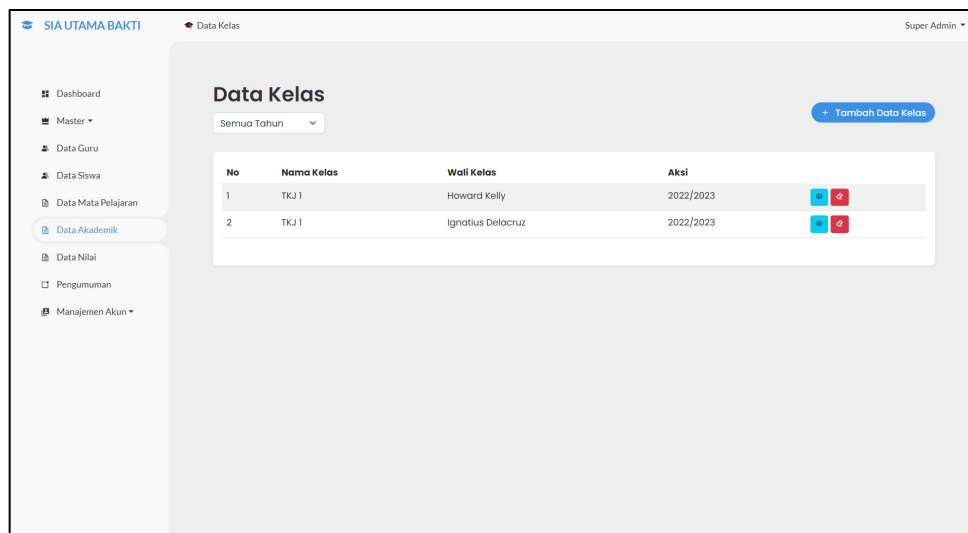
Pada gambar 4.49 merupakan hasil desain data guru. Pada halaman ini yang bisa mengakses adalah operator dan waka kurikulum.



**Gambar 4. 49 Desain Halaman Data Guru**

## 16. Desain Halaman Data Akademik

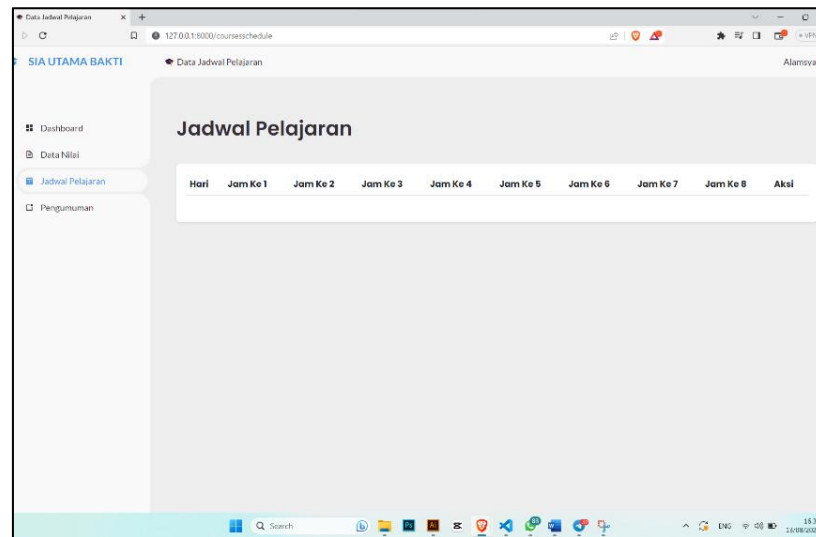
Pada gambar 4.50 merupakan hasil desain data akademik. Pada halaman ini yang bisa mengakses adalah operator dan waka kurikulum.



**Gambar 4. 50 Desain Halaman Data Akademik**

## 17. Desain Halaman Jadwal Pelajaran

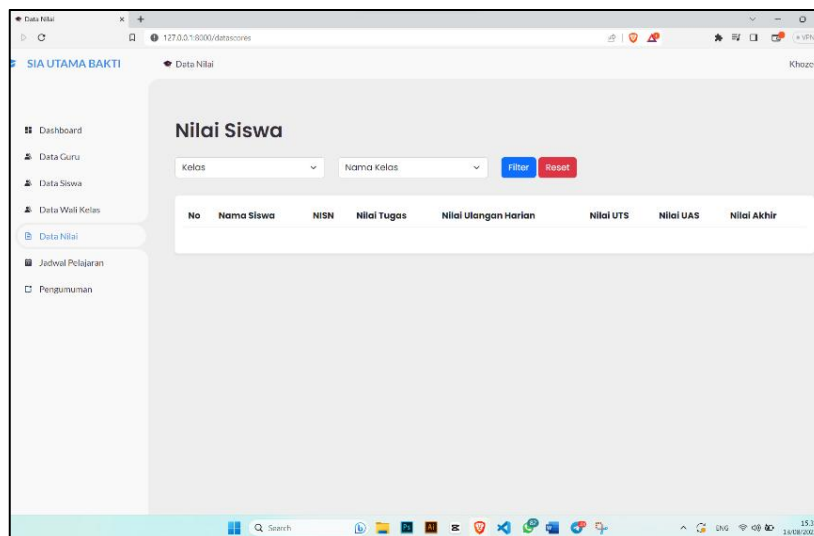
Pada gambar 4.51 merupakan hasil desain jadwal pelajaran. Pada halaman ini yang bisa mengakses adalah operator, waka kurikulum dan guru.



**Gambar 4. 51 Desain Halaman Jadwal Pelajaran**

## 18. Desain Halaman Data Nilai

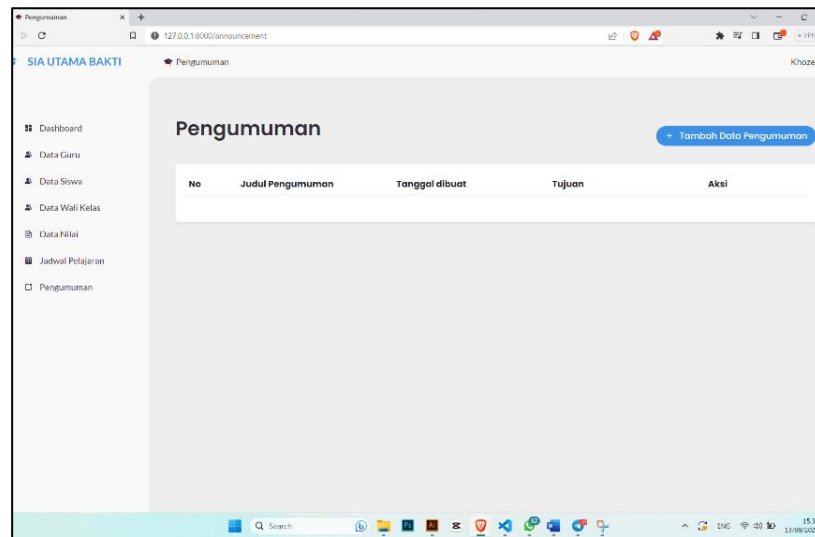
Pada gambar 4.52 merupakan hasil desain data nilai. Pada halaman ini yang bisa mengakses adalah operator, waka kurikulum dan guru.



**Gambar 4. 52 Desain Halaman Data Nilai**

## 19. Desain Halaman Pengumuman

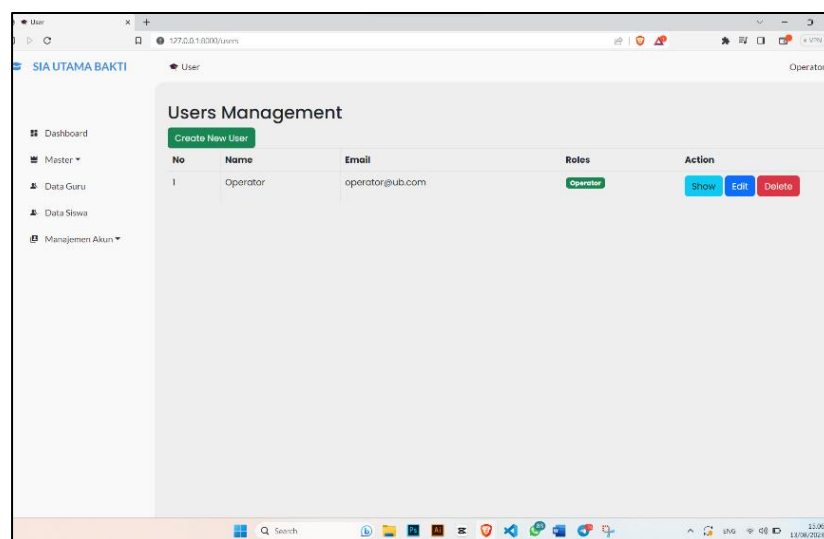
Pada gambar 4.53 merupakan hasil desain pengumuman. Pada halaman ini yang bisa mengakses adalah semua pengguna.



