

**TUGAS AKHIR**

**RANCANG BANGUN SISTEM INFORMASI  
AKADEMIK DI SD NEGERI SIWARAK WETAN  
MENGGUNAKAN METODE WATERFALL**



**ANDRA AULIA RIZALDY**  
**19104062**

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2023**

**TUGAS AKHIR**

**RANCANG BANGUN SISTEM INFORMASI  
AKADEMIK DI SD NEGERI SIWARAK WETAN  
MENGGUNAKAN METODE WATERFALL**

***DESIGN OF AN ACADEMIC INFORMATION SYSTEM  
AT SD NEGERI SIWARAK WETAN USING THE  
WATERFALL METHOD***

Disusun Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Komputer



**ANDRA AULIA RIZALDY  
19104062**

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2023**

**HALAMAN PERSETUJUAN PEMBIMBING**

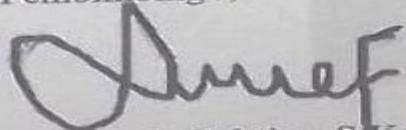
**RANCANG BANGUN SISTEM INFORMASI AKADEMIK  
DI SD NEGERI SIWARAK WETAN  
MENGGUNAKAN METODE WATERFALL**

***DESIGN OF AN ACADEMIC INFORMATION SYSTEM AT  
SD NEGERI SIWARAK WETAN USING THE  
WATERFALL METHOD***

Dipersiapkan dan Disusun oleh  
**ANDRA AULIA RIZALDY**  
**19104062**

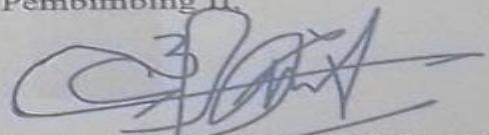
Usulan penelitian Tugas Akhir telah disetujui pada tanggal  
2 Maret 2023

Pembimbing I,



**Arief Rais Bahtiar, S.Kom., M.Kom**  
NIDN 0604119101

Pembimbing II,



**Nia Annisa Ferani T., S.Si., M.Sc**  
NIDN 0630049203

**HALAMAN PENGESAHAN TUGAS AKHIR**

**RANCANG BANGUN SISTEM INFORMASI  
AKADEMIK DI SD NEGERI SIWARAK WETAN  
MENGGUNAKAN METODE WATERFALL**

**DESIGN OF AN ACADEMIC INFORMATION SYSTEM  
AT SD NEGERI SIWARAK WETAN USING THE  
WATERFALL METHOD**

Disusun oleh

ANDRA AULIA RIZALDY

19104062

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tugas Akhir Pada Selasa,  
9 Mei 2023

Pengaji I,

Muhammad Afrizal Amrustian,  
S. Kom., M. Kom

NIDN. 0630119104

Pembimbing I,

Arief Rais Bahtiar, S.Kom.,  
M.Kom

NIDN. 0620039302

Pengaji II,

Dimas Fapny Hebrasianto  
Permadi, S.ST. M.Kom

NIDN. 0731039201

Pengaji III,

Resad Setyadi,  
S.T.,S.Si.,MMSI

NIDN. 0307057601

Pembimbing II,

Nia Annisa Ferani T., S.Si.,  
M.Sc

NIDN. 0630049203

Dekan,

Auliya Burhanuddin, S.Si.,  
M.Kom

NIK. 19820008

## **HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertandatangan di bawah ini,

**Nama mahasiswa : Andra Aulia Rizaldy**

**NIM : 19104062**

**Program Studi : Rekayasa Perangkat Lunak**

Menyatakan bahwa Tugas Akhir dengan judul berikut:

**RANCANG BANGUN SISTEM INFORMASI AKADEMIK DI SD NEGERI  
SIWARAK WETAN MENGGUNAKAN METODE WATERFALL**

Dosen Pembimbing Utama : **Arief Rais Bahtiar, S.Kom., M.Kom**

Dosen Pembimbing Pendamping : **Nia Annisa Ferani T., S.Si., M.Sc**

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Institut Teknologi Telkom Purwokerto maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan, dan penelitian Saya Sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing.
3. Dalam Karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggungjawab Saya, bukan tanggungjawab Institut Teknologi Telkom Purwokerto.
5. Pernyataan ini Saya buat dengan sesungguhnya, apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka Saya bersedia menerima Sanksi Akademik dengan pencabutan gelar yang sudah diperoleh serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Purwokerto, 20 Maret 2023,

**Yang Menyatakan**



**Andra Aulia Rizaldy**

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, atas berkat rahmat dan karunia-Nya kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Penyusunan tugas akhir ini dimaksudkan untuk memenuhi sebagian syarat-syarat guna mencapai gelar Sarjana di Institut Teknologi Telkom Purwokerto.

Penulis menyadari bahwa penulisan ini tidak dapat terselesaikan tanpa dukungan dari berbagai pihak baik moril maupun materil. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu dalam penyusunan tugas akhir ini terutama kepada:

1. Tuhan YME karena atas izin-Nya penulis dapat menyelesaikan laporan Tugas Akhir ini.
2. Kedua Orang Tua yang telah memberikan doa dan dukungan penuh.
3. Bapak Dr. Arfianto Fahmi, S.T., M.T. selaku Rektor Institut Teknologi Telkom Purwokerto.
4. Ibu Gita Fadila Fitriana, S.Kom., M.Kom. selaku Kaprodi S1 Rekayasa Perangkat Lunak.
5. Bapak Arief Rais Bahtiar, S.Kom., M.Kom. dan Ibu Nia Annisa Ferani T., S.Si., M.Sc selaku pembimbing yang telah sabar dan giat memberikan bimbingan dan arahan dalam menyelesaikan Tugas Akhir ini.
6. Seluruh Bapak/Ibu dosen Fakultas Informatika yang telah memberikan ilmu pengetahuan yang sangat bermanfaat selama masa perkuliahan.
7. Bapak/Ibu guru SD Negeri Siwarak Wetan Kabupaten Banyumas yang telah bersedia menjadi responden Tugas Akhir ini.
8. Yasirli Amriya Isnaini terimakasih telah menemani penulis di

hari hari yang tidak mudah selama penggerjaan tugas akhir, kebersamaan yang kokoh bukanlah karena tunduk pada keadaan, melainkan karena ketabahan dalam menghadapi segala rintangan hingga akhir waktu.

9. Teman-teman terbaik saya, Adam, Dani, Teguh, Rifki, Raka, Fikri, Naryanto, Vincent, dan yang lainnya yang tidak bisa saya sebutkan satu persatu yang telah berjuang bersama, mendukung dan saling memberikan motivasi.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan segala bentuk saran serta masukan bahkan kritik yang membangun dari berbagai pihak. Semoga tugas akhir ini dapat bermanfaat bagi para pembaca dan semua pihak khususnya dalam bidang Informatika.

Purwokerto, 21 Maret 2023

Penulis



Andra Aulia Rizaldy

## DAFTAR ISI

<b>HALAMAN SAMPUL.....</b>	i
<b>LEMBAR PERSETUJUAN PEMBIMBING .....</b>	ii
<b>HALAMAN PENGESAHAN TUGAS AKHIR .....</b>	iii
<b>HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....</b>	iv
<b>KATA PENGANTAR.....</b>	v
<b>DAFTAR ISI.....</b>	vii
<b>DAFTAR TABEL .....</b>	x
<b>DAFTAR GAMBAR.....</b>	xi
<b>DAFTAR LAMPIRAN .....</b>	xiv
<b>ABSTRAK .....</b>	xv
<b>ABSTRACT .....</b>	xvi
<b>BAB I.....</b>	1
<b>PENDAHULUAN .....</b>	1
<b>1.1 Latar Belakang Masalah.....</b>	1
<b>1.2 Rumusan Masalah.....</b>	3
<b>1.3 Pertanyaan Penelitian .....</b>	3
<b>1.4 Batasan Masalah.....</b>	4
<b>1.5 Tujuan Penelitian .....</b>	4
<b>1.6 Manfaat Penelitian .....</b>	5
<b>BAB II .....</b>	6
<b>TINJAUAN PUSTAKA DAN LANDASAN TEORI .....</b>	6
<b>2.1. Tinjauan Pustaka.....</b>	6
<b>2.2 Landasan Teori.....</b>	13
<b>2.2.1 Perancangan.....</b>	13

2.2.2	Sistem Informasi Akademik.....	13
2.2.3	CodeIgniter .....	14
2.2.4	Pengujian Perangkat Lunak .....	14
2.2.5	Metode Waterfall.....	14
2.2.6	Unified Modeling Language.....	16
2.2.7	Purposive Sampling .....	<b>Error! Bookmark not defined.</b>
2.2.8	UAT .....	<b>Error! Bookmark not defined.</b>
2.2.9	Statistical Product and Service Solutions (SPSS). ....	<b>Error! Bookmark not defined.</b>
2.2.10	Skala Likert.....	<b>Error! Bookmark not defined.</b>
<b>BAB III</b>	<b>METODOLOGI PENELITIAN .....</b>	<b>21</b>
<b>3.1</b>	<b>Subjek dan Objek Penelitian.....</b>	<b>21</b>
<b>3.2</b>	<b>Alat Penelitian.....</b>	<b>21</b>
<b>3.2.1</b>	<b>Perangkat Keras.....</b>	<b>21</b>
<b>3.2.2</b>	<b>Perangkat Lunak.....</b>	<b>21</b>
<b>3.3</b>	<b>Diagram Alir Penelitian .....</b>	<b>22</b>
<b>3.3.1</b>	<b>Analisis Kebutuhan .....</b>	<b>22</b>
<b>3.3.2</b>	<b>Desain .....</b>	<b>23</b>
<b>3.3.3</b>	<b>Implementasi.....</b>	<b>23</b>
<b>3.3.4</b>	<b>Pengujian.....</b>	<b>24</b>
<b>3.3.5</b>	<b>Pemeliharaan .....</b>	<b>24</b>
<b>BAB IV</b>	<b>.....</b>	<b>26</b>
<b>HASIL DAN PEMBAHASAN</b>	<b>.....</b>	<b>26</b>
<b>4.1</b>	<b>Hasil .....</b>	<b>26</b>
<b>4.2</b>	<b>Pembahasan .....</b>	<b>26</b>
<b>4.2.1</b>	<b>Analisis Kebutuhan .....</b>	<b>27</b>

<b>4.2.2</b>	<b>Desain .....</b>	<b>75</b>
<b>4.2.3</b>	<b>Implementasi.....</b>	<b>76</b>
<b>4.2.4</b>	<b>Pengujian.....</b>	<b>95</b>
<b>4.2.5</b>	<b>Pemeliharaan .....</b>	<b>110</b>
<b>BAB V .....</b>		<b>114</b>
<b>KESIMPULAN DAN SARAN .....</b>		<b>114</b>
<b>5.1</b>	<b>Kesimpulan .....</b>	<b>114</b>
<b>5.2</b>	<b>Saran.....</b>	<b>115</b>
<b>DAFTAR PUSTAKA .....</b>		<b>116</b>
<b>LAMPIRAN.....</b>		<b>121</b>

## DAFTAR TABEL

Tabel 2. 1 Literatur review dari penelitian sebelumnya.....	10
Tabel 2. 2 Komponen-komponen Use Case Diagram.....	16
Tabel 2. 3 Komponen-komponen Squence Diagram .....	17
Tabel 2. 4 Komponen-komponen Activity Diagram.....	19
Tabel 2. 5 Skor kriteria.....	<b>Error! Bookmark not defined.</b>
Tabel 2. 6 Nilai dan bobot jawaban .....	<b>Error! Bookmark not defined.</b>
Tabel 3. 1 Perangkat Lunak .....	21
Tabel 4. 1 Use Case Specification Login Pengguna .....	29
Tabel 4. 2 Use Case Specification Melihat Dashboard Pengguna .....	30
Tabel 4. 3 Use Case Specification Melihat Jadwal Pengguna .....	30
Tabel 4. 4 Use Case Specification Mengelola Data Presensi Pengguna.....	31
Tabel 4. 5 Use Case Specification Login Admin .....	31
Tabel 4. 6 Use Case Specification Melihat Dashboard Admin.....	32
Tabel 4. 7 Use Case Specification Mengelola Data Kelas Admin.....	32
Tabel 4. 8 Use Case Specification Mengelola Data Siswa Admin .....	33
Tabel 4. 9 Use Case Specification Mengelola Data Mata Pelajaran Admin.....	34
Tabel 4. 10 Use Case Specification Mengelola Data Tahun Akademik Admin ...	35
Tabel 4. 11 Use Case Specification Mengelola Data Jadwal Pelajaran Admin....	36
Tabel 4. 12 Use Case Specification Mengelola Data Guru Admin.....	36
Tabel 4. 13 Use Case Specification Mengelola Data Semester Admin .....	37
Tabel 4. 14 Use Case Specification Melihat Data Presensi Admin .....	38
Tabel 4. 15 Use Case Specification Mengelola Data User .....	38
Tabel 4. 16 Rencana Kelas Uji dan Butir Uji.....	96
Tabel 4. 17 Task Pegujian Sistem Informasi Akademik Admin.....	96
Tabel 4. 18 Task Pengujian Sistem Informasi Akademik Pengguna .....	106
Tabel 4. 19 Daftar pertanyaan pengujian .....	<b>Error! Bookmark not defined.</b>
Tabel 4. 20 Tabel hasil pengujian UAT .....	<b>Error! Bookmark not defined.</b>
Tabel 4. 21 Pemeliharaan Website Sistem Informasi Akademik	<b>Error! Bookmark not defined.</b>

## DAFTAR GAMBAR

Gambar 1. 1 Hasil kuesioner fitur yang akan diterapkan Sistem Informasi Akademik .....	2
Gambar 4. 1 Use Case Sistem Informasi Akademik SIAwarak.....	28
Gambar 4. 2 Squence Diagram Login Pengguna .....	40
Gambar 4. 3 Squence Diagram Melihat Dashboard Pengguna.....	40
Gambar 4. 4 Squence Diagram Melihat Jadwal Pengguna .....	41
Gambar 4. 5 Squence Diagram Mengelola Data Presensi Pengguna.....	42
Gambar 4. 6 Squence Diagram Login Admin.....	44
Gambar 4. 7 Squence Diagram Melihat Dashboard Admin .....	44
Gambar 4. 8 Squence Diagram Mengelola Data Kelas Admin .....	45
Gambar 4. 9 Squence Diagram Mengelola Data Siswa Admin .....	47
Gambar 4. 10 Squence Diagram Mengelola Data Mata Pelajaran Admin .....	49
Gambar 4. 11 Squence Diagram Mengelola Data Tahun Akademik Admin.....	51
Gambar 4. 12 Squence Diagram Mengelola Data Jadwal Pelajaran Admin.....	53
Gambar 4. 13 Squence Diagram Mengelola Data Guru Admin .....	55
Gambar 4. 14 Squence Diagram Mengelola Data Semester Admin .....	56
Gambar 4. 15 Squence Diagram Melihat Data Presensi Admin.....	58
Gambar 4. 16 Squence Diagram Mengelola Data User Admin .....	59
Gambar 4. 17 Activity Diagram Login Pengguna .....	60
Gambar 4. 18 Activity Diagram Melihat Halaman Dashboard Pengguna .....	61
Gambar 4. 19 Activity Diagram Melihat Jadwal Pelajaran Pengguna.....	61
Gambar 4. 20 Activity Diagram Mengelola Data Presensi Pengguna .....	62
Gambar 4. 21 Activity Diagram Login Admin .....	63
Gambar 4. 22 Activity Diagram Melihat Halaman Dashboard Admin .....	64
Gambar 4. 23 Activity Diagram Mengelola Data Kelas Admin.....	65
Gambar 4. 24 Activity Diagram Mengelola Data Siswa Admin .....	66
Gambar 4. 25 Activity Diagram Mengelola Data Mata Pelajaran Admin .....	68
Gambar 4. 26 Activity Diagram Mengelola Data Tahun Akademik Admin .....	69

Gambar 4. 27 Activity Diagram Mengelola Data Jadwal Pelajaran Admin .....	70
Gambar 4. 28 Activity Diagram Mengelola Data Guru Admin.....	71
Gambar 4. 29 Activity Diagram Mengelola Data Semester Admin .....	72
Gambar 4. 30 Activity Diagram Melihat Data Presensi Admin .....	73
Gambar 4. 31 Activity Diagram Mengelola Data User Admin .....	74
Gambar 4. 32 SIAwarak ERD (Entity Relationship Diagram).....	75
Gambar 4. 33 Low Fidelity Sistem Informasi Akademik SIAwarak User .....	76
Gambar 4. 34 Low Fidelity Sistem Informasi Akademik SIAwarak Admin.....	76
Gambar 4. 35 Halaman Login Admin dan User .....	77
Gambar 4. 36 Potongan Code Login.....	78
Gambar 4. 37 Halaman Utama User .....	78
Gambar 4. 38 Halaman Presensi User .....	79
Gambar 4. 39 Potongan Code Presensi .....	79
Gambar 4. 40 Potongan Code Model Siswa .....	80
Gambar 4. 41 Halaman Detail Presensi User.....	81
Gambar 4. 42 Potongan Code Detail Presensi .....	81
Gambar 4. 43 Potongan Code Model Presensi .....	82
Gambar 4. 44 Halaman Tambah Presensi User .....	83
Gambar 4. 45 Potongan Code Tambah Presensi .....	83
Gambar 4. 46 Halaman Mengubah Data Presensi User.....	84
Gambar 4. 47 Potongan Code Ubah Data Presensi.....	85
Gambar 4. 48 Halaman Hapus Data Presensi User.....	86
Gambar 4. 49 Potongan Code Hapus Data Presensi .....	86
Gambar 4. 50 Potongan Code Hapus Data Presensi Alert .....	87
Gambar 4. 51 Potongan Code Unduh Data Presensi .....	87
Gambar 4. 52 Halaman Utama Admin.....	88
Gambar 4. 53 Halaman Jadwal Pelajaran Admin .....	89
Gambar 4. 54 Potongan Code Jadwal .....	89
Gambar 4. 55 Halaman Detail Jadwal Pelajaran Admin.....	90
Gambar 4. 56 Potongan Code Detail Jadwal .....	90
Gambar 4. 57 Halaman Tambah Jadwal Pelajaran Admin .....	91