**Gambar 4. 53 Desain Halaman Pengumuman**

## 20. Desain Halaman *Management User*

Pada gambar 4.54 merupakan hasil desain *management user*. Pada halaman ini yang bisa mengakses adalah operator.



**Gambar 4. 54 Desain Halaman *Management User***

#### 4.4 Pengujian Perangkat Lunak

Setelah tahapan dan perancangan sistem akademik, selanjutnya dilakukan pengujian sebagai tahapan akhir sebelum digunakan oleh pihak terkait apakah sistem telah berjalan dengan baik dan benar.

Jenis pengujian perangkat lunak yang dilakukan dalam penulisan ini adalah *black box testing*. Dalam metode pengujian ini, fokus utama adalah menguji fungsi-fungsi dan interaksi antara *input* dan *output* dari perangkat lunak, tanpa memperhatikan detail implementasi internalnya. Dengan cara ini, dapat dipastikan bahwa perangkat lunak berjalan sesuai dengan spesifikasi yang telah ditentukan sebelumnya, dan dapat berperilaku dengan baik ketika digunakan oleh pengguna akhir. Rancangan tiap proses dapat dilihat pada tabel 4.15.

**Tabel 4. 15 Tabel Pengujian Sistem**

No	Role User	Kelas Pengujian	Butir Uji	Hasil
1	Admin	<i>Login</i>	Memasukan <i>Username</i> dan <i>Password User</i>	Berhasil
		Mengelola Data Siswa	Halaman untuk melihat, menambah, menghapus, dan mengubah data siswa	Berhasil
		Mengelola Data Kelas	Halaman untuk melihat, menambah, menghapus, dan mengubah data kelas	Berhasil
		Mengelola Data Guru	Halaman untuk melihat, menambah, menghapus, dan mengubah data guru	Berhasil

No	Role User	Kelas Pengujian	Butir Uji	Hasil
		Mengelola Manajemen Akun	Halaman untuk melihat, menambah, menghapus, dan mengubah data <i>user</i> akun	Berhasil
2.	Kepala Sekolah	Melihat Data Guru	Halaman untuk melihat laporan data guru	Berhasil
		Melihat Data Siswa	Halaman untuk melihat laporan data siswa	Berhasil
		Melihat Data Wali Kelas	Halaman untuk melihat laporan data wali kelas	Berhasil
		Melihat Data Pengumuman	Halaman untuk melihat laporan data pengumuman	Berhasil
3.	Wakil Kurikulum	Melihat Data Guru	Halaman untuk melihat laporan data guru	Berhasil
		Melihat Data Siswa	Halaman untuk melihat laporan data siswa	Berhasil
		Melihat Data Wali Kelas	Halaman untuk melihat laporan data wali kelas	Berhasil
		Melihat Data Nilai	Halaman untuk melihat laporan data nilai	Berhasil
		Melihat Jadwal Pelajaran	Halaman untuk melihat laporan data jadwal pelajaran	Berhasil
		Melihat Pengumuman	Halaman untuk melihat laporan data pengumuman	Berhasil
4.	Wali Kelas	Melihat Nilai Siswa	Halaman untuk melihat	Berhasil

No	Role User	Kelas Pengujian	Butir Uji	Hasil
			laporan data nilai siswa	
		Melihat Jadwal Pelajaran	Halaman untuk melihat laporan data jadwal Pelajaran	Berhasil
		Melihat Pengumuman	Halaman untuk melihat laporan data pengumuman	Berhasil
5.	Guru	Menambahkan Nilai siswa	Halaman untuk menambahkan nilai nilai siswa	Berhasil
		Melihat Jadwal Pelajaran	Halaman untuk melihat laporan data jadwal pelajaran	Berhasil
		Melihat Pengumuman	Halaman untuk melihat laporan data pengumuman	Berhasil
6.	Siswa	Melihat Nilai	Halaman untuk melihat nilai nilai tiap mata Pelajaran	Berhasil
		Melihat Jadwal Pelajaran	Halaman untuk melihat data jadwal Pelajaran	Berhasil
		Melihat Pengumuman	Halaman untuk melihat laporan pengumuman	Berhasil

#### 4.5 Kesimpulan Hasil Pengujian

Hasil evaluasi terhadap sistem akademik SMK Utama Bakti Palembang membuktikan bahwa perjalanan sistem ini telah melewati rangkaian tahapan yang fundamental, dimulai dari analisis awal hingga pengujian akhir. Proses yang



dimulai dengan tahap analisis mendalam, perancangan yang teliti, dan implementasi yang cermat, berbuah pada hasil uji coba yang memuaskan. Hasil dari rangkaian uji coba tersebut menunjukkan bahwa sistem telah berhasil mengatasi berbagai tantangan dengan baik dan mampu menjawab kebutuhan sistem akademik dengan optimal.

Dalam konteks ini, hasil dari pengujian menunjukkan kesesuaian dan kesiapan sistem untuk beroperasi secara efektif. Hasil positif dari pengujian ini memberikan keyakinan bahwa sistem telah berhasil mencapai kemampuannya dalam menjalankan tugas-tugas yang berkaitan dengan sistem akademik. Penggunaan sistem ini ditemukan mampu menghadirkan efisiensi, akurasi, dan keterpaduan dalam memproses berbagai aspek penting dari kegiatan akademik.

Oleh karena itu, dapat disimpulkan bahwa hasil pengujian sistem akademik SMK Utama Bakti Palembang secara jelas mengindikasikan bahwa sistem ini siap dan layak digunakan secara optimal dalam mendukung segala tugas dan fungsi yang berkaitan dengan sistem akademik. Dengan melewati semua tahapan dengan baik dan hasil pengujian yang menggembirakan, sistem ini membuktikan potensinya dalam meningkatkan efisiensi dan efektivitas kegiatan akademik di lingkungan SMK Utama Bakti Palembang.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil pembahasan yang telah dijabarkan pada bab bab sebelumnya, maka dengan ini penulis dapat menyimpulkan yaitu :

- 1) Sistem Akademik dapat memberikan informasi tentang seputar akademik di SMK Utama Bakti Palembang bagi guru, wali kelas, siswa, beserta kepala sekolah secara cepat dan lebih efisien dalam memperoleh informasi
- 2) Sistem akademik juga memberikan kemudahan dalam mengakses data akademik dimulai dari data siswa, data guru, mata pelajaran, ruangan, kelas, nilai yang memberikan manfaat yang besar bagi kegiatan akademik di SMK Utama Bakti Palembang
- 3) Data seputar akademik saling terintegrasi pada sistem akademik yang mempermudah operator dalam pengelolaan data akademik di SMK Utama Bakti Palembang

#### **5.2 Saran**

Berdasarkan kesimpulan diatas yang telah dijabarkan sebelumnya, maka penulis memiliki beberapa saran yang dapat dijadikan bahan dalam perbaikan dan pengembangan sistem selanjutnya bagi SMK Utama Bakti Palembang, Adapun sarannya sebagai berikut

- 1) Melakukan monitoring pada sistem akademik agar dapat berjalan dengan baik dari segi keandalan, keamanan, serta kenyamanan bagi sistem. Serta dapat

meningkatkan dari segi keamanan agar tidak terjadi hal yang tidak diinginkan yang dimanfaatkan oleh pihak yang tidak bertanggung jawab

- 2) Menambah beberapa fitur seperti sistem pembayaran yang langsung konfirmasi ketika siswa selesai melakukan pembayaran, menambahkan fitur absensi otomatis sesuai jadwal pelajaran secara cepat dan akurat