Gambar 4. 58 Potongan Code Tambah Jadwal .....	92
Gambar 4. 59 Halaman Ubah Jadwal Admin.....	92
Gambar 4. 60 Potongan Code Ubah Jadwal.....	93
Gambar 4. 61 Halaman Hapus Jadwal Admin .....	94
Gambar 4. 62 Potongan Code Hapus Semua Data Jadwal .....	94
Gambar 4. 63 Potongan Code Hapus Semua Jadwal Modal.....	95
Gambar 4. 64 Feedback Sistem Informasi Akademik .....	109

## **DAFTAR LAMPIRAN**

Lampiran 1 Pertanyaan wawancara .....	121
Lampiran 2 Hasil Jawaban Pertanyaan .....	122
Lampiran 3 Hasil Survey Google Form .....	123
Lampiran 4 Dokumentasi Wawancara kepada kepala sekolah .....	124
Lampiran 5 Dokumentasi Pengisian Google Form.....	124
Lampiran 6 Dokumentasi Pengujian Sistem Informasi Akademik.....	125

## **ABSTRAK**

### **RANCANG BANGUN SISTEM INFORMASI AKADEMIK DI SD NEGERI SIWARAK WETAN MENGGUNAKAN METODE WATERFALL**

Oleh

Andra Aulia Rizaldy

19104062

Pengolahan data akademik secara konvensional dalam bidang pendidikan tidak lagi cukup untuk meningkatkan mutu pendidikan, termasuk di salah satu sekolah di Kecamatan Tambak, Kabupaten Banyumas merupakan sekolah dasar negeri dan saat ini belum memiliki sistem informasi akademik secara *online*. Pada penelitian ini bertujuan untuk membuat sistem informasi akademik agar dapat menangani proses akademik SD Siwarak Wetan Kecamatan Tambak Kabupaten Banyumas menggunakan metode *waterfall* dan melakukan evaluasi akhir sistem informasi akademik menggunakan teknik pengujian *UAT* (*User Acceptance Test*), supaya sistem informasi akademik dapat diterima dengan baik oleh SD Negeri Siwarak Wetan. Pembuatan website sistem infomasi akademik SD Negeri Siwarak Wetan menggunakan metode pengembangan *waterfall* yang dibagi menjadi lima tahapan yaitu analisis kebutuhan, desain, pengimplementasian, tes perangkat lunak, dan juga pemeliharaan. Analisis kebutuhan dibagi menjadi beberapa tahapan yaitu mengidentifikasi suatu masalah, studi literatur untuk mencari referensi, pengumpulan data, serta analisis yaitu membuat kerangka jalanya sistem informasi akademik, serta mendesain basis data kemudian diimplementasikan menggunakan Bahasa pemrograman *PHP* dan *CodeIgniter*, dengan setiap rancangan tersebut diuji dengan menggunakan metode pengujian *blackbox* untuk mengetahui setiap rancangan sudah sesuai dan berfungsi dengan baik. Pada akhirnya penelitian ini dapat menghasilkan sebuah sistem informasi yang dapat menangani sesuai dengan kebutuhan pengguna yaitu guru serta operator SD Negeri Siwarak Wetan, serta evaluasi akhir bagaimana sistem informasi akademik menggunakan metode *UAT* dengan total persentase yang dihasilkan 77,00% dan berada didalam kategori setuju.

**Kata Kunci:** Rancang Bangun, Sistem Informasi Akademik, SD Negeri Siwarak Wetan  
*Waterfall*

## **ABSTRACT**

### ***DESIGN OF AN ACADEMIC INFORMATION SYSTEM AT SD NEGERI SIWARAK WETAN USING THE WATERFALL METHOD***

Author

Andra Aulia Rizaldy

19104062

*Conventional academic data processing in the field of education is no longer sufficient to improve the quality of education, including in one of the elementary schools in Tambak District, Banyumas Regency which is a public elementary school and currently does not have an online academic information system. This study aims to create an academic information system to handle the academic process of SD Siwarak Wetan Elementary School in Tambak District, Banyumas Regency using the waterfall method and to conduct a final evaluation of the academic information system using User Acceptance Testing (UAT) technique, so that the academic information system can be well received by SD Negeri Siwarak Wetan. The creation of the academic information system website for SD Negeri Siwarak Wetan uses the waterfall development method which is divided into five stages: requirement analysis, design, implementation, software testing, and maintenance. The requirement analysis is divided into several stages such as identifying a problem, literature review to find references, data collection, and analysis which is to create the framework of the academic information system and design the database, then it is implemented using PHP and CodeIgniter programming languages, with each design being tested using black-box testing method to ensure that each design is suitable and functioning properly. In the end, this study can produce an information system that can handle user needs, namely teachers and operators of SD Negeri Siwarak Wetan, as well as a final evaluation of how the academic information system works using the UAT method with a total percentage of 77.00% which falls in the agree category.*

**Keywords:** Design, Academic Information System, SD Negeri Siwarak Wetan, Waterfall

# **BAB I**

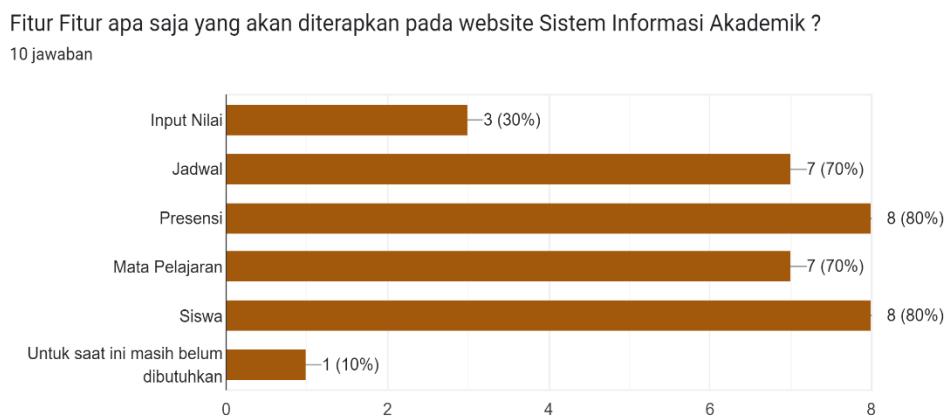
## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

*Website* memiliki potensi dan sumber daya yang dapat dimanfaatkan secara efektif dalam berbagai kegiatan, seperti pembelajaran dan administrasi di instansi pendidikan. *Website* memungkinkan instansi pendidikan untuk dengan cepat menerima dan menyampaikan informasi kepada sekolah dan guru, serta membangun kemitraan dengan orang tua atau wali murid [1]. Lembaga pendidikan dapat memanfaatkan teknologi komputer, baik itu dalam bentuk perangkat keras (*hardware*) maupun perangkat lunak (*software*), yang dikelola untuk pengambilan keputusan oleh manajemen tingkat atas di lingkungan pendidikan. Penggunaan sistem informasi untuk kegiatan internal di sebuah instansi dianggap sebagai faktor yang dapat menentukan keberhasilan dan kemajuan instansi pendidikan tersebut [2]. Ini menjadi alat penting, termasuk bagi sekolah dimana kompetensi serta keterampilan teknologi informasi dikembangkan menggunakan teknologi modern termasuk sistem informasi akademik [3]. Saat ini, penggunaan teknologi informasi dan komunikasi menjadi suatu keharusan dalam mengumpulkan, mengolah, menyimpan, dan melaporkan data akademik. Pesatnya perkembangan teknologi jaringan dan internet telah menghubungkan tidak hanya manusia dan perangkat mesin seperti komputer, laptop, tablet, dan server, tetapi juga menghubungkan interaksi antara mesin dengan mesin, yang sangat bermanfaat dalam memenuhi kebutuhan sehari-hari [4].

SD Negeri Siwarak Wetan merupakan salah satu satuan pendidikan dengan jenjang Sekolah Dasar (SD) di Desa Watuagung, Kec. Tambak, Kab. Banyumas, Jawa Tengah. SD Negeri Siwarak Wetan beralamat di Jalan Mahameru, Desa Watuagung, Kec. Tambak, Kab. Banyumas, Jawa Tengah, dengan kode pos 53196 yang berada di bawah naungan Kementerian Pendidikan dan Kebudayaan. SD Negeri Siwarak Wetan memiliki sepuluh guru dengan 172 siswa [5]. SD Negeri Siwarak Wetan juga sudah memiliki beragam fasilitas yang dapat mendukung

kegiatan belajar mengajar. Namun sistem informasi yang berada di SD Negeri Siwarak Wetan masih berjalan manual dan belum termodernisasi, sehingga sering terjadi penumpukan data siswa dan mengakibatkan sulitnya pencarian data siswa apabila sewaktu-waktu data itu dibutuhkan. Sementara itu masalah lain yang kerap timbul yaitu penjadwalan dari seluruh kelas masih belum terstruktur. Selanjutnya, terdapat sebuah kuesioner yang dilakukan kepada guru serta operator sekolah SD Negeri Siwarak Wetan mengenai fitur apa saja yang akan diterapkan pada website Sistem Informasi Akademik SD Negeri Siwarak Wetan terkait keperluan yang ada di sekolah dan didapatkan hasil yang dapat dilihat pada Gambar 1. 1.



Gambar 1. 1 Hasil kuesioner fitur yang akan diterapkan Sistem Informasi Akademik

Poin tertinggi terdapat 80% responden menyatakan bahwa Sistem Informasi Akademik terdapat fitur presensi dan siswa, selain itu 70% responden menyatakan memerlukan fitur jadwal dan mata pelajaran, sedangkan 30% responden memerlukan fitur input nilai, dan 10% responden belum membutuhkan fitur fitur tersebut. Sehingga, dari hasil kuesioner didapati bahwa keperluan fitur pada usulan dari guru dan operator sekolah yaitu terdapat presensi, siswa, mata pelajaran, dan jadwal diutamakan terlebih dahulu.

Dalam mengembangkan sebuah perangkat lunak terdapat beberapa metode yang selalu berusaha untuk memastikan perangkat lunak yang dikembangkan berjalan optimal. Salah satunya adalah *scrum*, yang merupakan metode pengembangan yang banyak digunakan, namun kurang efektif karena diperlukan

anggota tim yang berpengalaman untuk menyelesaikan proyek dengan tepat waktu. Jika tim terdiri dari anggota yang masih pemula, maka proyek dapat menghadapi kendala. Selain itu, setiap tugas harus didefinisikan secara jelas untuk mempengaruhi estimasi biaya dan waktu pengerjaan proyek [6]. Terdapat juga metode *prototype* yang merupakan teknik untuk mengumpulkan kebutuhan-kebutuhan informasi pengguna secara cepat. Metode ini digunakan untuk mendapatkan gambaran dari pemodelan aplikasi yang akan dibuat. Namun metode ini kurang efektif juga karena kurang fleksibel terhadap perubahan [7].

Dalam perancangan sistem informasi akademik, pada penelitian ini memilih menggunakan metode *waterfall* yang merupakan pengembangan yang praktis dan terstruktur serta mampu memastikan kualitas perangkat lunak tetap terjaga. Selain itu, karena model ini menghasilkan dokumentasi yang lengkap, proses pemeliharaan perangkat lunak menjadi lebih mudah [8]. Selain itu metode ini merupakan metode yang memiliki kontrol yang ketat dalam setiap tahap pengembangan, karena setiap tahap harus selesai sebelum melanjutkan ke tahap berikutnya. Hal ini dapat membantu dalam mengidentifikasi dan mengatasi masalah atau kecacatan pada tahap awal pengembangan [9].

Dengan adanya sistem yang akan dibuat bisa memudahkan proses pengolahan data informasi akademik berdasarkan kebutuhan dari SD Negeri Siwarak Wetan. Sehingga untuk mempermudah proses pengolahan data presensi dan jadwal serta adanya informasi yang berkaitan dengan pelaksanaan proposal penelitian maka penulis memilih judul “Rancang Bangun Sistem Informasi Akademik Di SD Negeri Siwarak Wetan Menggunakan Metode *Waterfall*”.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka disusun rumusan masalah ini yaitu bagaimana membangun suatu sistem informasi akademik untuk mempermudah kinerja guru serta operator sekolah untuk pengolahan data.

## 1.3 Pertanyaan Penelitian

Berdasarkan uraian di atas, penulis merumuskan beberapa pertanyaan terkait yang akan diteliti, yaitu

1. Bagaimana membangun website sistem informasi akademik SD Negeri Siwarak Wetan melalui metode *waterfall*?
2. Bagaimana mengevaluasi sistem informasi akademik supaya diterima oleh pengguna dengan metode pengujian *UAT (User Acceptance Test)*?

#### **1.4 Batasan Masalah**

Dari perumusan masalah, diperlukan batasan-batasan dalam penelitian untuk memastikan bahwa penelitian yang dilakukan relevan dengan masalah yang ingin diselesaikan. Oleh karena itu, batasan masalah penelitian diperoleh sebagai berikut:

1. Sistem ini dibuat menggunakan metode *waterfall* khusus untuk memenuhi kebutuhan akademik. Dengan guru sebagai user, sedangkan operator sekolah beserta kepala sekolah sebagai admin SD Negeri Siwarak Wetan Kecamatan Tambak Kabupaten Banyumas.
2. Setiap guru di SD Negeri Siwarak Wetan Kecamatan Tambak Kabupaten Banyumas memiliki akun *user* pada sistem yang dibuat oleh admin. Dalam sistem tersebut, guru dapat menambahkan presensi dan melihat jadwal dengan mudah.
3. Operator sekolah sebagai admin dapat menambahkan data guru, mata pelajaran, siswa, kelas, tahun ajaran, semester, akun, jadwal serta melihat presensi.
4. Mengevaluasi sistem informasi akademik menggunakan metode pengujian *UAT* supaya sistem diterima oleh pengguna.

#### **1.5 Tujuan Penelitian**

Tujuan penelitian ini berdasarkan rumusan masalah adalah sebagai berikut :

1. Membuat sistem informasi akademik agar dapat menangani proses akademik di SD Siwarak Wetan Kecamatan Tambak Kabupaten Banyumas menggunakan metode *waterfall* supaya hasilnya sesuai dengan apa yang diinginkan pengguna.
2. Mengevaluasi sistem informasi akademik menggunakan metode pengujian *UAT (User Acceptance Test)* supaya sistem diterima oleh SD Negeri Siwarak Wetan.

## **1.6 Manfaat Penelitian**

Adapun manfaat yang diharapkan dengan adanya penelitian ini adalah sebagai berikut.

1. Memberikan kemudahan bagi guru serta operator sekolah SD Negeri Siwarak Wetan Kecamatan Tambak Kabupaten Banyumas dalam hal penginputan data.
2. Sebagai referensi bagi penelitian lain dalam pengembangan perangkat lunak berbasis web.

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1. Tinjauan Pustaka

Dalam rangka meningkatkan informasi yang diperlukan dalam penelitian ini, diperlukan referensi dari penelitian-penelitian sebelumnya yang relevan dengan topik yang dibahas. Berikut adalah beberapa penelitian terkait yang telah disajikan:

Pada penelitian pertama yang dilakukan oleh Lia Dewi Saputri yang berjudul Rancang Bangun Sistem Informasi Akademik Menggunakan Metode *Waterfall* pada SMA Kosgoro Tangerang terdapat beberapa hal yang dapat dipelajari. Pada penelitian ini dengan menggunakan metode *Waterfall* agar dapat menaikan mutu pengolahan data agar lebih mudah serta lebih terstruktur. Pada penelitian tersebut menyebutkan alur penelitiannya yaitu melakukan pengumpulan data terdiri dari observasi, wawancara serta studi Pustaka. Dilanjutkan dengan model pengembangan sistem terdiri dari Analisa kebutuhan sistem, desain, *code generation, testing* dan *support*. Penelitian ini berisi tentang pembuatan web sistem informasi akademik pada SMA Kosgoro Tangerang yang memudahkan pengolahan informasi dengan baik [10].

Pada penelitian lain yang dilakukan oleh Sandra Melinda dkk dengan judul Penerapan Model *Waterfall* dalam Pengembangan Sistem Informasi Akademik Berbasis Web sebagai Sistem Pengolahan Nilai Siswa, juga terdapat beberapa hal yang dapat dijadikan acuan. Pada penelitian ini memuat tentang pengembangan website dengan metode *Waterfall* agar memudahkan pengguna dalam pengolahan web pada Sekolah Menengah Kejuruan serta dapat memberikan kemudahan pada proses penginputan nilai, Disebutkan juga beberapa keunggulan metode *waterfall* didalam penelitian ini yaitu sistematis serta berurutan. Selain itu dari segi rancangan hanya membuat usecase saja sehingga perlu adanya desain lain [11].

Sementara itu pada penelitian yang dilakukan oleh Marijan dan Siti Nurajizah yang berjudul Perancangan Sistem Informasi Akademik Berbasis Web Pada SD Islam Luqmanul Hakim Bekasi. Pada penelitian ini berisi tentang perancangan

sistem berbasis web di SD Islam Luqmanul Hakim Bekasi dengan menggunakan metode model *prototype*. Pada penelitian tersebut alasan memilih metode itu karena dapat saling berkomunikasi pada saat perancangan system, sebaliknya disisi *developer* kurang memperhatikan segi efisiensi dari sebuah algoritma [12].

Penelitian yang dilakukan oleh Faza Nadhira, Iwan Wahyudin, dan Ratih Titi Komala dengan judul Penerapan Metode *Agile Scrum* pada rancangan SisIAM4. Metode yang digunakan ialah *scrum*. Penelitian ini berdasarkan perkembangan dalam dunia informatika dan teknologi sangat pesat mengikuti perkembangan era di zaman sekarang. Maka, dengan begitu muncullah gagasan untuk membuat sistem informasi akademik berbasis website supaya dapat mempermudah guru-guru di SMA Muhammadiyah 4 untuk pengelolaan data nilai raport dan nilai ujian. Hasilnya, aplikasi sistem informasi akademik SisIAM4 dapat berjalan dengan baik. Sistem ini masih dapat dikembangkan lebih baik lagi. Kekurangan dari aplikasi ini adalah tampilannya yang kurang menarik dan fitur yang ada kurang memadai dalam hal absensi [13].

Setelah itu pada penelitian yang dilakukan oleh Dini Triasanti dengan judul Pembuatan Sistem Informasi Akademik Sekolah Menengah Atas Berbasis Website terdapat beberapa hal yang dapat dipelajari. Penelitian ini memuat metode pengembangan *waterfall* dimana Sistem Informasi Akademik telah berhasil dibuat dengan menggunakan aplikasi. Pada pengembangan pembuatan Sistem Informasi Akademik yang ada didalam penelitian ini yaitu pengolahan data serta pengarsipan data contohnya yaitu berita kegiatan non sekolah, materi yang ada pada pelajaran, data kelas, data guru, data siswa, nilai siswa, data mata pelajaran, dan jadwal mata pelajaran dengan metode pengujian pada penelitian ini yaitu *blackbox testing* berhasil diujicoba [14].

Pada penelitian selanjutnya yang dilakukan oleh Fitria Risyda, Yamin Nuryamin dengan judul Rancang Bangun Sistem Informasi Akademik Pada Sekolah SMK PGRI 28 Dengan Metode *Waterfall*. Terdapat beberapa masalah yang ada pada penelitian ini yaitu masih belum terintegrasi dengan baik oleh sebab itu masih banyak memiliki kekurangan untuk proses penginputan data akademik. Selain itu pada penelitian ini menerangkan bahwa alasan memilih metodologi

*Waterfall*, karena dilakukan dengan terstruktur. Penelitian ini dengan alat *Adobe Dreamweaver CS6* yang membantu untuk membuat sebuah website [15].

Pada penelitian berikutnya yang telah dilakukan oleh Dicky Wahyudi, Angga Putra, dan Irmayanti yang berjudul Penerapan *Framework Codeigniter* Pada Sistem Absensi *QR Code* Diskominfo Kabupaten Labuhan Batu Selatan. Metode yang digunakan melalui *waterfall*, dengan penggeraan sistematis dan berurutan. Penelitian ini berawal dari kurangnya keefektifan penerapan absensi manual yang dilakukan dinas komunikasi dan informatika (diskominfo). Terdapat beberapa permasalahan yang terjadi pada absensi diantaranya adanya pemborosan kertas absensi, rawan kesalahan input data, rentan akan absensi yang dimanipulasi, kurang efisien pada rekapitulasi data absensi. Penelitian tersebut didapatkan hasil bahwa *framework codeigniter* dapat diterapkan pada sistem absensi berbasis *QR code*, sehingga memberikan akses lebih mudah dalam absensi pada diskominfo kabupaten Labuhan Batu dengan proses penggeraan kode program yang lebih efektif [16].

Penelitian yang telah dilakukan oleh Ahmad Saripudin dan Maulana Ardiansyah dengan judul Sistem Informasi Akademik Sekolah Berbasis Web Menggunakan Model *Prototype*. Metode yang digunakan ialah metode *Prototype* mendapatkan hasil penelitian yaitu berdasarkan perkembangan innformasi yang begitu cepat dan pesat seiring berjalannya waktu, diperlukan adanya inovatif baru untuk menunjang sistem informasi yang ada pada SMK Bina Mandiri. SMK Bina Mandiri masih menggunakan cara penginputan nilai melalui *microsoft excel*, sehingga sangat dirassa kurang efektif dalam hal efisiensi waktu. Hasil yang diperoleh dalam penelitian ini adalah sistem informasi akademik SMK Bina Mandiri menjadi sebuah solusi dalam pengolahan akademik yaitu penginputan nilai. Kelemahan pada penelitian adalah penggunaannya masih dalam skala kecil yaitu dengan menu nilai [17].

Penelitian yang telah dilakukan Tri Haryati dengan judul Metode *Waterfall* Pada Sistem Informasi Akademik SMPIT Boarding School Thariq Bin Ziyad Cikarang. Metode pada penelitian ini ialah *waterfall*, dengan sebuah permasalahan dalam penelitian ini berasal dari sistem pengolahan akademiknya masih manual

dalam pengolahan datanya. Sehingga dalam hal ini ada inisiatif untuk menciptakan website akademik SMPIT tersebut. Hasil penelitian menunjukkan bahwa penggunaan sistem informasi akademik dapat memberikan manfaat dalam pengelolaan data siswa SMPIT Thariq bin Ziyad Cikarang. Kelemahan pada penelitian ini adalah pada proses pengujian *blackbox* semua fitur tidak diuji [18].

Berdasarkan hasil penelitian di atas, penulis mengambil inisiatif untuk membangun sebuah Sistem Informasi Akademik khusus untuk SD Negeri Siwarak Wetan. Situs *website* ini tidak hanya berfungsi sebagai sarana untuk menyampaikan informasi, tetapi juga dilengkapi dengan fitur penginputan jadwal pelajaran dan proses presensi akademik. Penelitian ini berbeda dengan penelitian sebelumnya karena sistem informasi yang dibuat tidak hanya sebagai media promosi, tetapi juga sebagai alat bantu bagi pihak sekolah dalam mengelola data sekolah, seperti penginputan jadwal pelajaran dan proses presensi akademik yang sesuai dengan kebutuhan SD Negeri Siwarak Wetan. Hasil analisis penelitian ini dijelaskan secara rinci dalam Tabel 2.1.

Tabel 2. 1 Literatur review dari penelitian

No	Judul	Nama	Tahun	Metode	Hasil
1	Rancang Bangun Sistem Informasi Akademik Menggunakan Metode <i>Waterfall</i> pada SMA Kosgoro Tangerang [10]	Lia Dewi Saputri	2019	Penelitian ini menggunakan metode <i>Waterfall</i> .	Penelitian ini berisi tentang pembuatan web sistem informasi akademik SMA Kosgoro Tangerang dengan menyediakan beberapa fitur yaitu laporan data siswa, laporan data guru, laporan wali kelas, serta laporan nilai siswa. Dengan memakai bahasa pemrograman <i>PHP</i> dan <i>MySQL</i> aplikasi ini dibangun agar membantu proses pengolahan data akademik serta dapat membantu memudahkan pengolahan informasi dengan baik.
2	Penerapan Model <i>Waterfall</i> dalam Pengembangan Sistem Informasi Akademik Berbasis Web sebagai Sistem Pengolahan Nilai Siswa [11]	Kevin Fahrezi, Ahmad Rifqi Mulana, Sandra Melinda, Nurhaliza dan Sri Mulyati	2021	Penelitian ini menggunakan Model <i>Waterfall</i>	Hasil penelitian tersebut menjadi sistem pengolahan nilai anak didik dikembangkan dengan enam tahap dengan berjalan dengan lancar. Selain itu terdapat kekurangan dari segi tampilan serta segi desain.
3	Perancangan Sistem Informasi Akademik Berbasis Web Pada SD Islam Luqmanul Hakim Bekasi [12]	Marijan dan Siti Nurajizah	2019	Penelitian ini dengan <i>model prototype</i>	Pada hasil penelitian tersebut menginformasikan kepada wali murid mencakup aspek nilai sikap serta prestasi yang diperoleh anak serta dapat dibuka kapanpun dan dimanapun berada

No	Judul	Nama	Tahun	Metode	Hasil
4	Penerapan Metode <i>Agile Scrum</i> pada rancangan SisIAM4 [13]	Faza Nadhira, Iwan Wahyudin, Dan Ratih Titi Komala	2022	Penelitian ini dengan metode <i>scrum</i>	Pada penelitian ini merancang web sistem informasi akademik SisIAM4 dengan metode <i>scrum</i> menggunakan <i>framework codeigniter</i> . Sistem informasi ini bisa diakses melalui komputer <i>client</i> ataupun <i>server</i> . Berdasarkan hasil pengujian yang dilakukan melalui <i>blackbox testing</i> , sistem informasi bekerja dengan baik. Serta sistem ini masih dapat dikembangkan lebih baik lagi.
5	Pembuatan Sistem Informasi Akademik Sekolah Menengah Atas Berbasis <i>Website</i> [14]	Dini Triasanti	2021	Penelitian ini dengan pengembangan <i>waterfall</i>	Pada penelitian ini membuat informasi akademik dengan menggunakan aplikasi <i>Visual Studio Code, PHP dan MySQL</i> . Selain itu penelitian ini berhasil di uji coba pada setiap fitur halaman siswa, guru dan admin secara fungsionalitas telah berjalan dengan baik
6	Rancang Bangun Sistem Informasi Akademik Pada Sekolah SMK PGRI 28 Dengan <i>Metode Waterfall</i> [15]	Fitria Risyda, Yamin Nuryamin	2018	Penelitian ini dengan metode <i>Waterfall</i>	Pada penelitian ini memuat tentang pembuatan web Sekolah SMK PGRI 28 Dengan <i>Metode Waterfall</i> . Hasil penelitian ini menunjukkan bahwa fungsi dan juga fitur yang ada pada <i>website</i> dapat berfungsi dengan baik sebagaimana mestinya sehingga bisa digunakan sebagai <i>website</i> SMK PGRI 28 JAKARTA .

No	Judul	Nama	Tahun	Metode	Hasil
7	Penerapan <i>Framework Codeigniter</i> pada sistem absensi <i>QR code</i> Diskominfo kabupaten Labuhan Batu Selatan [16]	Dicky Wahyudi, Angga Putra, dan Irmayanti	2021	Penelitian ini dengan metode <i>Waterfall</i>	Pada penelitian ini memuat tentang pembuatan sistem absensi berbasis <i>QR Code</i> dengan bahasa pemrograman <i>PHP</i> . Didapatkan hasil bahwa <i>framework codeigniter</i> dapat diterapkan pada sistem absensi berbasis <i>QR code</i> , sehingga memberikan akses lebih mudah dalam absensi pada diskominfo kabupaten labuhan batu dengan proses penggerjaan kode program yang lebih efektif. Pada penelitian ini juga dapat kekurangan yaitu belum adanya fitur <i>geo-location</i> sehingga setiap karyawan tidak bisa dilacak dimana posisinya.
8	Sistem Informasi Akademik Sekolah Berbasis Web Menggunakan Model Prototype [17]	Ahmad Saripudin dan Maulana Ardiansyah	2020	Penelitian ini dengan metode <i>Prototype</i>	Pada penelitian ini memuat tentang pembuatan sistem informasi akademik dengan bahasa pemrograman <i>PHP</i> dan <i>framework codeigniter</i> . Dengan hasil bahwa adanya sistem informasi akademik dapat membantu pengelolaan data secara cepat serta yang memiliki akses guru, siswa, serta admin.
9.	Metode <i>waterfall</i> pada sistem informasi akademik smpit boarding school Thariq bin Ziyad Cikarang [18]	Tri Haryati	2019	Penelitian ini dengan metode <i>Waterfall</i>	Pada penelitian ini memuat hasil bahwa adanya sistem informasi akademik dapat membantu pengelolaan data siswa SMPIT Thariq bin Ziyad Cikarang, sistem ini dibuat dengan <i>PHP</i> .

## 2.2 Landasan Teori

Landasan teori adalah hasil pengumpulan definisi dan konsep dari berbagai sumber yang menjadi dasar yang kuat dalam suatu penelitian. Oleh karena itu, dalam penelitian ini dijelaskan beberapa landasan teori yang mendukung pengembangan sistem informasi akademik di SD Negeri Siwarak Wetan dengan menggunakan metode *waterfall*.

### 2.2.1 Perancangan

Perancangan adalah menentukan proses dan juga data yang diperlukan oleh sistem baru sedangkan “Rancang bangun adalah proses membangun suatu sistem untuk membuat sistem baru, atau mengganti maupun memperbaiki sistem yang sudah ada secara keseluruhan ataupun hanya Sebagian” [19]. Adapun juga menurut pendapat lain yaitu "Definis rancang adalah sebuah proses menganalisis kebutuhan dan jelaskan secara merinci komponen-komponennya untuk diimplementasikan. Dan juga menjelaskan arti dari bangun adalah sedang menciptakan atau membuat sebuah sistem baru atau memperbaiki suatu sistem yang memang sudah ada agar bermanfaat bagi penggunanya" [20].

### 2.2.2 Sistem Informasi Akademik

Sistem Informasi Akademik adalah sistem yang dilakukan melalui jaringan untuk memudahkan penanganan urusan yang berkaitan dengan kegiatan administrasi akademik pada lembaga pendidikan. Beberapa kegiatan diantaranya pendaftaran siswa baru, penjadwalan mata pelajaran, pengelolaan data guru, data siswa, pengelolaan nilai, dan masih banyak lagi. Perkembangan teknologi informasi saat ini sedang gencar diterapkan di bidang akademik yang mengarah pada aplikasi berbasis web sebagai media dari semua data dan informasi yang terdapat dalam sistem [21].

Sistem informasi akademik adalah sistem pada lembaga pendidikan yang mengolah data akademik dari sekolah dasar sampai perguruan tinggi. Menurut Imelda dalam penelitiannya menjelaskan bahwa, sistem informasi akademik merupakan sistem yang menyediakan sebuah layanan informasi terkait akademik [22].

Dari informasi tersebut secara garis besar, data yang diolah dalam sistem informasi akademik meliputi data tenaga kependidikan, data siswa, data mata pelajaran, rencana pengajaran, dan data umum lainnya berdasarkan kebutuhan masing-masing lembaga pendidikan. Singkatnya, sistem informasi akademik dapat digambarkan sebagai aplikasi yang dapat membantu memfasilitasi pengelolaan data serta informasi yang berkaitan dengan lembaga pendidikan.

### **2.2.3 *CodeIgniter***

*CodeIgniter* adalah sebuah *web framework* yang ditulis dalam bahasa pemrograman *PHP* oleh EllisLab. *CodeIgniter* digunakan untuk mempermudah pekerjaan *web developer* dalam pembuatan sebuah website maupun *backend* dari sebuah perangkat lunak. Framework ini dirancang untuk meningkatkan kualitas perangkat lunak web karena sangat ringan digunakan dan ekspresif dan kumpulan fitur inti yang menghemat waktu dan memberikan pengalaman pengembangan *web* [23].

### **2.2.4 *Pengujian Perangkat Lunak***

Pengujian perangkat lunak merupakan bagian penting dalam proses pengembangan perangkat lunak. Pengujian *BlackBox* digunakan pada penelitian ini untuk memastikan kualitas perangkat lunak yang dibuat.

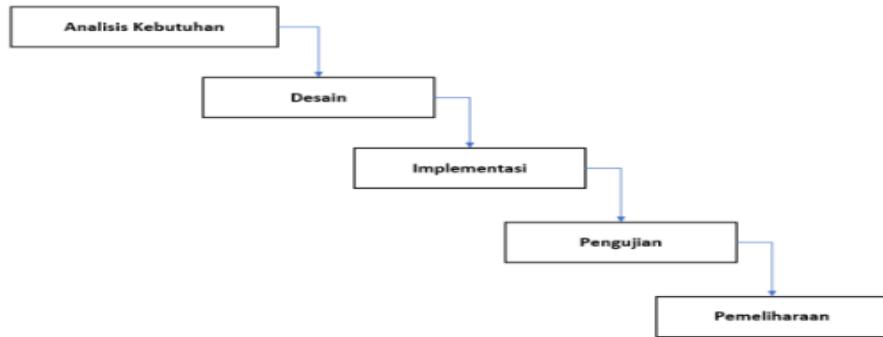
#### **2.2.4.1 *BlackBox Testing***

*BlackBox Testing* adalah metode pengujian yang fokus pada uji coba fungsionalitas aplikasi tanpa memperhatikan desain dan implementasi kode program. Ini dilakukan untuk menemukan masalah seperti kesalahan dalam fungsi aplikasi dan item aplikasi yang hilang [24]. Jadi dengan *BlackBox testing*, kita memungkinkan melakukan uji terhadap kinerja dan fungsionalitas aplikasi yang sudah dibuat tanpa perlu mempelajari bagaimana aplikasi itu dibuat dan bagaimana kode programnya bekerja [24].

### **2.2.5 *Metode Waterfall***

Menurut Hadi pada tahun 2019, didalam *System Development Life Cycle* terdapat beberapa model diantaranya yaitu *waterfall* [25]. Kelebihan dari metode ini adalah pendekatannya yang sistematis dan berurutan, sehingga memudahkan pengembang dalam proses pembuatan sistem dan menjaga kualitas perangkat lunak

yang dihasilkan. Setiap tahap dalam metode waterfall harus diselesaikan secara berurutan sebelum melanjutkan ke tahap selanjutnya [16]. Dapat dilihat tahapan pengembangan metode *waterfall* pada gambar 2.1



Gambar 2. 1 Tahapan Pengembangan Metode *Waterfall* [16]

#### **2.2.5.1 Analisis Kebutuhan**

Pada tahap ini, komunikasi yang diperlukan oleh pengembang sistem adalah untuk memahami kebutuhan pengguna dan batasan perangkat lunak yang akan dikembangkan. Informasi ini umumnya diperoleh melalui wawancara, survei, diskusi, serta pengumpulan data tambahan dari berbagai sumber seperti jurnal, artikel, dan internet [16].