## DAFTAR PUSTAKA

- Al'asyari, Syaiful Huda. *Perancangan Sistem Peramalan Inflasi Manajemen Toko Menggunakan Metode Exponential Smoothing (Studi Kasus Pada Toko Roti Al Huda)*. Universitas Muhammadiyah Malang, 2023.
- Alfiati, R. *Sistem Informasi Akademik SMA Negeri 12 Palembang Berbasis Website*. Institut Teknologi dan Bisnis Palcomtech Palembang, 2020.
- Arif, Erman, et al. "Perancangan Sistem Informasi Checklist Kendaraan Menggunakan Metode Rapid Application Development (RAD)." *Journal on Education*, vol. 5, no. 4, 2023, pp. 16286–95, 2023
- Herdayati, S. Pd, et al. "Desain Penelitian Dan Teknik Pengumpulan Data Dalam Penelitian." ISSN 2502-3632 ISSN 2356-0304 J. Online int. Nas. Vol. 7 No. 1, Januari–Juni 2019 Univ. 17 Agustus 1945 Jakarta, vol. 53, no. 9, 2019.
- Jaelani, Ahmad, et al. "Penggunaan Media Online Dalam Proses Kegiatan Belajar Mengajar Pai Dimasa Pandemi Covid-19 (Studi Pustaka Dan Observasi Online)." *Jurnal IKA PGSD (Ikatan Alumni PGSD) Unars*, vol. 8, no. 1, 2020.
- Nugraha, Agus Ramdhani, et al. "Sistem Informasi Akademik Sekolah Berbasis Web Di Sekolah Menengah Pertama Negeri 11 Tasikmalaya." *Jurnal Manajemen Informatika (JUMIKA)*, vol. 4, no. 2, 2020.
- Prihandoyo, M. Teguh. "Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web." *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 3, no. 1, 2018.

- Putri, Arum Ekasari. “*Evaluasi Program Bimbingan Dan Konseling: Sebuah Studi Pustaka.*” *Jurnal Bimbingan Konseling Indonesia*, vol. 4, no. 2, 2019.
- Sari, D. Purnama, et al. “*Implementasi Framework Laravel Pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang).*” *J. Informatika. Dan Rekayasa Perangkat Lunak*, vol. 2, no. 1, 2020.
- Wahid, Aceng Abdul. “*Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi.*” *J. Ilmu-Ilmu Informatika. Dan Manajemen. STMIK*, No. November, 2020.
- Widia, Dewa Made, and Salnan Ratih Asriningtias. *Cara Cepat Dan Praktis Membangun Web Dinamis Dengan PHP Dan MySQL*. Universitas Brawijaya Press, 2021.
- Widiati, Wina. “*Sistem Informasi Akademik Pada SMK Multicomp Depok.*” *SPEED-Sentra Penelitian Engineering Dan Edukasi*, vol. 11, no. 2, 2019.
- Yulianto, Eko, et al. “*Sistem Informasi Akademik Penilaian Berbasis Web Pada Sma Mandiri Balaraja-Kab. Tangerang.*” *Techno Nusa Mandiri*, vol. 15, no. 1, 2018.
- Zahara, Lisma, et al. “*Sistem Informasi Geografis Pemetaan Sekolah Menengah Kejuruan Di Kabupaten Labuhanbatu Menggunakan Webgis.*” *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, vol. 7, no. 2, 2021.

## LAMPIRAN

## 1. Announcement

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\User;
```

```
use App\Models\Announcement;
```

```
use App\Models\Teacher;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Routing\Controller;
```

```
use Spatie\Permission\Models\Role;
```

```
use Spatie\Permission\Models\Permission;
```

```
class AnnouncementController extends Controller
```

```
{
```

```
    function __construct()
```

```
    {
```

```
        $this->middleware('permission:pengumuman|pengumuman-  
create|pengumuman-edit|pengumuman-delete', ['only' => ['index', 'show']]);
```

```
        $this->middleware('permission:pengumuman-create', ['only' => ['create',  
'store']]);
```

```
        $this->middleware('permission:pengumuman-edit', ['only' => ['edit',  
'update']]);
```

```
        $this->middleware('permission:pengumuman-delete', ['only' => ['destroy']]);
```

```
    public function index(Request $request)
```

```
    {
```

```
        $announcements = Announcement::orderBy('id', 'DESC')->paginate(4);
```

```
        $roles = Role::all();
```

```
        $title = 'announcements';
```

```
        return view('announcement/index', compact('roles', 'announcements', 'title'));
```

```
    }
```

```

/**
 * Show the form for creating a new resource.
 */

public function create()
{
    $announcements = Announcement::all();
    $roles = Role::whereNotIn('name', ['Super Admin'])->get();
    return view('announcement/create', ["title" => "Pengumuman"],
compact('roles', 'announcements'));
}

public function store(Request $request)
{
    $request->validate(
        [
            'title' => 'required',
            'description' => 'required',
            'role' => 'required'
        ],
        [
            'title.required' => 'Judul wajib di isi',
            'description.required' => 'Isi Pengumuman wajib di isi',
            'role.required' => 'Role Wajib di isi'
        ]
    );
    $announcements = [
        'title' => $request->title,
        // 'role_id' => $request->role,
        'description' => $request->description,
    ];
}

```



```

$phone_numbers = [];

// return $teacher;
// $input = $request->all();
// $input['role'] = $request->input('role');

// $role = Role::create(['name' => $request->input('name')]);
$users = User::role($request->role)->get();
$phone_numbers = array();
for ($i = 0; $i < count($users); $i++) {
    $get_teacher = User::join("teachers", "users.id", "=", "teachers.user_id")
        ->select('teachers.phone_number')
        ->where('teachers.user_id', $users[$i]['id'])
        ->get();
    $get_student = User::join("students", "users.id", "=", "students.user_id")
        ->select('students.phone_number', 'students.father_phone_number',
'students.mother_phone_number', 'students.wali_phone_number')
        ->where('students.user_id', $users[$i]['id'])
        ->get();
    // dd($get_student);
    foreach ($get_teacher as $teach) {
        $phone_numbers[] = $teach->phone_number;
    }
    foreach ($get_student as $stud) {
        $phone_numbers[] = $stud->phone_number;
        $phone_numbers[] = $stud->father_phone_number;
        $phone_numbers[] = $stud->mother_phone_number;
        $phone_numbers[] = $stud->wali_phone_number;
    }
}
}

```

```

// dd($phone_numbers);
foreach ($phone_numbers as $i => $pn) {
    $this->send_wa($pn, strip_tags($request->description));
}

Announcement::create($announcements);
return redirect()->to('announcement')->with('success', 'Berhasil
menambahkan Pengumuman');
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(string $id)
{
    $announcements = Announcement::find($id);
    $roles = Role::all();
    $title = 'Form Pengumuman';
    return view('announcement/edit', compact('roles', 'announcements', 'title'));
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, string $id)
{
    $request->validate(
        [
            'title' => 'required',
            'description' => 'required',
            // 'role' => 'required'

```

```

    ],
    [
        'title.required' => 'Judul wajib di isi',
        // 'role.required' => 'Role Wajib di isi',
        'description.required' => 'Isi Pengumuman wajib di isi',
    ]
);
$announcements = [
    'title' => $request->title,
    'description' => $request->description,
    // 'role_id' => $request->role
];
Announcement::where('id', $id)->update($announcements);
return redirect()->to('announcement')->with('success', 'Berhasil edit
Pengumuman');
}
/**
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{
    Announcement::where('id', $id)->delete();
    return redirect()->to('announcement')->with('success', 'Berhasil hapus
Pengumuman');
}

private function send_wa($phone_no, $message)
{
    if (!empty($phone_no) && !empty($message)) {

        $message = preg_replace("/(\r)/", "<ENTER>", $message);
    }
}

```

```

$phone_no = preg_replace("/(\r)/", "", $phone_no);

$curl = curl_init();

$data = array(
    "phone_no" => $phone_no,
    "key" => "33ad77ffaa67d2a25de2020c4dff34654b40b3ea0cdb3a1d",
    "message" => $message
);
$data_string = json_encode($data);

curl_setopt_array($curl, array(
    CURLOPT_URL => 'http://116.203.92.59/api/async_send_message',
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 0,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => 'POST',
    CURLOPT_POSTFIELDS => $data_string,
    CURLOPT_HTTPHEADER => array(
        'Content-Type: application/json'
    ),
));

$result = curl_exec($curl);
return $result;
}
}

```

```
}
```

## 2. Classes

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Classes;
```

```
use App\Models\Expertise;
```

```
use Illuminate\Http\Request;
```

```
class ClassesController extends Controller
```

```
{
```

```
    // function __construct()
```

```
    // {
```

```
    //     $this->middleware('permission:kelas|kelas-create|kelas-edit|kelas-delete',  
['only' => ['index', 'show']]);
```

```
    //     $this->middleware('permission:kelas-create', ['only' => ['create', 'store']]);
```

```
    //     $this->middleware('permission:kelas-edit', ['only' => ['edit', 'update']]);
```

```
    //     $this->middleware('permission:kelas-delete', ['only' => ['destroy']]);
```

```
    // }
```

```
/**
```

```
 * Display a listing of the resource.
```

```
*/
```

```
public function index()
```

```
{
```

```
    $classes = Classes::all();
```

```
    $title = 'Kelas';
```

```
    return view('master/classes/index', compact('classes', 'title'));
```

```
}
```

```

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    $classes = Classes::all();
    $expertises = Expertise::all();
    $title = 'Kelas';
    return view('master/classes/create', compact('expertises', 'classes', 'title'));
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    $request->validate(
        [
            'grade' => 'required',
            'expertise' => 'required',
            'name' => 'required',
            'limit' => 'required'
        ],
        [
            'grade.required' => 'Tingkatan Wajib di Isi',
            'name.required' => 'Nama Kelas Wajib di Isi',
            'name.unique' => 'Kelas sudah ada',
            'limits.required' => 'Jumlah Kelas Wajib Di Isi'
        ]
    );
    $classes = [

```

```

        'grade' => $request->grade,
        'expertise_id' => $request->expertise,
        'name' => $request->name,
        'limit' => $request->limit,
    ];
    Classes::create($classes);
    return redirect()->to('/classes')->with('success', 'Berhasil menambahkan
kelas');
}
public function edit(string $id)
{
    $classes = Classes::find($id);
    $expertises = Expertise::all();
    // dd($expertises);
    $title = 'Kelas';
    return view('master/classes/edit', compact('expertises', 'classes', 'title'));
}
public function update(Request $request, string $id)
{
    $request->validate(
        [
            'grade' => 'required',
            'expertise' => 'required',
            'name' => 'required',
            'limit' => 'required'
        ],
        [
            'grade.required' => 'Tingkatan Wajib di Isi',
            'name.required' => 'Nama Kelas Wajib di Isi',
            'name.unique' => 'Kelas sudah ada',
            'limits.required' => 'Jumlah Kelas Wajib Di Isi'
        ]
    );
}

```

```

        ]
    );
    $classes = [
        'grade' => $request->grade,
        'expertise_id' => $request->expertise,
        'name' => $request->name,
        'limit' => $request->limit,
    ];
    Classes::where('id', $id)->update($classes);
    return redirect()->to('/classes')->with('success', 'Berhasil Edit kelas');
}
public function destroy(string $id)
{
    Classes::where('id', $id)->delete();
    return redirect()->to('/classes')->with('success', 'Berhasil Hapus kelas');
}
}

```

### 3. Classroom

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Room;
```

```
use App\Models\User;
```

```
use App\Models\Hours;
```

```
use App\Models\Course;
```

```
use App\Models\Classes;
```

```
use App\Models\Courses;
```

```
use App\Models\Student;
```

```
use App\Models\Teacher;
```



```

use App\Models\Schedule;
use App\Models\Classroom;
use App\Models\Expertise;
use App\Models\SchoolYear;
use GuzzleHttp\Psr7\Query;
use Illuminate\Http\Request;
use Illuminate\Validation\Rule;
use App\Models\StudentClassroom;
use Illuminate\Routing\Controller;
use Illuminate\Support\Facades\DB;
use League\CommonMark\Extension\Table\Table;

class ClassroomController extends Controller
{
    // function __construct()
    // {
    //     // dd('oke');
    //     $this->middleware('permission:data-kelas|data-kelas-create|data-kelas-
edit|data-kelas-delete', ['only' => ['index', 'show']]);
    //     $this->middleware('permission:data-kelas-create', ['only' => ['create',
'store']]);
    //     $this->middleware('permission:data-kelas-edit', ['only' => ['edit', 'update']]);
    //     $this->middleware('permission:data-kelas-delete', ['only' => ['destroy']]);
    // }

    // $data = Classroom::with('studentClassroom.student')->find($id);
    // $in = $data->studentClassroom->map(function ($data) {
    //     return $data->student_id;
    // });
    // $st = Student::find($in);

```

```

public function index()
{
    $user = auth()->user();
    $school_year = SchoolYear::all();
    $year = request()->input('year');
    $query = Classroom::query();
    if (isset($year)) {
        $query->whereHas('schoolYear', function ($q) use ($year) {
            return $q->where('id', $year);
        });
    }

    if ($user->hasRole(['Wali Kelas'])) {
        $query->where('teacher_id', $user->teacher->id);
    }

    $datas = $query->orderBy('id', 'DESC')->get();
    $title = 'Data Kelas';
    return view('data/data_class/index', compact('school_year', 'datas', 'title'));
}

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    $years = SchoolYear::all();
    $datas = Classroom::all();
    $class = Classes::all();
}

```

```

$teacher = Teacher::all();
$title = 'Data Kelas';
return view('data/data_class/create', compact('years', 'datas', 'class', 'teacher',
'title'));
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    $request->validate(
        [
            'teacher' => 'required',
            // 'teacher' => 'required|unique:classrooms,teacher_id',
            // 'expertise' => 'required|unique:classrooms,expertise_id',
            // 'class' => 'required|unique:classrooms,class_id',
            'class' => 'required',
            'year' => 'required|numeric',
            // 'schoolyears' => 'required',
        ],
        [
            // 'teacher.unique' => 'Wali Kelas Sudah Ada',
            // 'expertise.unique' => 'Jurusan Sudah Ada',
            // 'class.unique' => 'kelas Sudah Ada'
        ]
    );
    $datas = [
        'class_id' => $request->class,
        'teacher_id' => $request->teacher,
        'school_year_id' => $request->year,
    ];
}

```

```

        // 'expertise_id' => $request->expertise,
        // 'schoolyear_id' => $request->schoolyears
    ];

    $insert = Classroom::create($datas);

    $insertClassTeahcers = Classroom::where('id', $request->classes)->update([
        'teacher_id' => $insert->id
    ]);

    $user = User::join('teachers', 'users.id', '=', 'teachers.user_id')
        ->select('users.*')
        ->where('teachers.id', $request->teacher)
        ->first();
    $user->assignRole('Wali Kelas');

    return redirect()->to('dataclass')->with('success', 'Berhasil menambahkan
    Kelas');
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(string $id)
{
    $datas = Classroom::find($id);
    $expertise = Expertise::all();
    // $schoolyears = SchoolYears::all();
    $teacher = Teacher::all();
    $class = Classes::all();
    $title = 'Data Kelas';
}

```

```

        return view('data/data_class/edit', compact('datas', 'expertise', 'class', 'teacher',
'title'));
    }

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, string $id)
{
    $request->validate(
        [
            'teacher' => 'required',
            'expertise' => 'required',
            'class' => 'required',
            // 'schoolyears' => 'required',
        ]
    );
    $datas = [
        'teacher_id' => $request->teacher,
        'expertise_id' => $request->expertise,
        'class_id' => $request->class,
        // 'schoolyear_id' => $request->schoolyears
    ];
    Classroom::where('id', $id)->update($datas);
    return redirect()->to('dataclass')->with('success', 'Berhasil edit Kelas');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{

```

```

Classroom::where('id', $id)->delete();
return redirect()->to('dataclass')->with('success', 'Berhasil hapus Kelas');
}

public function data($id)
{
    $data = Classroom::with('studentClassroom.student')->findOrFail($id);
    $title = 'Data Kelas';
    return view('/data/data_class/classroom/classroom', compact('data', 'title'));
}

//DETAIL KELAS
//DATA SISWA
public function classroom($id)
{
    $limit = Classroom::with('classID')->find($id)->first();
    $maxdata = $limit->classID->limit;
    $classroomId = $limit->id;
    // dd($maxdata);

    // $userId = auth()->user()->id;
    // $teacher = Teacher::where('user_id', $userId)->first();

    $data = Classroom::with('students')->findOrFail($id);
    // dd($data);
    // $data = Student::where('classroom_id', $classroomId)->get();
    $title = 'Data Kelas';
    return view('/data/data_class/classroom/classroom', compact('data', 'title'));
}

```