#### **2.2.5.2 Desain**

Pada tahap desain, dilakukan pengalokasian kebutuhan sistem, termasuk perangkat keras dan perangkat lunak, dengan membentuk arsitektur sistem secara keseluruhan. Dalam perancangan perangkat lunak, identifikasi sistem dasar perangkat lunak dan hubungannya juga terlibat [25].

#### **2.2.5.3 Implementasi**

Pada tahap ini, dilakukan proses transformasi desain perancangan menjadi format yang dapat dimengerti oleh mesin dengan menggunakan kode-kode bahasa pemrograman. Kode program yang dihasilkan masih berupa modul-modul kecil yang akan disatukan pada tahap berikutnya [16].

#### **2.2.5.4 Pengujian**

Pada tahap ini, modul-modul yang telah dibuat digabungkan dan diuji untuk mengevaluasi kesesuaian antara perangkat lunak yang telah dibuat dengan

desainnya, serta untuk mengidentifikasi apakah terdapat kesalahan atau tidak dalam fungsi perangkat lunak tersebut [16].

#### **2.2.5.5 Pemeliharaan**

Ini merupakan tahap akhir dalam model waterfall. Setelah perangkat lunak selesai dibangun, tahap selanjutnya adalah menjalankan *software* tersebut dan melakukan pemeliharaan. Pemeliharaan meliputi perbaikan kesalahan yang mungkin tidak ditemukan pada tahap sebelumnya, perbaikan implementasi unit sistem, serta peningkatan fitur sistem sesuai dengan kebutuhan baru [16].

#### **2.2.6 Unified Modeling Language**

*UML*, yang merupakan singkatan dari *Unified Modeling Language*, adalah suatu teknik pemodelan visual yang diterapkan dalam desain dan pembuatan perangkat lunak berbasis objek. *UML* berfungsi sebagai standar atau panduan yang menggambarkan proses bisnis pembuatan kelas menggunakan bahasa yang ditentukan. Saat mengembangkan suatu sistem, terdapat beberapa jenis diagram *UML* yang umumnya digunakan, termasuklah yang telah disebutkan dalam [26].

##### **2.2.6.1 Use Case Diagram**

*Diagram Use Case* adalah representasi visual dari aksi dan interaksi yang diinginkan antara aktor dan sistem dalam suatu sistem tertentu [26]. Dalam sebuah *Use Case*, biasanya terdapat aktor yang dapat berupa individu manusia atau sistem yang melakukan aktivitas dalam sistem tersebut [26]. Terdapat beberapa komponen yang terdapat dalam sebuah *Use Case*, termasuk aktor, *use case*, asosiasi, generalisasi, *include*, dan *extend* [27]. Informasi lebih rinci dapat ditemukan dalam Tabel 2.3 yang disebutkan sebelumnya.

Tabel 2. 2 Komponen-komponen *Use Case Diagram* [27]

Simbol	Keterangan
 Actor	Aktor komponen <i>use case</i> yang mewakili entitas yang berinteraksi dengan sistem. Aktor dapat berupa orang, sistem lain, atau alat yang terlibat dalam skenario penggunaan <i>use case</i> .

	<i>Use Case</i> merupakan representasi grafis dari fungsionalitas sistem yang menunjukkan interaksi antara aktor dan sistem
	<i>Association</i> adalah salah satu relasi dalam <i>diagram use case</i> yang menunjukkan hubungan antara aktor dan <i>use case</i> . <i>Association</i> merepresentasikan keterlibatan atau interaksi antara aktor dan <i>use case</i> yang terkait
	Generalisasi menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	<i>Include</i> merupakan relasi antara <i>use case</i> yang menunjukkan bahwa fungsionalitas dari <i>use case</i> yang satu sepenuhnya tergantung pada fungsionalitas <i>use case</i> lain.
	<i>Extend</i> merupakan relasi antara <i>use case</i> yang menunjukkan bahwa suatu <i>use case</i> (yang disebut <i>use case tambahan</i> ) dapat dieksekusi hanya jika kondisi tertentu terpenuhi dalam <i>use case</i> lain (yang disebut <i>use case utama</i> )

### 2.2.6.2 Sequence Diagram

*Sequence Diagram* adalah representasi visual dari hubungan dan pesan yang diterima dan dikirim antara objek dalam sistem dan lingkungannya dalam urutan waktu [26]. Terdapat beberapa komponen dalam *sequence diagram*, diantaranya adalah *actor*, *activation box*, *object*, *lifelines*, *option loop*, *synchronous*, *reply message*, *delete message*, *boundary*, *control*, dan *entity* [28]. Untuk penjelasan lebih lengkap bisa dilihat pada Tabel 2.3 berikut :

Tabel 2. 3 Komponen-komponen *Sequence Diagram* [28]

Simbol	Keterangan
	Komponen <i>stick figure</i> dalam <i>sequence diagram</i> dikenal sebagai aktor, yang mewakili pengguna yang terlibat dalam interaksi dengan sistem, baik secara internal maupun eksternal
	<i>Activation box</i> adalah suatu komponen persegi panjang pada <i>sequence diagram</i> yang mewakili durasi waktu yang dibutuhkan oleh suatu objek untuk menyelesaikan tugas tertentu. Semakin lama waktu yang dibutuhkan, maka <i>activation box</i> akan semakin panjang
	<i>Object</i> merupakan sebuah komponen berbentuk kotak dalam <i>diagram</i> yang digunakan untuk mendemonstrasikan bagaimana sebuah objek akan berinteraksi dalam suatu sistem dan bagaimana perlakunya dalam konteks yang diberikan

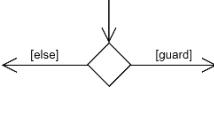
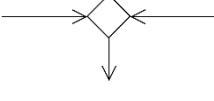
	<i>Lifelines</i> adalah komponen <i>diagram</i> yang berbentuk garis putus-putus, digunakan untuk menunjukkan urutan kejadian yang terjadi pada objek selama pembuatan <i>diagram</i>
	<i>Option loop</i> adalah komponen <i>diagram</i> yang sangat berguna untuk menggambarkan bagaimana sistem berperilaku dalam situasi tertentu, terutama jika ada percabangan atau alternatif jalur yang harus dipertimbangkan
	<i>Synchronous message</i> adalah salah satu komponen <i>diagram</i> yang dilambangkan dengan simbol panah tebal ke kanan, dan digunakan untuk menunjukkan bahwa pengirim harus menunggu <i>respon</i> pesan sebelum melanjutkan
	<i>Reply message</i> adalah salah satu komponen <i>diagram</i> yang dilambangkan dengan simbol panah putus-putus ke kiri, dan berfungsi sebagai balasan atas suatu panggilan tertentu
	<i>Delete message</i> merupakan komponen <i>diagram</i> yang dilambangkan dengan tanda cakra di tengahnya terdapat garis putus-putus, dan berfungsi untuk menghapus objek tertentu
	Komponen <i>boundary</i> dalam <i>diagram</i> biasanya terletak pada batas luar sistem, dan dapat berupa antarmuka pengguna atau alat lain yang digunakan untuk berinteraksi antara pengguna dan <i>system</i>
	Komponen <i>control</i> dalam sebuah sistem bertanggung jawab untuk mengatur alur informasi dalam skenario bisnis, dan dapat mempengaruhi perilaku teknis dari sistem
	<i>Entity</i> adalah komponen sistem yang berperan menyimpan informasi atau data, dan biasanya direpresentasikan sebagai objek model atau beans

### 2.2.6.3 Activity Diagram

*Activity Diagram* adalah sebuah representasi visual yang menggambarkan urutan dari aktivitas yang dilakukan oleh suatu sistem, menunjukkan aliran aktivitas secara berurutan [26]. Pada sebuah *activity diagram*, terdapat beberapa komponen yang ada didalamnya, seperti *initial state/start*, *final state/end*, *activity*,

*transisi/association, decision, dan merge* [29]. Untuk penjelasan lebih lengkap mengenai komponen-komponen tersebut, bisa dilihat pada Tabel 2.4 berikut :

Tabel 2. 4 Komponen-komponen *Activity Diagram* [29]

Simbol	Keterangan
	<i>Initial state</i> merupakan titik awal atau <i>starting point</i> dari alur kerja dalam sebuah <i>activity diagram</i> , dan hanya terdapat satu <i>initial state</i> pada setiap <i>diagramnya</i>
	<i>Final state</i> merupakan komponen yang menunjukkan akhir dari suatu alur kerja. Pada satu <i>activity diagram</i> , bisa terdapat lebih dari satu <i>final state</i> tergantung pada desain alur kerja
	<i>Activity diagram</i> adalah sebuah <i>diagram</i> yang digunakan untuk memodelkan urutan aktivitas atau tugas dalam suatu alur kerja atau proses bisnis. <i>Diagram</i> ini membantu dalam menggambarkan hubungan antara aktivitas-aktivitas tersebut, sehingga memudahkan dalam analisis proses bisnis atau alur kerja dalam sebuah sistem
	Transisi/ <i>Association</i> pada <i>Activity Diagram</i> menghubungkan antara dua aktivitas atau lebih dan menunjukkan urutan dari aktivitas tersebut. Komponen ini berfungsi untuk menggabungkan aktivitas selanjutnya setelah aktivitas sebelumnya dan menunjukkan aliran kerja atau alur yang harus diikuti oleh sistem
	Keputusan atau "Decision" dalam <i>activity diagram</i> digunakan untuk merepresentasikan sebuah pilihan atau kondisi dimana alur kerja dapat bercabang menjadi dua atau lebih jalur
	<i>Merge</i> berfungsi untuk menggabungkan kembali aliran kerja yang sebelumnya telah dipecah oleh suatu <i>decision point</i> , sehingga mengembalikan aliran kerja menjadi satu jalur tunggal.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Subjek dan Objek Penelitian**

Berdasarkan latar belakang pada bab I, penelitian ini menggunakan metode pengembangan *waterfall*. Subjek penelitian dalam penelitian ini yaitu guru serta operator sekolah SD Negeri Siwarak Wetan Kecamatan Tambak guna mempermudah kinerja guru dalam mengelola data peserta didik. Adapun objek dalam penelitian ini, yaitu Sistem Informasi Akademik SD Negeri Siwarak Wetan di Kecamatan Tambak, Kabupaten Banyumas.

#### **3.2 Alat Penelitian**

Pada penelitian ini menggunakan beberapa syarat minimum dari perangkat keras maupun perangkat lunak laptop.

##### **3.2.1 Perangkat Keras**

Penelitian ini menggunakan spesifikasi kebutuhan perangkat keras yang digunakan untuk merancang Sistem Informasi Akademik di SD Negeri Siwarak Wetan yaitu Processor Intel(R) Celeron(R) CPU N2840 @ 2.16GHz 2.16 GHz dan Ram 2GB.

##### **3.2.2 Perangkat Lunak**

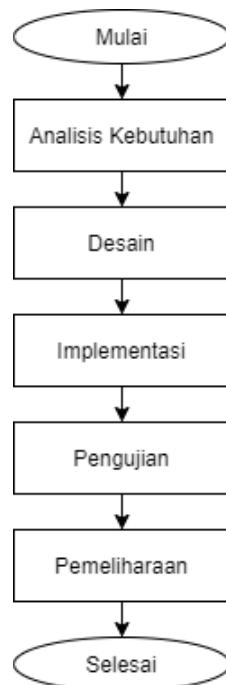
Perangkat lunak yang akan digunakan pada pembuatan serta pengembangan website ini dapat dilihat pada Tabel 3.1 Perangkat Lunak

Tabel 3. 1 Perangkat Lunak

No	Kebutuhan	Keterangan
1	Sistem operasi	Windows 10
2	Aplikasi	<i>Google Chrome</i>
		<i>Visual Studio Code</i>
		<i>Mysql</i>
		<i>Figma</i>
		<i>Draw io</i>

### 3.3 Diagram Alir Penelitian

Dalam pelaksanaan penelitian pengembangan dan realisasi sistem informasi akademik ini mempunyai beberapa alur dan tahapan, yaitu analisis kebutuhan, desain, implementasi, pengujian perangkat lunak serta pemeliharaan. Alur atau tahapan dalam penelitian ini diterapkan dalam diagram, yang dapat dilihat pada Gambar 3.1 diagram alir penelitian.



Gambar 3. 1 Diagram Alir Penelitian

#### 3.3.1 Analisis Kebutuhan

Di tahap analisis kebutuhan akan dilakukan dengan identifikasi masalah, studi literatur, pengumpulan data, serta analisis.

##### 3.3.1.1 Identifikasi Masalah

Tahap awal pada penelitian ini akan dilakukan identifikasi masalah dengan observasi ke sekolah terutama guru SD Negeri Siwarak Wetan. Umumnya harus memasukan data secara manual sehingga memakan waktu yang lama pada sistem SD Negeri Siwarak Wetan. Masalah lain yang kerap timbul yaitu penjadwalan dari seluruh kelas masih belum terstruktur. Selain itu merujuk pada latar belakang,

sistem informasi yang berada di SD Negeri Siwarak Wetan masih belum termodernisasi, sehingga sering terjadi penumpukan data siswa dan mengakibatkan sulitnya pencarian data siswa apabila sewaktu-waktu data itu dibutuhkan.

### **3.3.1.2 Studi Literatur**

Untuk melakukan proses pengumpulan data yang dibutuhkan, studi literatur merupakan salah satu metode yang dapat dimanfaatkan untuk mengumpulkan informasi yang diperlukan. Studi literatur akan dilakukan dengan membaca referensi-referensi yang dapat membantu proses penyelesaian masalah pada penelitian ini baik dengan membaca buku, hasil penelitian yang adasebelumnya dan informasi dari internet yang memiliki informasi terkait dalam pembuatan sistem informasi akademik ini.

### **3.3.1.3 Pengumpulan Data**

Untuk mengumpulkan data yang akan diperlukan tidak hanya diperoleh dari referensi yang bersumber pada internet, buku maupun karya ilmiah terdahulu, tetapi juga melalui wawancara dengan kepala sekolah. Selain cara-cara tersebut juga akan dilakukan penyebaran kuisioner secara online terhadap guru, hal tersebut bertujuan agar pembuatan sistem informasi sekolah yang dihasilkan tidak hanya sesuai dengan sudut pandang peneliti saja namun juga dari tenaga pendidik. Daftar pertannyaan dapat dilihat pada lampiran.

### **3.3.1.4 Analisis**

Setelah mengumpulkan data akan dilakukan analisis. Pada tahapan analisis akan digunakan diagram *Unified Modelling Language (UML)* yaitu pembuatan *use case diagram, activity diagram, dan sequence diagram*.

### **3.3.2 Desain**

Pada tahapan ini akan perancangan design yang akan dilakukan adalah perancangan basis data dengan menggunakan *ERD (Entity Relationship Diagram)*.

### **3.3.3 Implementasi**

Tahap Implementasi merupakan tahap di mana perencanaan dan rancangan yang telah dibuat akan diwujudkan menjadi sebuah aplikasi melalui proses

pengkodean (*Coding*). Aplikasi sistem informasi akademik ini menggunakan bahasa pemrograman *PHP* dan *framework CodeIgniter*, dengan dukungan *tools Visual Studio Code*. Setelah itu, aplikasi akan dihosting agar dapat diakses dengan mudah melalui internet.

### 3.3.4 Pengujian

Untuk melihat keberhasilan *website* yang dirancang oleh peneliti, Penelitian ini akan dilakukan tahap pengujian dengan menggunakan metode *blackbox testing* untuk memastikan bahwa fitur dan fungsi yang telah dirancang berjalan sesuai dengan yang diharapkan. Teknik *purposive sampling* merupakan pengambilan sampel dengan pertimbangan atau syarat khusus. Pengujian perangkat lunak ini akan dilakukan dengan menggunakan *purposive sampling* dikarenakan penelitian ini merujuk pada pertimbangan yang akan diajukan harus memenuhi kriteria yang ditetapkan. Berikut kriteria sebagai responden :

1. Bekerja sebagai guru di SD Negeri Siwarak Wetan Kabupaten Banyumas
2. Berstatus aktif
3. Memiliki wewenang sebagai wali kelas, guru khusus, atau kepala sekolah

Dengan demikian maka terdapat 10 orang yang sesuai dengan kriteria responden.