```

public function addStudent($id)
{

    $data = Classroom::findOrFail($id);
    // $in = $data->id->map(function ($data) {
    //     return $data->id;
    // });
    $in = $data->id;
    // dd($in);
    $student = Student::whereNull('classroom_id')->orderBy('id', 'DESC')->get();
    // dd($student);
    $title = 'Data Kelas';
    return view('/data/data_class/classroom/add_student', compact('student', 'data',
'title'));
}

```

```

public function storeStudent(Request $request, string $id)
{
    $datas = Classroom::find($request->input('classroom_id'));
    $limit = Classroom::with('classID')->find($id);
    $maxData = $limit->classID->limit;
    $existingDataCount = StudentClassroom::where('classroom_id', $id)-
>count();
    if ($existingDataCount >= $maxData) {
        return redirect()->to('dataclass/classroom/' . $id)->with('success', 'Jumlah
Siswa Sudah Mencapai Batas');
    }
    // dd($maxData);
    $request->validate(
    [

```

```

        'student' => 'required|unique:student_classroom,student_id',
    ],
    [
        'student.required' => 'Siswa Sudah ada di dalam kelas',
        'student.unique' => 'Siswa Sudah ada di dalam kelas',
    ]
);

$student = Student::find($request->input('student'));
$student->classroom_id = $limit->id;
$student->save();
return redirect()->to('dataclass/classroom/' . $id)->with('success', 'Berhasil
menambahkan Kelas');
}

public function destroyStudent(string $id)
{
    $classroom = StudentClassroom::find($id);
    $in = $classroom->classroom_id;
    StudentClassroom::where('id', $id)->delete();
    return redirect()->to('dataclass/classroom/' . $in)->with('success', 'Berhasil
hapus siswa');
}

public function detailStudent(string $id)
{
    // $data = Student::find($id);
    $data = Classroom::with('studentClassroom.student')->find($id);
    $in = $data->studentClassroom->map(function ($data) {
        return $data->student_id;
    });
}

```



```

    });
    $st = Student::find($in);
    // $datas = Student::whereIn('id', $in)->get();
    // $detail = $datas->student->function($a);
    // dd($st);
    // $detail = $data->studentClassroom->student;
    // dd($data->toArray());
    $title = 'Data Kelas';
    return view('/data/data_class/classroom', compact('st', 'title'));
}

```

```

//JADWAL PELAJARAN

```

```

public function schedule($id)
{
    $data = Classroom::with('schedule.courses')->orderBy('id', 'DESC')-
>findOrFail($id);
    $a = $data->schedule;
    // $b = $a->courses->id;
    // dd($data);
    $teacher = Teacher::all();
    $title = 'Data Kelas';
    return view('/data/data_class/schedule/schedule', compact('teacher', 'data',
'title'));
}

```

```

public function addSchedule(Request $request, $id)
{
    // $teacher = Teacher::find($id);
    // $courses = $teacher->courses;

```

```

    $data = Classroom::with('schoolYear')->findOrFail($id);
    $hours = Hours::all();
    $courses = Course::all();
    $teachers = Teacher::all();

    $title = 'Data Kelas';
    return view('/data/data_class/schedule/add_schedule', compact('teachers',
'courses', 'data', 'hours', 'title'));
}

public function storeSchedule(Request $request, string $id)
{

    $data = Classroom::find($id);
    // dd($data->school_year_id);
    $validated = $request->validate(
        [
            'courses' => 'required',
            'teachers' => 'required',
        ]
    );
    $datas = [
        'course_id' => $request->courses,
        'school_year_id' => $data->school_year_id,
        'classroom_id' => $data->id
    ];
    $courseTeacher = Courses::create($datas);
    $courseTeacher->teachers()->attach([$validated['teachers']]);

    $courseId = $courseTeacher->id;

```

```

$datas = [
    'day' => $request->day,
    'hour_id' => $request->hour,
    'courses_id' => $courseTeacher->id,
    'classroom_id' => $data->id
];

Schedule::create($datas);
return redirect()->to('dataclass/schedule/' . $id)->with('success', 'Berhasil
menambahkan Jadwal');
}

public function destroySchedule(string $id)
{
    $classroom = Schedule::with('Classroom')->findOrFail($id);
    $schedule = $classroom->classroom_id;
    // dd($schedule);
    Schedule::where('id', $id)->delete();
    return redirect()->to('dataclass/schedule/' . $schedule)->with('success',
'Berhasil hapus jadwal pelajaran');
}
}

```

#### 4. Score

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Room;
```

```
use App\Models\User;
```

```
use App\Models\Hours;
```

```

use App\Models\Course;
use App\Models\Classes;
use App\Models\Courses;
use App\Models\Student;
use App\Models\Teacher;
use App\Models\Schedule;
use App\Models\Classroom;
use App\Models\Expertise;
use App\Models\SchoolYear;
use GuzzleHttp\Psr7\Query;
use Illuminate\Http\Request;
use Illuminate\Validation\Rule;
use App\Models\StudentClassroom;
use Illuminate\Routing\Controller;
use Illuminate\Support\Facades\DB;
use League\CommonMark\Extension\Table\Table;

class ClassroomController extends Controller
{
    // function __construct()
    // {
    //     // dd('oke');
    //     $this->middleware('permission:data-kelas|data-kelas-create|data-kelas-
edit|data-kelas-delete', ['only' => ['index', 'show']]);
    //     $this->middleware('permission:data-kelas-create', ['only' => ['create',
'store']]);
    //     $this->middleware('permission:data-kelas-edit', ['only' => ['edit', 'update']]);
    //     $this->middleware('permission:data-kelas-delete', ['only' => ['destroy']]);
    // }

    // $data = Classroom::with('studentClassroom.student')->find($id);

```

```

// $in = $data->studentClassroom->map(function ($data) {
//     return $data->student_id;
// });
// $st = Student::find($in);

public function index()
{
    $user = auth()->user();
    $school_year = SchoolYear::all();
    $year = request()->input('year');
    $query = Classroom::query();
    if (isset($year)) {
        $query->whereHas('schoolYear', function ($q) use ($year) {
            return $q->where('id', $year);
        });
    }

    if ($user->hasRole(['Wali Kelas'])) {
        $query->where('teacher_id', $user->teacher->id);
    }

    $datas = $query->orderBy('id', 'DESC')->get();
    $title = 'Data Kelas';
    return view('data/data_class/index', compact('school_year', 'datas', 'title'));
}

/**
 * Show the form for creating a new resource.
 */

```

```

public function create()
{
    $years = SchoolYear::all();
    $datas = Classroom::all();
    $class = Classes::all();
    $teacher = Teacher::all();
    $title = 'Data Kelas';
    return view('data/data_class/create', compact('years', 'datas', 'class', 'teacher',
'title'));
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    $request->validate(
        [
            'teacher' => 'required',
            // 'teacher' => 'required|unique:classrooms,teacher_id',
            // 'expertise' => 'required|unique:classrooms,expertise_id',
            // 'class' => 'required|unique:classrooms,class_id',
            'class' => 'required',
            'year' => 'required|numeric',
            // 'schoolyears' => 'required',
        ],
        [
            // 'teacher.unique' => 'Wali Kelas Sudah Ada',
            // 'expertise.unique' => 'Jurusan Sudah Ada',
            // 'class.unique' => 'kelas Sudah Ada'
        ]
    );
}

```

```

);
$datas = [
    'class_id' => $request->class,
    'teacher_id' => $request->teacher,
    'school_year_id' => $request->year,
    // 'expertise_id' => $request->expertise,
    // 'schoolyear_id' => $request->schoolyears
];

$insert = Classroom::create($datas);

$insertClassTeachers = Classroom::where('id', $request->classes)->update([
    'teacher_id' => $insert->id
]);

$user = User::join('teachers', 'users.id', '=', 'teachers.user_id')
    ->select('users.*')
    ->where('teachers.id', $request->teacher)
    ->first();
$user->assignRole('Wali Kelas');

return redirect()->to('dataclass')->with('success', 'Berhasil menambahkan
Kelas');
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(string $id)
{
    $datas = Classroom::find($id);

```

```

    $expertise = Expertise::all();
    // $schoolyears = SchoolYears::all();
    $teacher = Teacher::all();
    $class = Classes::all();
    $title = 'Data Kelas';
    return view('data/data_class/edit', compact('datas', 'expertise', 'class', 'teacher',
'title'));
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, string $id)
{
    $request->validate(
        [
            'teacher' => 'required',
            'expertise' => 'required',
            'class' => 'required',
            // 'schoolyears' => 'required',
        ]
    );
    $datas = [
        'teacher_id' => $request->teacher,
        'expertise_id' => $request->expertise,
        'class_id' => $request->class,
        // 'schoolyear_id' => $request->schoolyears
    ];
    Classroom::where('id', $id)->update($datas);
    return redirect()->to('dataclass')->with('success', 'Berhasil edit Kelas');
}

```



```

/**
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{
    Classroom::where('id', $id)->delete();
    return redirect()->to('dataclass')->with('success', 'Berhasil hapus Kelas');
}

public function data($id)
{
    $data = Classroom::with('studentClassroom.student')->findOrFail($id);
    $title = 'Data Kelas';
    return view('/data/data_class/classroom/classroom', compact('data', 'title'));
}

//DETAIL KELAS
//DATA SISWA
public function classroom($id)
{
    $limit = Classroom::with('classID')->find($id)->first();
    $maxdata = $limit->classID->limit;
    $classroomId = $limit->id;
    // dd($maxdata);

    // $userId = auth()->user()->id;
    // $teacher = Teacher::where('user_id', $userId)->first();

    $data = Classroom::with('students')->findOrFail($id);
    // dd($data);
}

```

```

    // $data = Student::where('classroom_id', $classroomId)->get();
    $title = 'Data Kelas';
    return view('/data/data_class/classroom/classroom', compact('data', 'title'));
}

public function addStudent($id)
{

    $data = Classroom::findOrFail($id);
    // $in = $data->id->map(function ($data) {
    //     return $data->id;
    // });
    $in = $data->id;
    // dd($in);
    $student = Student::whereNull('classroom_id')->orderBy('id', 'DESC')->get();
    // dd($student);
    $title = 'Data Kelas';
    return view('/data/data_class/classroom/add_student', compact('student', 'data',
'title'));
}

public function storeStudent(Request $request, string $id)
{
    $datas = Classroom::find($request->input('classroom_id'));
    $limit = Classroom::with('classID')->find($id);
    $maxData = $limit->classID->limit;
    $existingDataCount = StudentClassroom::where('classroom_id', $id)-
>count();
    if ($existingDataCount >= $maxData) {

```

```

        return redirect()->to('dataclass/classroom/' . $id)->with('success', 'Jumlah
Siswa Sudah Mencapai Batas');
    }
    // dd($maxData);
    $request->validate(
        [
            'student' => 'required|unique:student_classroom,student_id',
        ],
        [
            'student.required' => 'Siswa Sudah ada di dalam kelas',
            'student.unique' => 'Siswa Sudah ada di dalam kelas',
        ]
    );

    $student = Student::find($request->input('student'));
    $student->classroom_id = $limit->id;
    $student->save();
    return redirect()->to('dataclass/classroom/' . $id)->with('success', 'Berhasil
menambahkan Kelas');
}

public function destroyStudent(string $id)
{
    $classroom = StudentClassroom::find($id);
    $in = $classroom->classroom_id;
    StudentClassroom::where('id', $id)->delete();
    return redirect()->to('dataclass/classroom/' . $in)->with('success', 'Berhasil
hapus siswa');
}

public function detailStudent(string $id)

```

```

{

// $data = Student::find($id);
$data = Classroom::with('studentClassroom.student')->find($id);
$in = $data->studentClassroom->map(function ($data) {
    return $data->student_id;
});
$st = Student::find($in);
// $datas = Student::whereIn('id', $in)->get();
// $detail = $datas->student->function($a);
// dd($st);
// $detail = $data->studentClassroom->student;
// dd($data->toArray());
$title = 'Data Kelas';
return view('/data/data_class/classroom', compact('st', 'title'));
}

```

```

//JADWAL PELAJARAN
public function schedule($id)
{
    $data = Classroom::with('schedule.courses')->orderBy('id', 'DESC')-
>findOrFail($id);
    $a = $data->schedule;
    // $b = $a->courses->id;
    // dd($data);
    $teacher = Teacher::all();
    $title = 'Data Kelas';
    return view('/data/data_class/schedule/schedule', compact('teacher', 'data',
'title'));
}

```

```

public function addSchedule(Request $request, $id)
{

    // $teacher = Teacher::find($id);
    // $courses = $teacher->courses;
    $data = Classroom::with('schoolYear')->findOrFail($id);
    $hours = Hours::all();
    $courses = Course::all();
    $teachers = Teacher::all();

    $title = 'Data Kelas';
    return view('/data/data_class/schedule/add_schedule', compact('teachers',
'courses', 'data', 'hours', 'title'));
}

```

```

public function storeSchedule(Request $request, string $id)
{

    $data = Classroom::find($id);
    // dd($data->school_year_id);
    $validated = $request->validate(
        [
            'courses' => 'required',
            'teachers' => 'required',
        ]
    );
    $datas = [
        'course_id' => $request->courses,
        'school_year_id' => $data->school_year_id,
        'classroom_id' => $data->id
    ]
}

```

```

];
$courseTeacher = Courses::create($datas);
$courseTeacher->teachers()->attach([$validated['teachers']]);

$courseId = $courseTeacher->id;

$datas = [
    'day' => $request->day,
    'hour_id' => $request->hour,
    'courses_id' => $courseTeacher->id,
    'classroom_id' => $data->id
];

Schedule::create($datas);
return redirect()->to('dataclass/schedule/' . $id)->with('success', 'Berhasil
menambahkan Jadwal');
}

public function destroySchedule(string $id)
{
    $classroom = Schedule::with('Classroom')->findOrFail($id);
    $schedule = $classroom->classroom_id;
    // dd($schedule);
    Schedule::where('id', $id)->delete();
    return redirect()->to('dataclass/schedule/' . $schedule)->with('success',
'Berhasil hapus jadwal pelajaran');
}
}
}
5. Student
<?php

```

```

namespace App\Http\Controllers;

use App\Models\Classes;
use App\Models\Classroom;
use App\Models\Course;
use App\Models\Regency;
use App\Models\Student;
use App\Models\Village;
use App\Models\District;
use App\Models\Province;
use App\Models\Religion;
use Illuminate\Http\Request;
use Psy\Readline\Hoa\Console;
use App\Models\Expertise;
use App\Models\User;
use Illuminate\Contracts\View\View;
use Illuminate\Routing\Controller;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Symfony\Contracts\Service\Attribute\Required;

class StudentController extends Controller
{
    function __construct()
    {
        $this->middleware('permission:student', ['only' => ['index']]);
        $this->middleware('permission:student-add', ['only' => ['create', 'store', 'edit',
'update', 'destroy', 'show']]);
    }
    /**
     * Display a listing of the resource.

```

```

*/

public function index()
{
    // $user_auth = Auth::user();

    // if ($user_auth->hasRole('Wali Kelas')) {
    //     $students = Student::select('id', 'NISN', 'name', 'gender', 'date_of_birth',
'phone_number', 'expertise_id')
    //         ->where('class_id', $user_auth->teacher->class_teacher->class_id)
    //         ->paginate(2);
    // } else {
    //     $students = Student::select('id', 'NISN', 'name', 'gender', 'date_of_birth',
'phone_number', 'expertise_id')->paginate(2);
    // }

    // $classes = Classes::all();
    // $id = $classes->id;
    // dd($id);
    // $a = Student::find(20);
    // $b = $a->classroom->classID->name;
    // dd($b);
    $students = Student::orderBy('id', 'desc')->paginate(12);

    // dd($students);

    $title = 'Students';
    return view('student/students', compact('students', 'title'));
}

/**

```



```

* Show the form for creating a new resource.
*/
public function create()
{
    $provinces = Province::all();
    $religions = Religion::all();
    $courses = Course::all();
    $title = "Students";
    // $classes = Classes::all();
    // $expertise_program = Expertise::all();
    $users = User::all();
    return view('/student/formstudent', compact('provinces', 'religions', 'courses',
'title', 'users'));
}

/**
* Store a newly created resource in storage.
*/
public function store(Request $request)
{
    $request->validate(
        [
            'NISN' => 'required|numeric|unique:students,NISN',
            'name' => 'required',
            'gender' => 'required',
            'religion_id' => 'required',
            'place_of_birth' => 'required',
            'date_of_birth' => 'required',
            'email' => 'required|unique:students,email',
            'phone_number' => 'required|numeric|unique:students,phone_number',
            'address' => 'required',