### 3.3.5 Pemeliharaan

Tahapan terakhir yaitu dilakukan pemeliharaan atau *maintenance*. Tahap pemeliharaan ini dilakukan agar tidak ada teradinya error atau bug dan gangguan pada sistem, sehingga tahap ini dapat dilakukan secara berkala. Pada masa ini, sistem informasi akademik yang sudah dipasarkan akan di pantau untuk melihat apakah website yang sudah berjalan dengan baik. Dan apabila sudah baik maka pada bagian ini dapat dikatakan sudah berhasil dan juga tetap di jaga agar nantinya tetap bias berjalan sebagaimana mestinya, kalaupun ada dan terdapat celah celah makan dapat dilakukan penyempurnaan.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Hasil**

Penelitian ini menghasilkan sebuah sistem informasi akademik yang dapat digunakan untuk menangani proses akademik yang terdigitalisasi di SD Siwarak Wetan Kecamatan Tambak Kabupaten Banyumas berdasarkan kebutuhan SD Negeri Siwarak Wetan serta dapat berfungsi dengan baik secara fungsionalitasnya sehingga mempermudah kinerja guru dan operator sekolah dalam proses penginputan data. Melalui sistem informasi yang dilakukan melalui beberapa tahap awal yakni analisis kebutuhan, desain, implementasi, tes perangkat lunak, serta yang terakhir yaitu pemeliharaan atau pemeliharaan. Pada analisis kebutuhan mendapatkan sebuah permasalahan yang kemudian menganalisis dengan menggunakan *Unified Modelling Language (UML)*. Selanjutnya mendesain sebuah basis data yang digunakan untuk implementasi kedalam sebuah program. Dengan menggunakan figma untuk membuat *low fidelity* kemudian diaksikan ke dalam kodingan sehingga menjadi *website* yang dapat dijalankan. Langkah selanjutnya yaitu tes perangkat lunak dengan menggunakan *blackbox testing*. Dengan *purposive sampling* memiliki sebuah kriteria responden sehingga menghasilkan data nilai kelayakan sangat baik dengan hasil persentase kelayakan 100% sehingga dapat disimpulkan bahwa sistem sudah berfungsional semestinya dan tidak ada kegagalan, selanjutnya pemeliharaan perlu dilakukan karena untuk meminimalisir terjadinya *bug* atau *error* sehingga mengganggu kenyamanan pengguna. Pemeliharaan dilakukan selama 4 bulan yaitu Januari, Februari, Maret, April.

#### **4.2 Pembahasan**

Dalam pembahasan ini, membahas secara detail mengenai hasil yang telah dijelaskan sebelumnya, seperti analisis kebutuhan, desain, implementasi, tes perangkat lunak, serta yang terakhir yaitu pemeliharaan atau *maintenance*.

#### **4.2.1 Analisis Kebutuhan**

Pada tahap analisis kebutuhan merupakan tahap untuk melakukan rencana dan mempersiapkan data yang akan digunakan pada tahap-tahap selanjutnya. Pada tahap ini terbagi menjadi beberapa tahapan lagi sebagai berikut.

##### **4.2.1.1 Identifikasi Masalah**

Dari identifikasi yang telah dilakukan sebelumnya mendapatkan hasil permasalahan permasalahan yang ada terkait pengembangan sistem informasi akademik yang menjadi topik penelitian, seperti harus memasukan data secara manual sehingga memakan waktu yang lama pada SD Negeri Siwarak Wetan. Masalah lain yang kerap timbul yaitu penjadwalan dari seluruh kelas masih belum terstruktur. Selain itu sistem informasi yang berada di SD Negeri Siwarak Wetan masih belum termodernisasi, sehingga sering terjadi penumpukan data siswa dan mengakibatkan sulitnya pencarian data siswa apabila sewaktu-waktu data itu dibutuhkan.

##### **4.2.1.2 Pengumpulan Data**

Dalam proses pengumpulan informasi, dilakukan dengan cara melakukan survei ke SD Negeri Siwarak Wetan. Penulis berhasil melakukan pengumpulan data sebanyak 10 responden tenaga pendidikan di SD Negeri Siwarak Wetan terdiri dari guru kelas satu sampai enam, dua guru mapel, satu operator sekolah dan satu kepala sekolah. Hasil dari survei dengan responden 90% tenaga pendidikan di SD Negeri Siwarak Wetan Kabupaten Banyumas perlu dilakukan digitalisasi dengan menerapkan sistem informasi akademik berbasis website. Dengan 80% memiliki fitur yang akan diterapkan yaitu terdapat presensi dan siswa, serta 70% memiliki fitur jadwal dan mata pelajaran. Sementara itu 30% memilih terdapat fitur input nilai dan 10% memilih untuk belum membutuhkan sistem informasi akademik. Data hasil survei *google form* secara lengkap dapat dilihat pada **lampiran 2**.

#### 4.2.1.3 Analisis

Pada tahap analisis, membahas mengenai rancangan sistem website dan analisis yang mencakup perancangan sebuah sistem dengan menggunakan *Unified Modelling Language(UML)*.

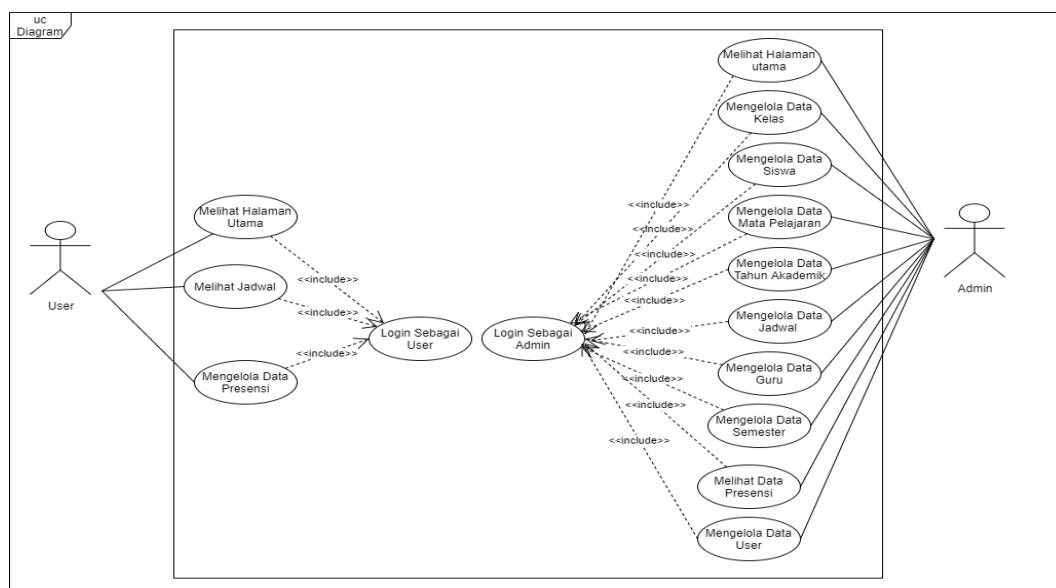
##### 1. *Unified Modelling Language*

*Unified Modelling Language (UML)* disini meliputi pembuatan *use case diagram*, *activity diagram*, dan *sequence diagram*.

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

Dari kebutuhan yang dibutuhkan oleh sistem informasi akademik SD Negeri Siwarak Wetan, beberapa Use Case diagram dirancang seperti yang dapat dilihat pada gambar 4.1 di bawah.

Dalam perancangan website SD Negeri Siwarak, ada dua tokoh utama, yaitu admin dalam website ini merupakan operator sekolah dan kepala sekolah yang bertindak sebagai pengelola data pada website sistem informasi akademik dan user merupakan guru kelas atau wali kelas yang juga sebagai pengguna dari sistem informasi akademik SIAwarak



Gambar 4. 1 *Use Case* Sistem Informasi Akademik SIAwarak

Gambar 4.1 di atas menjelaskan rancangan *Use Case* dari seorang user dan admin yang mana di dalam website sistem informasi akademik, user dapat melakukan *login* akun, melihat halaman utama/*dashboard*, melihat jadwal pelajaran yang akan diajar, dan mengelola presensi akademik siswa. Lalu admin dapat melakukan proses *login*, melihat halaman utama/*dashboard*, mengelola data kelas, mengelola data siswa, mengelola data mata pelajaran, mengelola data tahun akademik, mengelola data jadwal pelajaran, mengelola data guru, mengelola data semester, mengelola data user, dan melihat data presensi yang telah diinputkan oleh guru sebagai user.

Dari *Use Case* yang telah dirancang, maka dapat dijabarkan ke dalam *Use Case specification* seperti yang tertera pada Tabel 4.1 sampai dengan Tabel 4.15 berikut ini.

Tabel 4. 1 *Use Case Specification Login Pengguna*

Nomor	1
Nama	: <i>Login</i>
Actor	: User
Extension	: -
Include	: -
Pre-Condition	: <i>Login User</i>
Skenario utama	: <ol style="list-style-type: none"> <li>1. Pengguna masuk kehalaman login website</li> <li>2. Pengguna mengisi <i>username</i> dan <i>password</i></li> </ol>
Post Condition	: Wesbste mampu menunjukkan halaman utama atau halamang
Skenario Abnormal	: : Jika informasi yang dimasukkan tidak sah, website tidak mampu menunjukkan halaman utama

Pada Tabel 4.1 merupakan sebuah *Use Case* dari seorang user yang dapat melakukan *login* ke dalam website, *Use Case* dimulai ketika website menampilkan menu *login* kemudian dapat diisi oleh pengguna dengan memasukkan *username* dan *password*. Jika data masukan yang dimasukkan benar, maka website akan meneruskan ke halaman *dashboard*, tetapi jika data *inputan* tidak valid, website tidak akan bisa melanjutkan ke proses berikutnya, kemudian memasukan *username* dan *password* kembali.

Tabel 4. 2 *Use Case Specification* Melihat Dashboard Pengguna

Nomor	2
Nama	: Melihat <i>Dashboard</i>
<i>Actor</i>	: User
Extension	: -
<i>Include</i>	: <i>Login User</i>
<i>Pre-Condition</i>	: berhasil dijalankan dalam keadaan pengguna sudah memiliki akun dan berhasil <i>login</i>
Skenario utama	: <ol style="list-style-type: none"> <li>1. User berhasil mengakses website</li> <li>2. User telah berhasil masuk dengan menggunakan <i>login</i></li> <li>3. Halaman utama ditampilkan oleh website</li> </ol>
<i>Post Condition</i>	: Website berhasil menampilkan halaman utama
Skenario Abnormal	: User belum menyelesaikan proses login, sehingga aplikasi tidak bisa menunjukkan halaman dashboard

Tabel 4.2 menjelaskan gambaran dari sebuah *Use Case* pengguna, yaitu melihat halaman utama, *Use Case* ini dimulai dari pengguna yang telah berhasil melakukan proses *login* kemudian website akan menampilkan halaman utama dengan benar. Website tidak akan menampilkan halaman utama apabila pengguna belum melakukan login pada website.

Tabel 4. 3 *Use Case Specification* Melihat Jadwal Pengguna

Nomor	3
Nama	: Melihat Jadwal
<i>Actor</i>	: User
Extension	: -
<i>Include</i>	: <i>Login User</i>
<i>Pre-Condition</i>	: User memilih menu jadwal yang ada pada sub menu akademik
Skenario utama	: <ol style="list-style-type: none"> <li>1. User memilih menu akademik</li> <li>2. User memilih jadwal pada sub menu akademik</li> <li>3. Website menunjukkan jadwal pelajaran</li> </ol>
<i>Post Condition</i>	: Website menunjukkan jadwal pelajaran
Skenario Abnormal	: -

Tabel 4.3 menjelaskan gambaran dari sebuah *Use Case* pengguna, yaitu melihat jadwal, *Use Case* ini dimulai dari pengguna yang telah berhasil melakukan proses *login* kemudian website akan menampilkan halaman utama dengan benar. Selanjutnya pengguna memilih menu akademik, lalu pengguna

memilih jadwal pada sub menu akademik. Kemudian Website menampilkan halaman jadwal.

Tabel 4. 4 *Use Case Specification* Mengelola Data Presensi Pengguna

Nomor	4
Nama	: Mengelola Data Presensi
Actor	: User
Extension	: -
Include	: <i>Login User</i>
Pre-Condition	: User memilih menu presensi yang ada pada sub menu akademik
Skenario utama	: <ol style="list-style-type: none"> <li>1. User memilih menu akademik</li> <li>2. User memilih presensi pada sub menu akademik</li> <li>3. User mengelola data presensi</li> </ol>
Post Condition	: User mengelola data presensi pada website
Skenario Abnormal	: Memunculkan alert dialog pada website

Tabel 4.4 diatas merupakan gambaran dari sebuah *Use Case* pengguna yaitu melakukan pengelolaan terhadap data presensi yang dimana *Use Case* dimulai ketika pengguna telah berhasil melakukan *login* ke dalam website, kemudian website menampilkan halaman utama, dan pengguna dapat memilih menu akademik lalu memilih presensi pada sub menu akademik. Website akan menampilkan halaman siswa yang akan di presensi, pengguna dapat mengelola data presensi siswa. Pengguna dapat melakukan penambahan data presensi siswa. Selain itu pengguna dapat mengedit, dan menghapus data yang kurang tepat. Pengguna dapat mendownload data presensi siswa. Pada *Use Case* dikatakan tidak berhasil apabila ketika tambah data presensi terdapat data kosong, maka website akan memunculkan alert. Selain itu ketika hapus data presensi maka ada alert pada website.

Tabel 4. 5 *Use Case Specification Login Admin*

Nomor	5
Nama	: <i>Login Admin</i>
Actor	: Admin
Extension	: -
Include	: -
Pre-Condition	: Website berhasil dijalankan

Skenario utama :	
1.	Website sudah berjalan di perangkat admin
2.	Admin melakukan pengisian <i>form</i> untuk melanjutkan proses <i>login</i>
<i>Post Condition</i>	: Admin berhasil <i>login</i> dan akan masuk ke dalam tampilan admin
Skenario Abnormal	: Website tidak dapat menampilkan tampilan admin dan tidak dapat melakukan proses berikutnya jika admin tidak mengisi <i>form</i> dengan sesuai

Tabel 4.5 merupakan gambaran dari Use Case seorang admin yang dapat melakukan login, dimulai dengan website menampilkan halaman login dan mengisi *username* dan *password*. Jika data yang dimasukkan benar, maka administrator akan masuk ke halaman utama admin. Namun, jika website tidak dapat menampilkan halaman utama admin, maka kasus penggunaan ini dikatakan gagal.

Tabel 4. 6 *Use Case Specification* Melihat Dashboard Admin

Nomor	6
Nama	: Melihat Dashboard
<i>Actor</i>	: Admin
Extension	: -
<i>Include</i>	: Login Admin
<i>Pre-Condition</i>	: berhasil dijalankan dalam keadaan sudah memiliki akun dan berhasil <i>login</i>
Skenario utama	: <ul style="list-style-type: none"> <li>1. Admin berhasil mengakses website</li> <li>2. Admin telah berhasil masuk dengan menggunakan <i>login</i></li> <li>3. Halaman utama ditampilkan oleh website</li> </ul>
<i>Post Condition</i>	: Website berhasil menampilkan halaman utama
Skenario Abnormal	: Admin belum menyelesaikan proses <i>login</i> , sehingga aplikasi tidak bisa menunjukkan halaman dashboard

Tabel 4.6 menjelaskan gambaran dari sebuah *Use Case* admin, yaitu melihat dashboard, *Use Case* ini dimulai dari pengguna yang telah berhasil melakukan proses *login* kemudian website akan menampilkan halaman utama dengan benar. Website tidak akan menampilkan halaman utama apabila pengguna belum melakukan login pada website.

Tabel 4. 7 *Use Case Specification* Mengelola Data Kelas Admin

Nomor	7
Nama	: Mengelola Data Kelas

<i>Actor</i>	: Admin
<i>Extension</i>	: -
<i>Include</i>	: <i>Login Admin</i>
<i>Pre-Condition</i>	: Admin memilih menu kelas yang ada pada sub menu akademik
Skenario utama	: <ol style="list-style-type: none"> <li>1. Admin memilih menu akademik</li> <li>2. Admin memilih kelas pada sub menu akademik</li> <li>3. Admin mengelola data kelas</li> </ol>
<i>Post Condition</i>	: Admin mengelola data kelas pada website
Skenario Abnormal	: Memunculkan alert dialog pada website

Tabel 4.7 diatas merupakan gambaran dari sebuah *Use Case* admin yaitu melakukan pengelolaan terhadap data kelas yang dimana *Use Case* dimulai ketika admin telah berhasil melakukan *login* ke dalam website, kemudian website menampilkan halaman utama, dan admin dapat memilih menu akademik lalu memilih kelas pada sub menu akademik. Website akan menampilkan halaman kelas, admin dapat mengelola data kelas. Admin dapat melakukan penambahan data kelas. Selain itu admin dapat mengedit, dan menghapus data yang kurang tepat. Pada *Use Case* dikatakan tidak berhasil apabila ketika tambah data kelas terdapat data kosong, maka website akan memunculkan *alert*. Selain itu ketika hapus data kelas maka aka ada alert pada website.

Tabel 4. 8 *Use Case Specification* Mengelola Data Siswa Admin

Nomor	8
Nama	: Mengelola Data Siswa
<i>Actor</i>	: Admin
<i>Extension</i>	: -
<i>Include</i>	: <i>Login Admin</i>
<i>Pre-Condition</i>	: Admin memilih menu siswa yang ada pada sub menu akademik
Skenario utama	: <ol style="list-style-type: none"> <li>1. Admin memilih menu akademik</li> <li>2. Admin memilih siswa pada sub menu akademik</li> <li>3. Admin mengelola data siswa</li> </ol>
<i>Post Condition</i>	: Admin mengelola data siswa pada website
Skenario Abnormal	: Memunculkan alert dialog pada website

Tabel 4.8 diatas merupakan gambaran dari sebuah *Use Case* admin yaitu melakukan pengelolaan terhadap data siswa yang dimana *Use Case* dimulai ketika admin telah berhasil melakukan *login* ke dalam website, kemudian website menampilkan halaman utama, dan admin dapat memilih menu akademik lalu memilih siswa pada sub menu akademik. Website akan menampilkan halaman siswa, admin dapat mengelola data siswa. Admin dapat melakukan penambahan data siswa dengan melalui website atau upload csv. Admin juga melihat rincian data siswa pada detail siswa. Selain itu admin dapat mengedit, dan menghapus data yang kurang tepat. Admin juga dapat download data siswa. Pada *Use Case* dikatakan tidak berhasil apabila ketika tambah data siswa terdapat data kosong, maka website akan memunculkan alert. Selain itu ketika hapus data siswa maka akan ada alert pada website.