```

```
'province' => 'required',
'regency' => 'required',
'district' => 'required',
'village' => 'required',
'origin_of_school' => 'required',
// 'expertise_program_id' => 'required',
// 'class' => 'required',
'father_name' => 'required',
'father_work' => 'required',
'father_phone_number' =>
'required|numeric:student,father_phone_number',
'mother_name' => 'required',
'mother_work' => 'required',
'mother_phone_number' =>
'required|numeric:student,mother_phone_number',
'parent_address' => 'required',
'wali_name' => 'required',
'wali_work' => 'required',
'wali_address' => 'required',
'wali_phone_number' => 'required|numeric:student,wali_phone_number'
],
[
'NISN.required' => 'NISN wajib di isi',
'NISN.numeric' => 'NISN wajib di isi dengan angka',
'NISN.unique' => 'NISN sudah digunakan',
'name.required' => 'Nama wajib di isi',
'gender.required' => 'Jenis kelamin wajib di isi',
'religion_id.required' => 'Agama wajib di isi',
'place_of_birth.required' => 'Tempat lahir wajib di isi',
'date_of_birth.required' => 'Tanggal lahir wajib di isi',
'email.required' => 'Email Wajib di isi',
```

```
'phone_number.unique' => 'Nomor HP sudah digunakan',
'phone_number.required' => 'Nomor HP wajib di isi',
'phone_number.numeric' => 'Nomor HP wajib di isi dengan angka',
'address.required' => 'Alamat wajib di isi',
'province.required' => 'Provinsi wajib di isi',
'regency.required' => 'Kota/Kabupaten wajib di isi',
'district.required' => 'Kecamatan wajib di isi',
'village.required' => 'Kelurahan wajib di isi',
'origin_of_school.required' => 'Asal sekolah wajib di isi',
// 'expertise_program_id.required' => 'Program keahlian wajib di isi',
'father_name.required' => 'Nama ayah wajib di isi',
'father_work.required' => 'Pekerjaan ayah wajib di isi',
'father_phone_number.required' => 'Nomor HP ayah wajib di isi',
'father_phone_number.numeric' => 'Nomor HP ayah wajib di isi dengan
angka',
'mother_name.required' => 'Nama ibu wajib di isi',
'mother_work.required' => 'Pekerjaan ibu wajib di isi',
'mother_phone_number.required' => 'Nomor HP ibu wajib di isi',
'mother_phone_number.numeric' => 'Nomor HP ibu wajib di isi dengan
angka',
'parent_address.required' => 'Alamat orang tua wajib di isi',
'wali_name.required' => 'Nama wali wajib di isi',
'wali_work.required' => 'Nama wali wajib di isi',
'wali_address.required' => 'Nama wali wajib di isi',
'wali_phone_number.required' => 'Nomor HP Wali wajib di isi',
'wali_phone_number.numeric' => 'Nomor HP Wali wajib di isi dengan
angka'
]
);

$insertUser = User::create([
```

```
'name' => $request->name,  
'email' => $request->email,  
'password' => Hash::make('12345')  
]);
```

```
//ADD DATA SISWA TO TABLE
```

```
$students = [  
    'NISN' => $request->NISN,  
    'user_id' => $insertUser->id,  
    'name' => $request->name,  
    'gender' => $request->gender,  
    'religion_id' => $request->religion_id,  
    'place_of_birth' => $request->place_of_birth,  
    'date_of_birth' => $request->date_of_birth,  
    'email' => $request->email,  
    'phone_number' => $request->phone_number,  
    'address' => $request->address,  
    'province' => $request->province,  
    'regency' => $request->regency,  
    'district' => $request->district,  
    'village' => $request->village,  
    'origin_of_school' => $request->origin_of_school,  
    // 'expertise_id' => $request->expertise_program_id,  
    'class_id' => $request->class,  
    'father_name' => $request->father_name,  
    'father_work' => $request->father_work,  
    'father_phone_number' => $request->father_phone_number,  
    'mother_name' => $request->mother_name,  
    'mother_work' => $request->mother_work,  
    'mother_phone_number' => $request->mother_phone_number,  
    'parent_address' => $request->parent_address,
```

```

        'wali_name' => $request->wali_name,
        'wali_work' => $request->wali_work,
        'wali_address' => $request->wali_address,
        'wali_phone_number' => $request->wali_phone_number
    ];
    Student::create($students);
    $insertUser->assignRole('Siswa');
    return redirect()->to('/students')->with('success', 'Berhasil menambahkan data
siswa');
}

/**
 * Display the specified resource.
 */
public function show(string $id)
{
    $students = Student::find($id);
    $title = 'students';
    return view('student/showstudent', compact('students', 'title'));
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(Request $request, int $id)
{
    $student = Student::find($id);
    // $expertise_program = Expertise::all();
    $provinces = Province::all();
    // $regencies = Regency::all();

```

```

$regencies = Regency::where('province_id', $student->province)->get();
$districts = District::where('regency_id', $student->regency)->get();
$villages = Village::where('district_id', $student->district)->get();
$religions = Religion::all();
$courses = Course::all();
// $classes = Classes::all();
$title = 'edit';

return view('/student/editstudent', compact('student', 'provinces', 'regencies',
'districts', 'villages', 'religions', 'title'));
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, string $id)
{

$request->validate(
    [
        // 'NISN' => "",
        'name' => 'required',
        'gender' => 'required',
        'religion_id' => 'required',
        'place_of_birth' => 'required',
        'date_of_birth' => 'required',
        'email' => 'required',
        'phone_number' => 'required|numeric:student,phone_number',
        'address' => 'required',
        'province' => 'required',
    ]
);
}

```

```

        'regency' => 'required',
        'district' => 'required',
        'village' => 'required',
        'origin_of_school' => 'required',
        'expertise_program_id' => 'required',
        'father_name' => 'required',
        'father_work' => 'required',
        'father_phone_number' =>
'required|numeric:student,father_phone_number',
        'mother_name' => 'required',
        'mother_work' => 'required',
        'mother_phone_number' =>
'required|numeric:student,mother_phone_number',
        'parent_address' => 'required',
        'wali_name' => 'required',
        'wali_work' => 'required',
        'wali_address' => 'required',
        'wali_phone_number' => 'required|numeric:student,wali_phone_number'
    ]
);

```

```
//EDIT DATA SISWA TO TABLE
```

```

$students = [
    // 'NISN' => $request->NISN,
    'name' => $request->name,
    'gender' => $request->gender,
    'religion_id' => $request->religion_id,
    'place_of_birth' => $request->place_of_birth,
    'date_of_birth' => $request->date_of_birth,
    'email' => $request->email,
    'phone_number' => $request->phone_number,

```

```

        'address' => $request->address,
        'province' => $request->province,
        'regency' => $request->regency,
        'district' => $request->district,
        'village' => $request->village,
        'origin_of_school' => $request->origin_of_school,
        'expertise_id' => $request->expertise_program_id,
        'class_id' => $request->class,
        'father_name' => $request->father_name,
        'father_work' => $request->father_work,
        'father_phone_number' => $request->father_phone_number,
        'mother_name' => $request->mother_name,
        'mother_work' => $request->mother_work,
        'mother_phone_number' => $request->mother_phone_number,
        'parent_address' => $request->parent_address,
        'wali_name' => $request->wali_name,
        'wali_work' => $request->wali_work,
        'wali_address' => $request->wali_address,
        'wali_phone_number' => $request->wali_phone_number
    ];
    Student::where('id', $id)->update($students);
    return redirect()->to('/students')->with('success', 'Berhasil edit data siswa');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{
    Student::where('id', $id)->delete();
    return redirect()->to('/students')->with('success', 'Berhasil hapus data siswa');
}

```



```
}
```

```
public function getregency(Request $request)
```

```
{
```

```
    $id_province = $request->id_province;
```

```
    $regencies = Regency::where('province_id', $id_province)->get();
```

```
    foreach ($regencies as $regency_id) {
```

```
        echo "<option value='{$regency_id->id}'>{$regency_id->name}</option>";
```

```
    }
```

```
}
```

```
public function getdistrict(Request $request)
```

```
{
```

```
    $id_regency = $request->id_regency;
```

```
    $districts = District::where('regency_id', $id_regency)->get();
```

```
    foreach ($districts as $district_id) {
```

```
        echo "<option value='{$district_id->id}'>{$district_id->name}</option>";
```

```
    }
```

```
}
```

```
public function getvillage(Request $request)
```

```
{
```

```
    $id_district = $request->id_district;
```

```
    $villages = Village::where('district_id', $id_district)->get();
```

```
        foreach ($villages as $village_id) {
            echo "<option value='$village_id->id'$village_id->name</option>";
        }
    }
}
```

## 6. Teacher

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\User;
use App\Models\Course;
use App\Models\Regency;
use App\Models\Teacher;
use App\Models\Village;
use App\Models\District;
use App\Models\Position;
use App\Models\Province;
use App\Models\Religion;
use App\Models\Positions;
use Illuminate\Http\Request;
use App\Models\Expertise;
use Illuminate\Routing\Controller;
use Illuminate\Support\Facades\Hash;
```

```
class TeachersController extends Controller
```

```
{
    function __construct()
    {
```

```

    $this->middleware('permission:teacher', ['only' => ['index']]);
    $this->middleware('permission:teacher-add', ['only' => ['create', 'store', 'edit',
'update', 'destroy', 'show']]);
}
/**
 * Display a listing of the resource.
 */
public function index()
{
    $title = 'Data Guru';
    $teachers = Teacher::select('id', 'NIP', 'name', 'educational_level',
'lesson_teacher', 'phone_number')->paginate(4);

    return view('/teacher/teachers', compact('teachers', 'title'));
}

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    $user = User::all();
    $provinces = Province::all();
    $religions = Religion::all();
    $title = "Teachers";
    $expertise_program = Expertise::all();
    $positions = Positions::all();
    // dd($positions);
    return view('/teacher/formteacher', compact('user', 'positions', 'provinces',
'religions', 'title', 'expertise_program'));
}

```

```

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{

    $request->validate(
        [
            'NIP' => 'required|numeric|unique:teachers,NIP',
            'name' => 'required',
            'educational_level' => 'required',
            // => 'required',
            // 'course' => 'required',
            'lesson_teacher' => 'required',
            'gender' => 'required',
            'religion_id' => 'required',
            'place_of_birth' => 'required',
            'date_of_birth' => 'required',
            'phone_number' => 'required|numeric|unique:teachers,phone_number',
            'email' => 'required',
            'address' => 'required',
            'province' => 'required',
            'regency' => 'required',
            'district' => 'required',
            'village' => 'required'
        ],
        [
            'NIP.required' => 'NIP wajib di isi',
            'NIP.numeric' => 'NIP wajib di isi dengan angka',

```

```
'NIP.unique' => 'NIP sudah digunakan',
'name' => 'Nama wajib di isi',
'educational_level.required' => 'Pendidikan wajib di isi',
// 'position_id.required' => 'Jabatan wajib di isi',
// 'course' => 'Mata Pelajaran wajib di isi',
'lesson_teacher.required' => 'Status Guru wajib di isi',
'gender.required' => 'Jenis Kelawajib di isi',
'religion_id.required' => 'Agama wajib di isi',
'place_of_birth.required' => ' Tempat lahir wajib di isi',
'date_of_birth.required' => ' Tanggal wajib di isi',
'phone_number.required' => 'Nomor HP wajib di isi',
'phone_number.unique' => 'Nomor HP sudah digunakan',
'phone_number.numeric' => 'Nomor HP wajib di isi dengan angka',
'email.required' => 'Email Wajib di isi',
'address.required' => 'Alamat wajib di isi',
'province.required' => 'Provinsi wajib di isi',
'regency.required' => 'Kota/Kabupaten wajib di isi',
'district.required' => 'Kecamatan wajib di isi',
'village.required' => 'Kelurahan wajib di isi'
]
);
```

```
$insertUser = User::create([
    'name' => $request->name,
    'email' => $request->email,
    'password' => Hash::make('guruub')
]);
```

```
$teachers = [
    'user_id' => $insertUser->id,
    'NIP' => $request->NIP,
```

```

        'name' => $request->name,
        'educational_level' => $request->educational_level,
        // => $request->position_id,
        'lesson_teacher' => $request->lesson_teacher,
        'gender' => $request->gender,
        'religion_id' => $request->religion_id,
        'place_of_birth' => $request->place_of_birth,
        'date_of_birth' => $request->date_of_birth,
        'phone_number' => $request->phone_number,
        'email' => $request->email,
        'address' => $request->address,
        'province' => $request->province,
        'regency' => $request->regency,
        'district' => $request->district,
        'village' => $request->village,
    ];
    // dd($teachers);
    Teacher::create($teachers);
    $insertUser->assignRole('Guru');
    return redirect()->to('/teachers')->with('success', 'Berhasil menambahkan data
guru');
}

/**
 * Display the specified resource.
 */
public function show(string $id)
{
    $teachers = Teacher::find($id);
    // dd($teachers);
}

```

```

        return view('teacher/showteacher', ['teachers' => $teachers, "title" =>
"Identitas "]);
    }

/**
 * Show the form for editing the specified resource.
 */
public function edit(string $id)
{
    $teachers = Teacher::find($id);
    $provinces = Province::all();
    $regencies = Regency::where('province_id', $teachers->province)->get();
    $districts = District::where('regency_id', $teachers->regency)->get();
    $villages = Village::where('district_id', $teachers->district)->get();
    $expertise_program = Expertise::all();
    $provinces = Province::all();
    $religions = Religion::all();
    $title = 'edit';
    $positions = Positions::all();

    return view('/teacher/editteacher', compact('teachers', 'provinces', 'religions',
'regencies', 'districts', 'villages', 'title', 'expertise_program', 'positions'));
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, string $id)
{
    $request->validate(
        [

```

```

        'name' => 'required',
        'name' => 'required',
        'educational_level' => 'required',
        // => 'required',
        'lesson_teacher' => 'required',
        'gender' => 'required',
        'religion_id' => 'required',
        'place_of_birth' => 'required',
        'date_of_birth' => 'required',
        'phone_number' => 'required|numeric:teachers,phone_number',
        'email' => 'required',
        'address' => 'required',
        'province' => 'required',
        'regency' => 'required',
        'district' => 'required',
        'village' => 'required'
    ]
);
$teachers = [
    'name' => $request->name,
    'educational_level' => $request->educational_level,
    // => $request->position_id,
    'lesson_teacher' => $request->lesson_teacher,
    'gender' => $request->gender,
    'religion_id' => $request->religion_id,
    'place_of_birth' => $request->place_of_birth,
    'date_of_birth' => $request->date_of_birth,
    'phone_number' => $request->phone_number,
    'email' => $request->email,
    'address' => $request->address,
    'province' => $request->province,

```



```

        'regency' => $request->regency,
        'district' => $request->district,
        'village' => $request->village,
    ];

    Teacher::where('id', $id)->update($teachers);
    // dd($teachers);
    return redirect()->to('/teachers')->with('success', 'Berhasil edit data guru');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{
    Teacher::where('id', $id)->delete();
    return redirect()->to('/teachers')->with('success', 'Berhasil hapus data guru');
}

public function getregency(Request $request)
{
    $id_province = $request->id_province;

    $regencies = Regency::where('province_id', $id_province)->get();

    foreach ($regencies as $regency) {
        echo "<option value='$regency->id'$regency->name</option>";
    }
}

```

```

public function getdistrict(Request $request)
{
    $id_regency = $request->id_regency;

    $districts = District::where('regency_id', $id_regency)->get();

    foreach ($districts as $district) {
        echo "<option value='$district->id'$>$district->name</option>";
    }
}

```

```

public function getvillage(Request $request)
{
    $id_district = $request->id_district;

    $villages = Village::where('district_id', $id_district)->get();

    foreach ($villages as $village) {
        echo "<option value='$village->id'$>$village->name</option>";
    }
}
}

```

## 8. User

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Contracts\Auth\MustVerifyEmail;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```

use Illuminate\Database\Eloquent\Relations\HasOne;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;
use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable implements MustVerifyEmail
{
    use HasApiTokens, HasFactory, Notifiable, HasRoles;

    // public function jabatan()
    // {
    //     return $this->belongsTo(Register::class, 'jabatan', 'id');
    // }
    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'jabatan',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */

```

```
protected $hidden = [  
    'password',  
    'remember_token',  
];  
  
/**  
 * The attributes that should be cast.  
 *  
 * @var array<string, string>  
 */  
protected $casts = [  
    'email_verified_at' => 'datetime',  
    'password' => 'hashed',  
];  
  
public function teacher(): HasOne  
{  
    return $this->hasOne(Teacher::class);  
}  
}
```