Tabel 4.9 *Use Case Specification* Mengelola Data Mata Pelajaran Admin

Nomor	9
Nama	: Mengelola Data Mata Pelajaran
Actor	: Admin
Extension	: -
Include	: <i>Login</i> Admin
Pre-Condition	: Admin memilih menu mata pelajaran yang ada pada sub menu akademik
Skenario utama	: <ol style="list-style-type: none"> <li>1. Admin memilih menu akademik</li> <li>2. Admin memilih mata pelajaran pada sub menu akademik</li> <li>3. Admin mengelola data mata pelajaran</li> </ol>
Post Condition	: Admin mengelola data mata pelajaran pada website
Skenario Abnormal	: Memunculkan alert dialog pada website

Tabel 4.9 diatas merupakan gambaran dari sebuah *Use Case* admin yaitu melakukan pengelolaan terhadap data mata pelajaran yang dimana *Use Case* dimulai ketika admin telah berhasil melakukan *login* ke dalam website, kemudian website menampilkan halaman utama, dan admin dapat memilih menu akademik lalu memilih mata pelajaran pada sub menu akademik. Website akan menampilkan halaman mata pelajaran, admin dapat mengelola data mata

pelajaran. Admin dapat melakukan penambahan data mata pelajaran. Admin juga melihat rincian data mata pelajaran pada detail mata pelajaran. Selain itu admin dapat mengedit, dan menghapus data yang kurang tepat. Pada *Use Case* dikatakan tidak berhasil apabila ketika tambah data siswa terdapat data kosong, maka website akan memunculkan alert. Selain itu ketika hapus data siswa maka ada alert pada website.

Tabel 4. 10 *Use Case Specification* Mengelola Data Tahun Akademik Admin

Nomor	10
Nama	: Mengelola Data Tahun Akademik
Actor	: Admin
Extension	: -
Include	: <i>Login Admin</i>
Pre-Condition	: Admin memilih menu tahun akademik yang ada pada sub menu akademik
Skenario utama	: <ol style="list-style-type: none"> <li>1. Admin memilih menu akademik</li> <li>2. Admin memilih tahun akademik pada sub menu akademik</li> <li>3. Admin mengelola data tahun akademik</li> </ol>
<i>Post Condition</i>	: Admin mengelola data tahun akademik pada website
Skenario Abnormal	: Memunculkan alert dialog pada website

Tabel 4.10 diatas merupakan gambaran dari sebuah *Use Case* admin yaitu melakukan pengelolaan terhadap data tahun akademik yang dimana *Use Case* dimulai ketika admin telah berhasil melakukan *login* ke dalam website, kemudian website menampilkan halaman utama, dan admin dapat memilih menu akademik lalu memilih tahun akademik pada sub menu akademik. Website akan menampilkan halaman tahun akademik, admin dapat mengelola data tahun akademik. Admin dapat melakukan penambahan data tahun akademik. Selain itu admin dapat mengedit, dan menghapus data yang kurang tepat. Pada *Use Case* dikatakan tidak berhasil apabila ketika tambah data tahun akademik terdapat data kosong, maka website akan memunculkan alert. Selain itu ketika hapus data tahun akademik maka ada alert pada website.

Tabel 4. 11 *Use Case Specification* Mengelola Data Jadwal Pelajaran Admin

Nomor	11
Nama	: Mengelola Data Jadwal Pelajaran
Actor	: Admin
Extension	: -
Include	: <i>Login Admin</i>
Pre-Condition	: Admin memilih menu Jadwal Pelajaran yang ada pada sub menu akademik
Skenario utama	: <ol style="list-style-type: none"> <li>1. Admin memilih menu akademik</li> <li>2. Admin memilih jadwal pelajaran pada sub menu akademik</li> <li>3. Admin mengelola data jadwal pelajaran</li> </ol>
Post Condition	: Admin mengelola data jadwal pelajaran pada website
Skenario Abnormal	: Memunculkan alert dialog pada website

Tabel 4.11 diatas merupakan gambaran dari sebuah *Use Case* admin yaitu melakukan pengelolaan terhadap data tahun akademik yang dimana *Use Case* dimulai ketika admin telah berhasil melakukan *login* ke dalam website, kemudian website menampilkan halaman utama, dan admin dapat memilih menu akademik lalu memilih jadwal pelajaran pada sub menu akademik. Website akan menampilkan halaman jadwal yang berisi data kelas, selanjutnya admin dapat mengelola data jadwal. Admin dapat melakukan penambahan data jadwal. Selain itu admin dapat mengedit, dan menghapus data yang kurang tepat. Pada *Use Case* dikatakan tidak berhasil apabila ketika tambah data jadwal pelajaran terdapat data kosong, maka website akan memunculkan alert. Selain itu ketika hapus data jadwal pelajaran maka ada alert pada website.

Tabel 4. 12 *Use Case Specification* Mengelola Data Guru Admin

Nomor	12
Nama	: Mengelola Data Guru
Actor	: Admin
Extension	: -
Include	: <i>Login Admin</i>
Pre-Condition	: Admin memilih menu guru yang ada pada sub menu akademik
Skenario utama	: <ol style="list-style-type: none"> <li>1. Admin memilih menu akademik</li> <li>2. Admin memilih guru pada sub menu akademik</li> <li>3. Admin mengelola data guru</li> </ol>

<i>Post Condition</i>	: Admin mengelola data guru pada website
Skenario Abnormal	: Memunculkan alert dialog pada website

Tabel 4.12 diatas merupakan gambaran dari sebuah *Use Case* admin yaitu melakukan pengelolaan terhadap data guru yang dimana *Use Case* dimulai ketika admin telah berhasil melakukan *login* ke dalam website, kemudian website menampilkan halaman utama, dan admin dapat memilih menu akademik lalu memilih guru pada sub menu akademik. Website akan menampilkan halaman guru, admin dapat mengelola data guru. Admin dapat melakukan penambahan data guru. Admin juga melihat rincian data guru pada detail guru. Selain itu admin dapat mengedit, dan menghapus data yang kurang tepat. Pada *Use Case* dikatakan tidak berhasil apabila ketika tambah data guru terdapat data kosong, maka website akan memunculkan alert. Selain itu ketika hapus data guru maka akan ada alert pada website.

Tabel 4. 13 *Use Case Specification* Mengelola Data Semester Admin

Nomor	13
Nama	: Mengelola Data Semester
Actor	: Admin
Extension	: -
Include	: <i>Login Admin</i>
Pre-Condition	: Admin memilih menu semester yang ada pada sub menu akademik
Skenario utama	<ul style="list-style-type: none"> <li>1. Admin memilih menu akademik</li> <li>2. Admin memilih semester pada sub menu akademik</li> <li>3. Admin mengelola data semester</li> </ul>
<i>Post Condition</i>	: Admin mengelola data semester pada website
Skenario Abnormal	: Memunculkan alert dialog pada website

Tabel 4.13 diatas merupakan gambaran dari sebuah *Use Case* admin yaitu melakukan pengelolaan terhadap data semester yang dimana *Use Case* dimulai ketika admin telah berhasil melakukan *login* ke dalam website, kemudian website menampilkan halaman utama, dan admin dapat memilih menu akademik lalu memilih semester pada sub menu akademik. Website akan menampilkan halaman

semester, admin dapat mengelola data tahun akademik. Admin dapat melakukan penambahan data tahun akademik. Selain itu admin dapat mengedit, dan menghapus data yang kurang tepat. Pada *Use Case* dikatakan tidak berhasil apabila ketika tambah data tahun akademik terdapat data kosong, maka website akan memunculkan alert. Selain itu ketika hapus data tahun akademik maka ada alert pada website.

Tabel 4. 14 *Use Case Specification* Melihat Data Presensi Admin

Nomor	14
Nama	: Melihat data presensi
Actor	: Admin
Extension	: -
Include	: <i>Login Admin</i>
Pre-Condition	: Admin memilih menu presensi yang ada pada sub menu akademik
Skenario utama	: <ul style="list-style-type: none"> <li>4. Admin memilih menu akademik</li> <li>5. Admin memilih presensi pada sub menu akademik</li> <li>6. Website menunjukkan daftar nama presensi siswa</li> </ul>
Post Condition	: Website menunjukkan presensi siswa
Skenario Abnormal	: -

Tabel 4.14 menjelaskan gambaran dari sebuah *Use Case* admin, yaitu melihat presensi, *Use Case* ini dimulai dari admin yang telah berhasil melakukan proses *login* kemudian website akan menampilkan halaman utama dengan benar. Selanjutnya admin memilih menu akademik, lalu admin memilih presensi pada sub menu akademik. Kemudian Website menampilkan halaman siswa yang akan dilihat presensinya. Admin dapat memilih “aksi” untuk melihat detail presensi siswa.

Tabel 4. 15 *Use Case Specification* Mengelola Data User

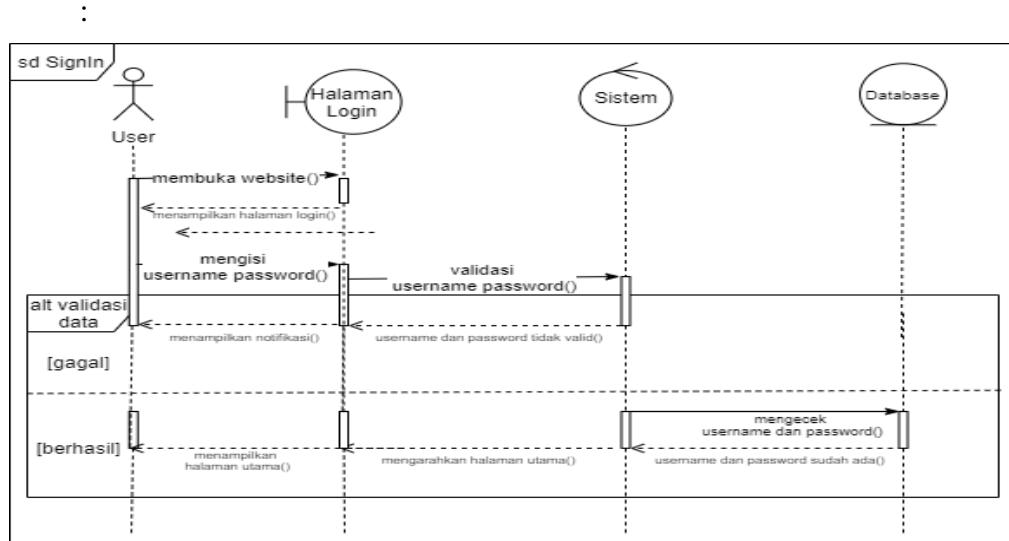
Nomor	15
Nama	: Mengelola Data User
Actor	: Admin
Extension	: -
Include	: <i>Login Admin</i>
Pre-Condition	: Admin memilih menu user yang ada pada sub menu akademik

Skenario utama :	
1.	Admin memilih menu akademik
2.	Admin memilih user pada sub menu akademik
3.	Admin mengelola data user
<i>Post Condition</i>	: Admin mengelola data user pada website
Skenario Abnormal	: Memunculkan alert dialog pada website

Tabel 4.15 diatas merupakan gambaran dari sebuah *Use Case* admin yaitu melakukan pengelolaan terhadap data user yang dimana *Use Case* dimulai ketika admin telah berhasil melakukan *login* ke dalam website, kemudian website menampilkan halaman utama, dan admin dapat memilih menu akademik lalu memilih user pada sub menu akademik. Website akan menampilkan halaman user, admin dapat mengelola data user. Admin dapat melakukan penambahan data user. Selain itu admin dapat mengedit, dan menghapus data yang kurang tepat. Pada *Use Case* dikatakan tidak berhasil apabila ketika tambah data user terdapat data kosong, maka website akan memunculkan alert. Selain itu ketika hapus data user maka akan ada alert pada website.

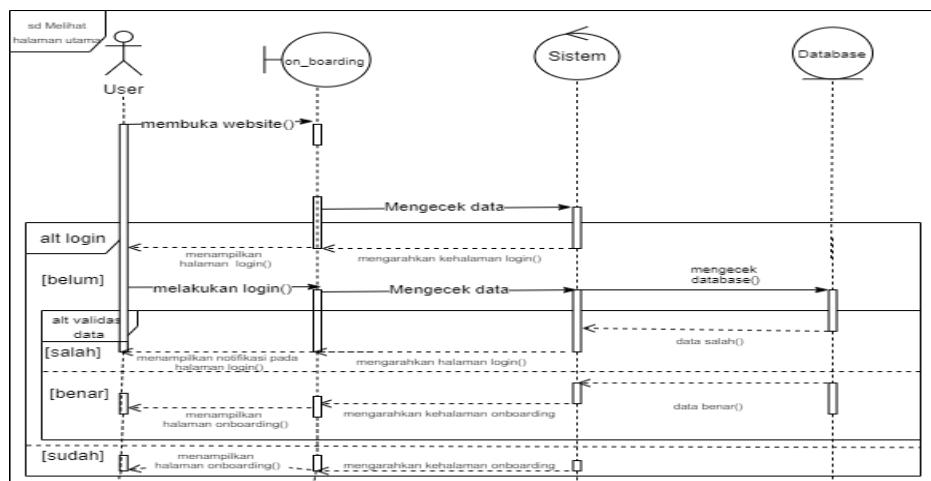
#### b. *Squence Diagram*

*Squence diagram* adalah sebuah *diagram* yang menjelaskan bagaimana suatu objek pada sebuah *Use Case* terlibat dalam sebuah skenario dan urutan pesan/informasi yang perlu ditukarkan antar objek yang dibutuhkan untuk dapat menjalankan sebuah *fungsionalitas* skenario. Berikut merupakan *squence diagram* pada website sistem informasi akademik SIAwarak



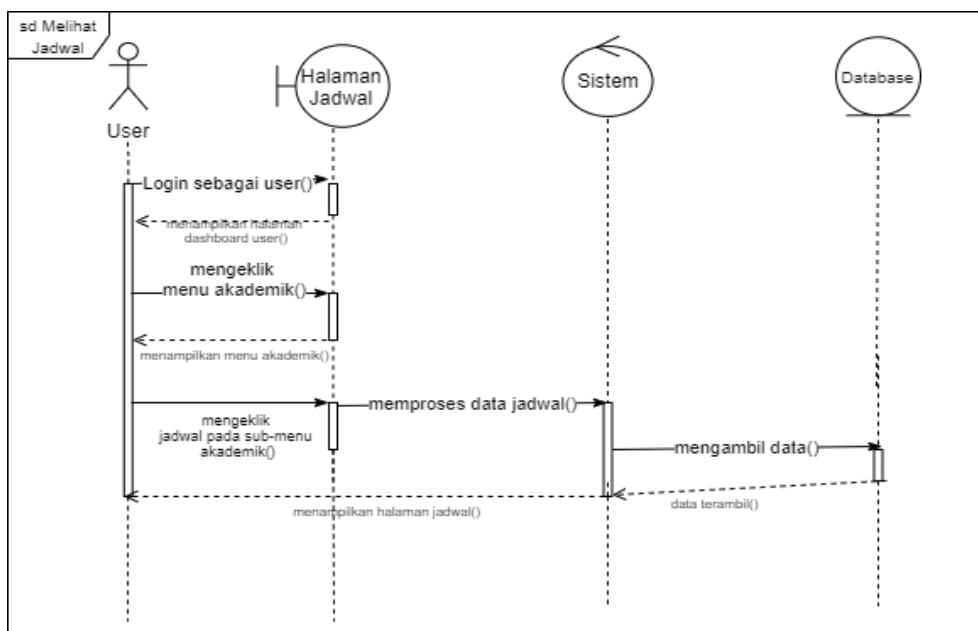
Gambar 4. 2 Squence Diagram Login Pengguna

Gambar 4. 2 menjelaskan bagaimana *squence diagram* dari *use case login* pada *actor* pengguna. Pengguna membuka website, lalu website akan menampilkan halaman *login* pengguna,kemudian pengguna mengisi *username* dan *password* pada halaman *login* pengguna. Lalu sistem akan memvalidasi *username* dan *password* kemudian jika tidak valid maka sistem akan menampilkan notifikasi namun jika valid maka sistem akan mengarahkan ke halaman utama pengguna.



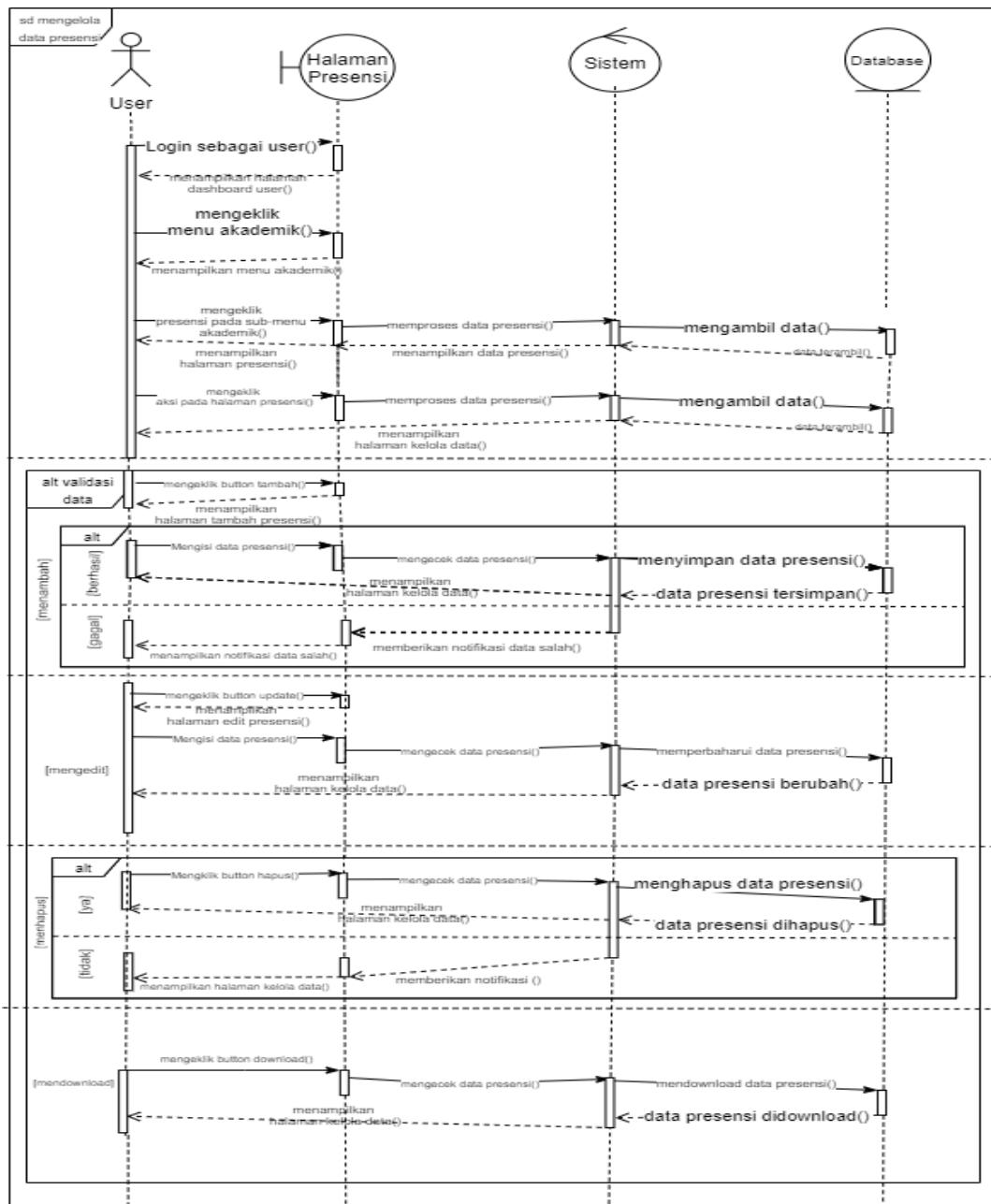
Gambar 4. 3 Squence Diagram Melihat Dashboard Pengguna

Gambar 4. 3 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* melihat halaman utama atau *dashboard* pada *actor* pengguna. Pengguna membuka website kemudian menampilkan halaman *login* dan pengguna mengisi *username* dan *password* pada halaman *login*, kemudian sistem akan memvalidasi, jika valid maka pengguna akan dapat menuju dan melihat halaman utama. Jika tidak maka akan muncul notifikasi dan memasukan kembali *username* dan *password*.



Gambar 4. 4 *Squence Diagram* Melihat Jadwal Pengguna

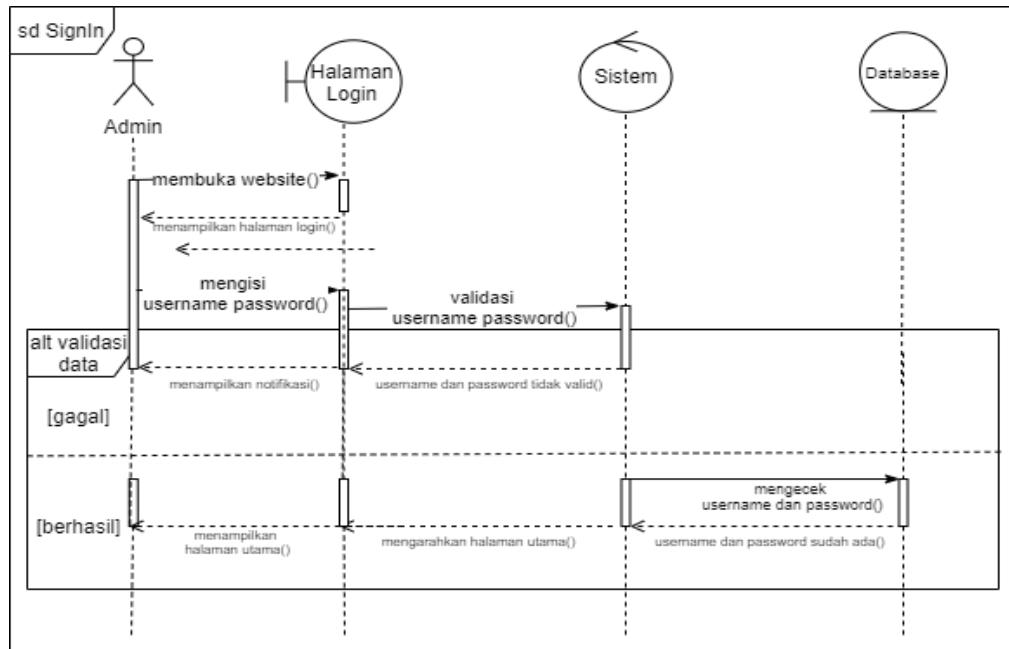
Gambar 4. 4 menjelaskan bagaimana *squence diagram* dari sebuah *UseCase* melihat data jadwal oleh pengguna. Pengguna dapat melakukan *login* sebagai pengguna, kemudian sistem akan menampilkan halaman *dashboard* pengguna, lalu pengguna dapat menekan menu akademik, maka sistem akan menampilkan menu yang ada pada akademik. Kemudian pengguna menekan menu jadwal pada sub menu akademik. Selanjutnya sistem akan memproses data jadwal dan sistem akan mengambil data dari *database* dan setelah itu sistem akan menampilkan data jadwal pada halaman jadwal.



Gambar 4. 5 *Squence Diagram* Mengelola Data Presensi Pengguna

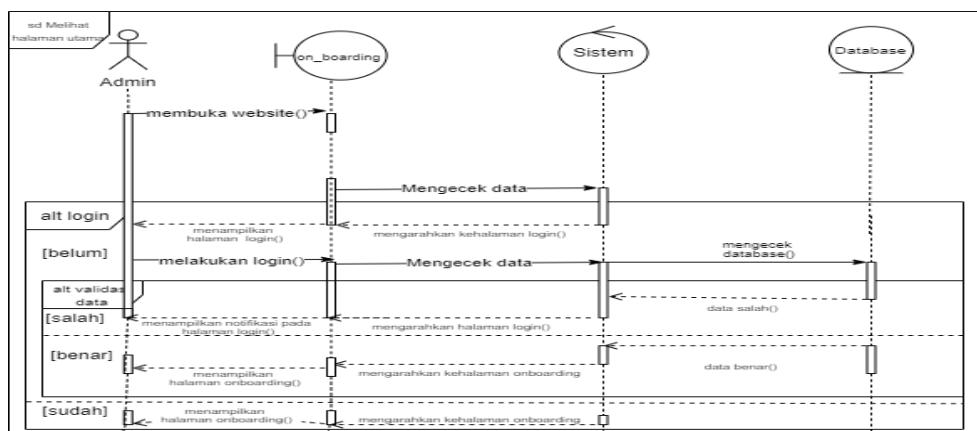
Gambar 4. 5 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* mengelola data presensi oleh pengguna. Pengguna dapat melakukan *login* sebagai pengguna, kemudian sistem akan menampilkan halaman *dashboard* pengguna, lalu pengguna dapat menekan menu akademik, maka sistem akan

menampilkan menu yang ada pada akademik. Kemudian pengguna menekan menu presensi pada sub menu akademik. Selanjutnya sistem akan memproses data presensi dan sistem akan mengambil data dari *database* dan setelah itu sistem akan menampilkan data presensi pada halaman presensi. Selanjutnya pengguna dapat menekan aksi pada halaman presensi, kemudian sistem akan memproses data presensi dan sistem akan mengambil data dari *database* setelah itu sistem akan menampilkan data presensi pada halaman detail presensi. Pada halaman ini, pengguna dapat menambah dengan menekan tombol tambah lalu sistem akan menampilkan halaman tambah presensi. Selanjutnya pengguna dapat mengisi data presensi, jika data berhasil maka sistem akan mengecek data presensi serta menyimpan data presensi kedalam *database* lalu sistem akan menampilkan data presensi yang telah ditambah di halaman presensi, namun jika data ada yang diisi kosong maka sistem akan menampilkan alert. Pengguna juga dapat mengedit dengan menekan tombol update lalu sistem akan menampilkan halaman edit presensi. Selanjutnya pengguna dapat mengisi data presensi yang akan diedit setelah itu sistem akan mengecek data presensi yang sudah diedit serta memperbarui data presensi pada *database* kemudian sistem menampilkan halaman presensi yang sudah diedit. Pengguna dapat menghapus dengan menekan tombol hapus selanjutnya sistem menampilkan notifikasi jika menekan tombol iya maka sistem mengecek data presensi yang akan dihapus, setelah itu menghapus data presensi pada database kemudian sistem akan menampilkan halaman presensi yang telah dihapus namun jika menekan tombol tidak maka sistem menampilkan halaman presensi. Pengguna juga dapat mendownload data presensi dengan menekan tombol download lalu sistem mengecek data presensi selanjutnya mendownload data presensi yang ada pada *database* lalu menampilkan hasil download yang ada pada website sistem informasi akademik SIAwarak.



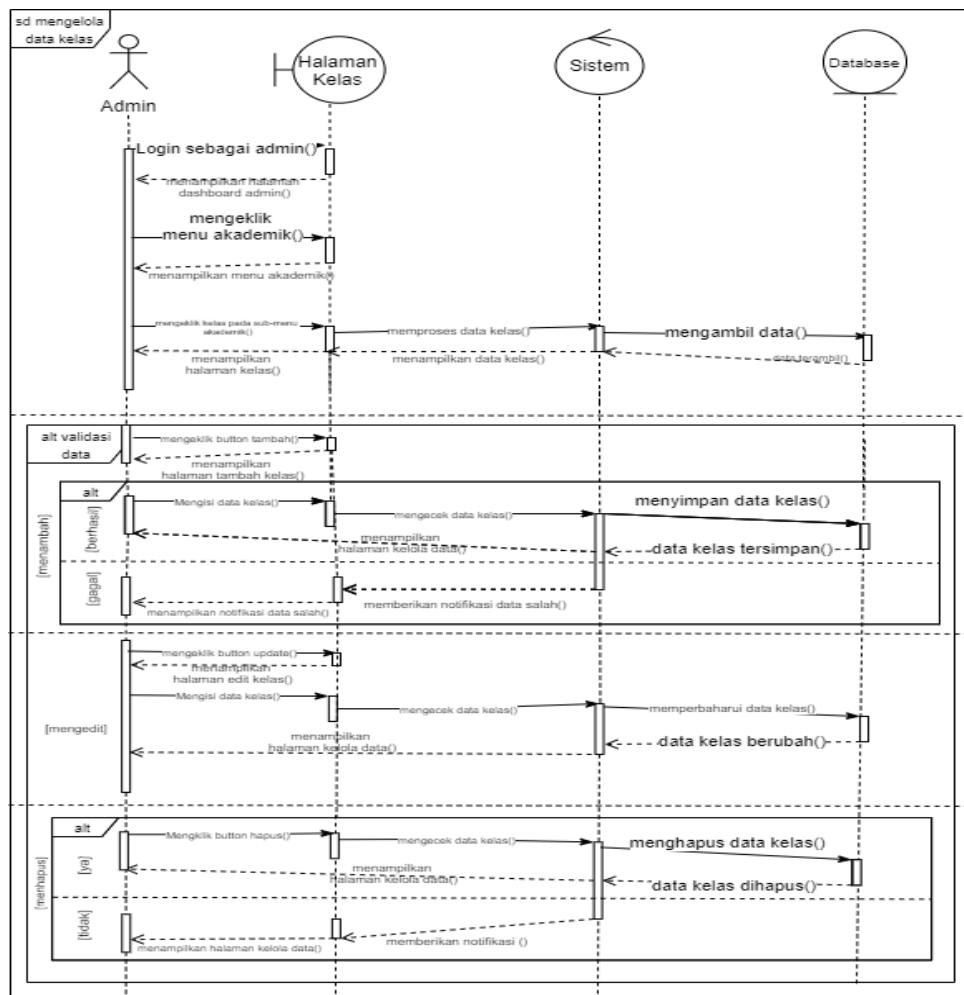
Gambar 4. 6 Squence Diagram Login Admin

Gambar 4. 6 menjelaskan bagaimana *squence diagram* dari *use case login* pada *actor admin*. Admin membuka website, lalu website akan menampilkan halaman *login* admin,kemudian admin mengisi *username* dan *password* pada halaman *login* admin. Lalu sistem akan memvalidasi *username* dan *password* kemudian jika tidak valid maka sistem akan menampilkan notifikasi namun jika valid maka sistem akan mengarahkan ke halaman utama pengguna.



Gambar 4. 7 Squence Diagram Melihat Dashboard Admin

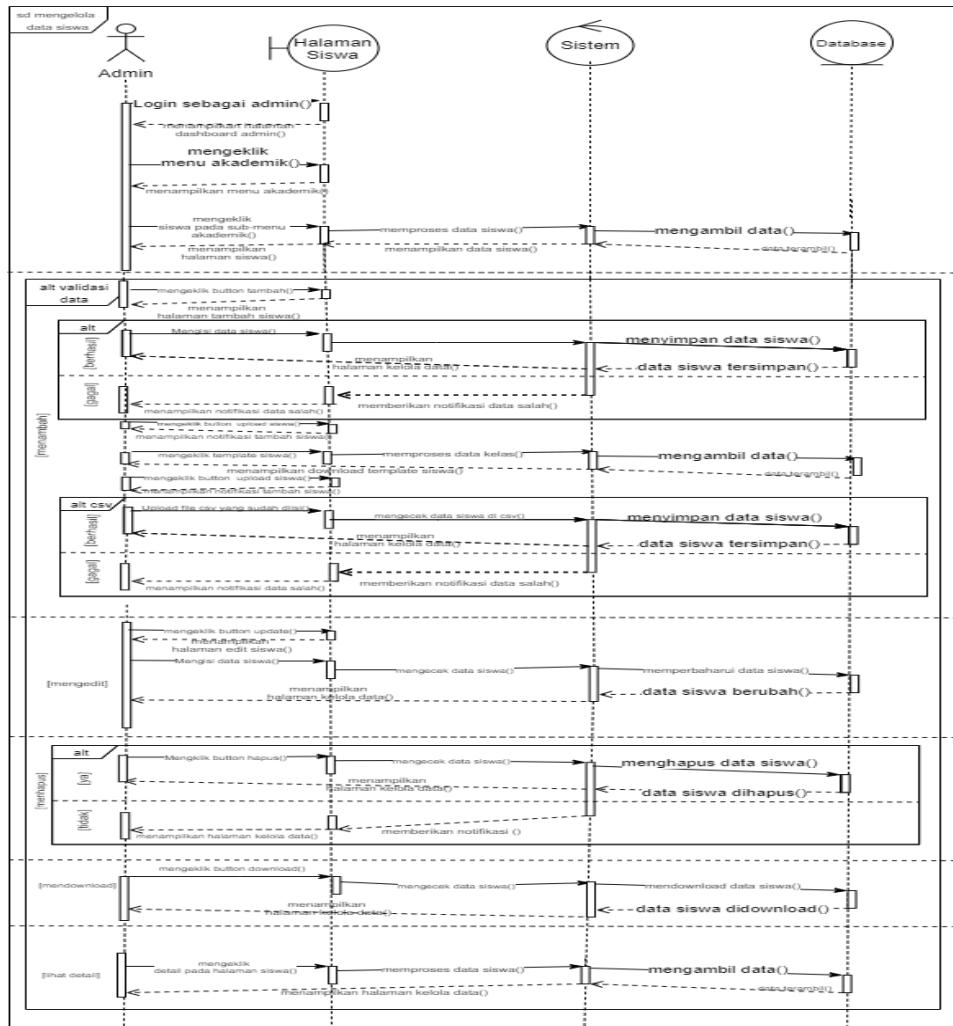
Gambar 4. 7 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* melihat halaman utama atau *dashboard* pada *actor* admin. Admin membuka website kemudian menampilkan halaman *login* dan admin mengisi *username* dan *password* pada halaman *sign\_in*, kemudian sistem akan memvalidasi, jika valid maka admin akan dapat menuju dan melihat halaman utama. Jika tidak maka akan muncul notifikasi dan memasukan kembali *username* dan *password*.



Gambar 4. 8 *Squence Diagram* Mengelola Data Kelas Admin

Gambar 4. 8 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* mengelola data kelas oleh admin. Admin dapat melakukan *login* sebagai

admin, kemudian sistem akan menampilkan halaman *dashboard* admin, lalu admin dapat menekan menu akademik, maka sistem akan menampilkan menu yang ada pada akademik. Kemudian admin menekan menu kelas pada sub menu akademik. Selanjutnya sistem akan memproses data kelas dan sistem akan mengambil data dari *database* setelah itu sistem akan menampilkan data kelas pada halaman kelas. Pada halaman ini, admin dapat menambah dengan menekan tombol tambah lalu sistem akan menampilkan halaman tambah kelas. Selanjutnya admin dapat mengisi data kelas, jika data berhasil maka sistem akan mengecek data kelas serta menyimpan data kelas kedalam *database* lalu sistem akan menampilkan data kelas yang telah ditambah di halaman kelas, namun jika data ada yang diisi kosong maka sistem akan menampilkan alert. Admin juga dapat mengedit dengan menekan tombol update lalu sistem akan menampilkan halaman edit kelas. Selanjutnya admin dapat mengisi data kelas yang akan diedit setelah itu sistem akan mengecek data kelas yang sudah diedit serta memperbarui data kelas pada *database* kemudian sistem menampilkan halaman kelas yang sudah diedit. Admin dapat menghapus dengan menekan tombol hapus selanjutnya sistem menampilkan notifikasi jika menekan tombol iya maka sistem mengecek data kelas yang akan dihapus, setelah itu menghapus data kelas pada *database* kemudian sistem akan menampilkan halaman kelas yang telah dihapus namun jika menekan tombol tidak maka sistem menampilkan halaman kelas yang ada pada website sistem informasi akademik SIAwarak.

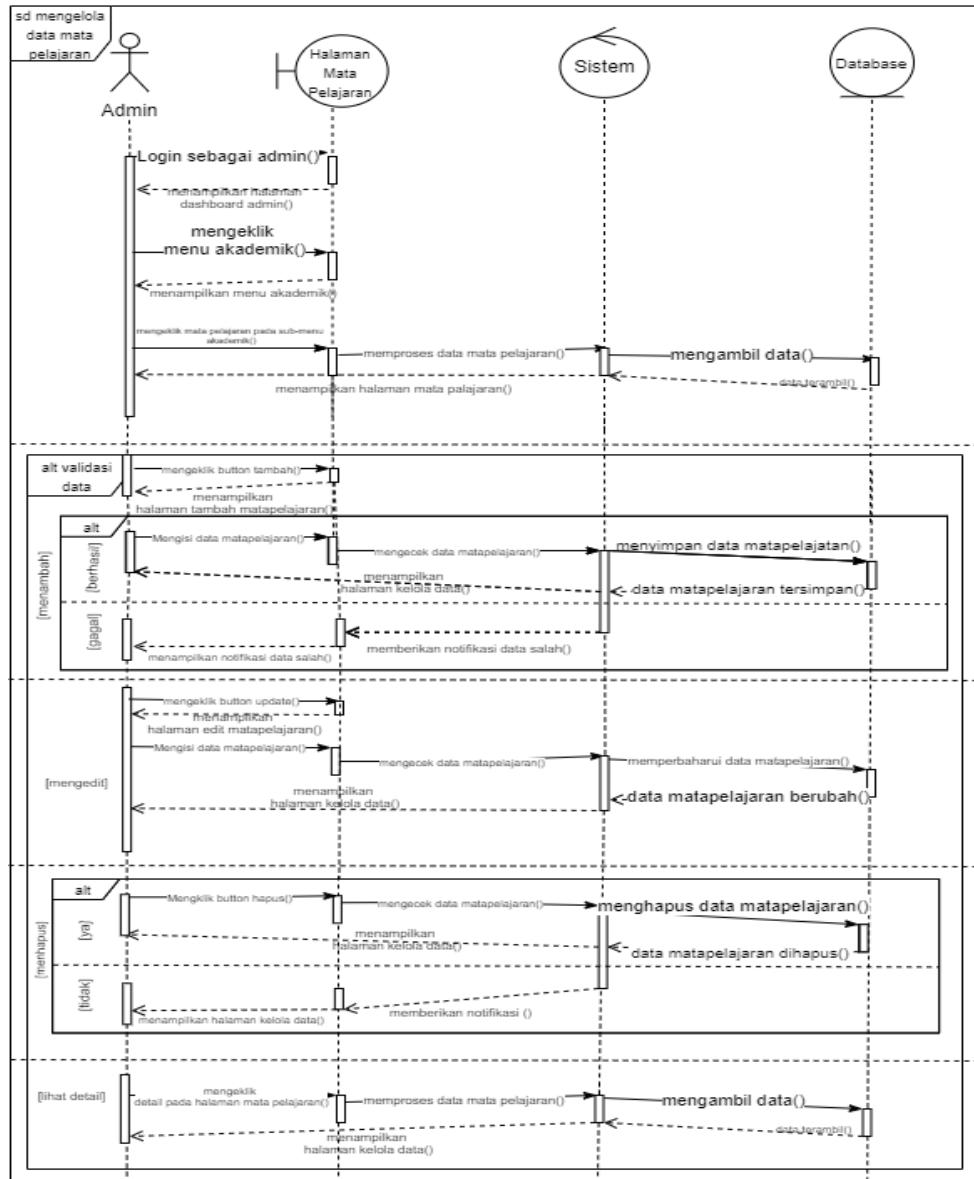


Gambar 4. 9 *Squence Diagram* Mengelola Data Siswa Admin

Gambar 4. 9 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* mengelola data siswa oleh admin. Admin dapat melakukan *login* sebagai admin, kemudian sistem akan menampilkan halaman *dashboard* admin, lalu admin dapat menekan menu akademik, maka sistem akan menampilkan menu yang ada pada akademik. Kemudian admin menekan menu siswa pada submenu akademik. Selanjutnya sistem akan memproses data siswa dan sistem akan mengambil data dari *database* setelah itu sistem akan menampilkan data siswa pada halaman siswa. Pada halaman ini, admin dapat menambah dengan dua cara yaitu upload melalui *csv* dan dengan menekan tombol tambah. Jika

menekan tombol tambah lalu sistem akan menampilkan halaman tambah siswa. Selanjutnya admin dapat mengisi data siswa, jika data berhasil maka sistem akan mengecek data siswa serta menyimpan data siswa kedalam *database* lalu sistem akan menampilkan data siswa yang telah ditambah di halaman siswa, namun jika data ada yang diisi kosong maka sistem akan menampilkan *alert*. Sementara itu dengan tombol upload *csv* yaitu dengan menekan tombol upload siswa kemudian sistem akan menampilkan notifikasi tambah siswa, disana sudah tertera template siswa jika akan melakukan upload siswa dengan cara menekan tombol template siswa lalu sistem akan memproses data dan mengambil data pada *database* kemudian menampilkan hasil download template siswa untuk ditambahkan dengan *csv*. Setelah data ditambahkan buka kembali tombol upload siswa kemudian menekan tombol *choose file* untuk memasukan file yang sudah diisi, jika data berhasil maka sistem akan mengecek data siswa di *csv* serta menyimpan data siswa kedalam *database* lalu sistem akan menampilkan data siswa yang telah ditambah di halaman siswa, namun jika data ada yang diisi kosong maka sistem akan menampilkan *alert*. Admin juga dapat mengedit dengan menekan tombol update lalu sistem akan menampilkan halaman edit siswa. Selanjutnya admin dapat mengisi data siswa yang akan diedit setelah itu sistem akan mengecek data siswa yang sudah diedit serta memperbarui data siswa pada *database* kemudian sistem menampilkan halaman siswa yang sudah diedit. Admin dapat menghapus dengan menekan tombol hapus selanjutnya sistem menampilkan notifikasi jika menekan tombol iya maka sistem mengecek data siswa yang akan dihapus, setelah itu menghapus data siswa pada *database* kemudian sistem akan menampilkan halaman siswa yang telah dihapus namun jika menekan tombol tidak maka sistem menampilkan halaman siswa. Admin juga dapat mendownload data siswa dengan menekan tombol download lalu sistem mengecek data siswa selanjutnya mendownload data siswa yang ada pada *database* lalu menampilkan hasil download yang ada pada website sistem

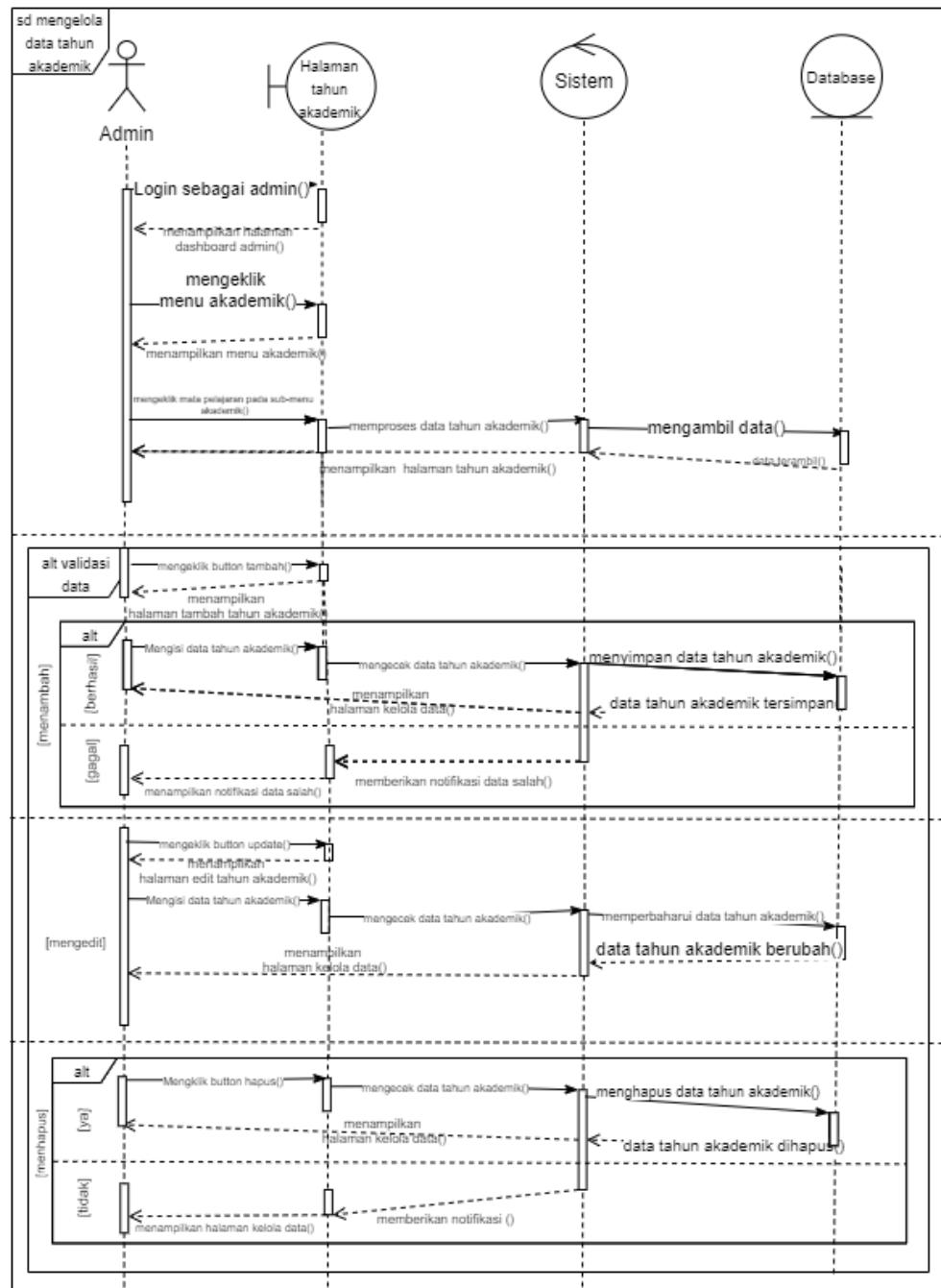
informasi akademik. Untuk melihat detail siswa secara detail menekan tombol lihat siswa detail kemudian sistem memproses data siswa lalu mengambil data pada *database* kemudian ditampilkan dihalaman siswa.



Gambar 4. 10 *Squence Diagram* Mengelola Data Mata Pelajaran Admin

Gambar 4.10 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* mengelola data mata pelajaran oleh admin. Admin dapat melakukan

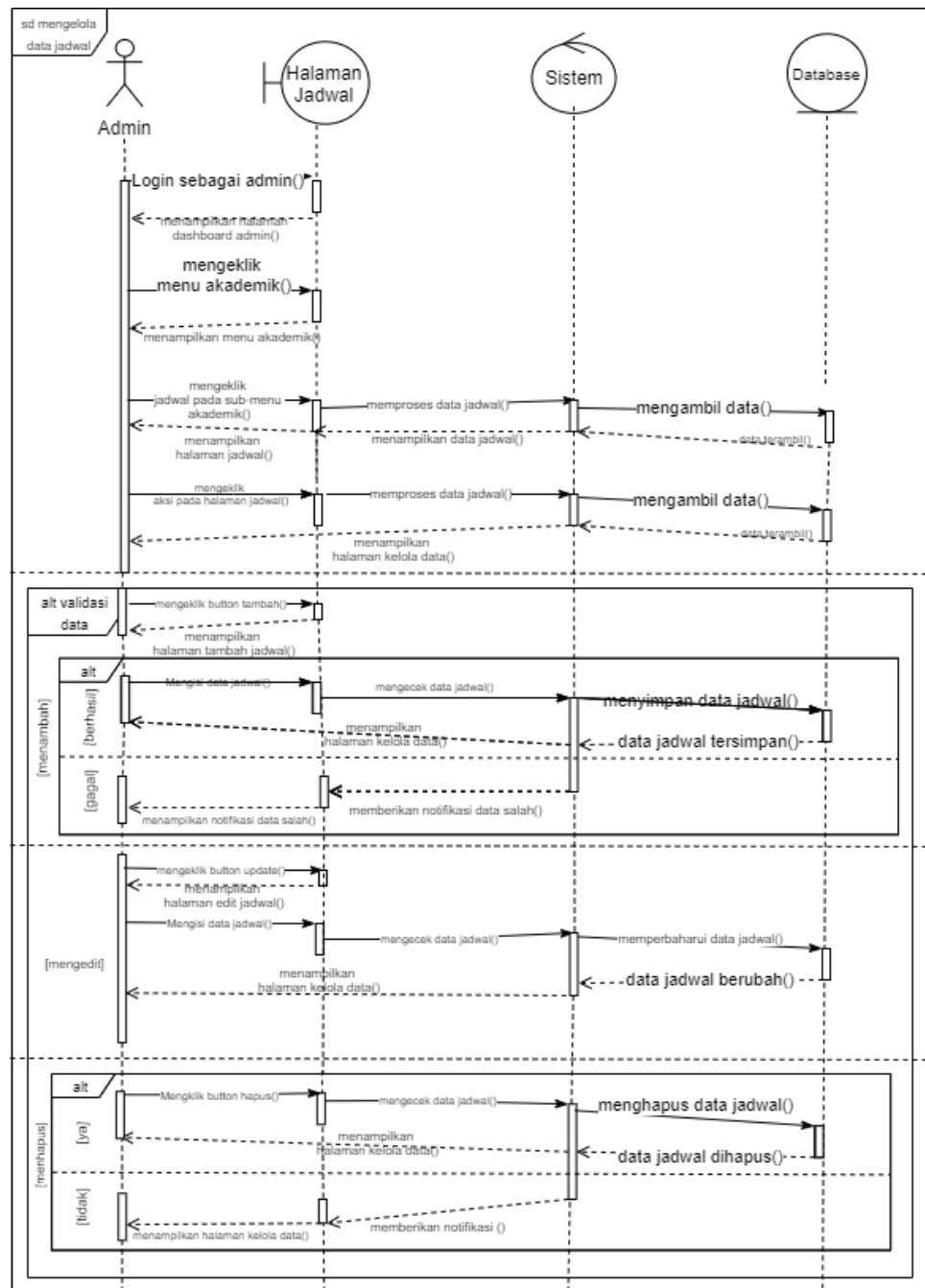
*login* sebagai admin, kemudian sistem akan menampilkan halaman *dashboard* admin, lalu admin dapat menekan menu akademik, maka sistem akan menampilkan menu yang ada pada akademik. Kemudian admin menekan menu mata pelajaran pada sub menu akademik. Selanjutnya sistem akan memproses data mata pelajaran dan sistem akan mengambil data dari *database* setelah itu sistem akan menampilkan data mata pelajaran pada halaman mata pelajaran. Pada halaman ini, admin dapat menambah dengan menekan tombol tambah lalu sistem akan menampilkan halaman tambah mata pelajaran. Selanjutnya admin dapat mengisi data mata pelajaran, jika data berhasil maka sistem akan mengecek data mata pelajaran serta menyimpan data mata pelajaran kedalam *database* lalu sistem akan menampilkan data mata pelajaran yang telah ditambah di halaman mata pelajaran, namun jika data ada yang diisi kosong maka sistem akan menampilkan alert. Admin juga dapat mengedit dengan menekan tombol update lalu sistem akan menampilkan halaman edit mata pelajaran. Selanjutnya admin dapat mengisi data mata pelajaran yang akan diedit setelah itu sistem akan mengecek data mata pelajaran yang sudah diedit serta memperbarui data mata pelajaran pada *database* kemudian sistem menampilkan halaman mata pelajaran yang sudah diedit. Admin dapat menghapus dengan menekan tombol hapus selanjutnya sistem menampilkan notifikasi jika menekan tombol iya maka sistem mengecek data mata pelajaran yang akan dihapus, setelah itu menghapus data mata pelajaran pada *database* kemudian sistem akan menampilkan halaman mata pelajaran yang telah dihapus namun jika menekan tombol tidak maka sistem menampilkan halaman mata pelajaran . Untuk melihat detail mata pelajaran secara detail menekan tombol lihat mata pelajaran detail kemudian sistem memproses data mata pelajaran lalu mengambil data pada *database* kemudian ditampilkan dihalaman mata pelajaran yang ada pada website sistem informasi akademik SIAwarak.



Gambar 4.11 *Squence Diagram* Mengelola Data Tahun Akademik Admin

Gambar 4.11 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* mengelola data tahun akademik oleh admin. Admin dapat melakukan *login* sebagai admin, kemudian sistem akan menampilkan halaman *dashboard*

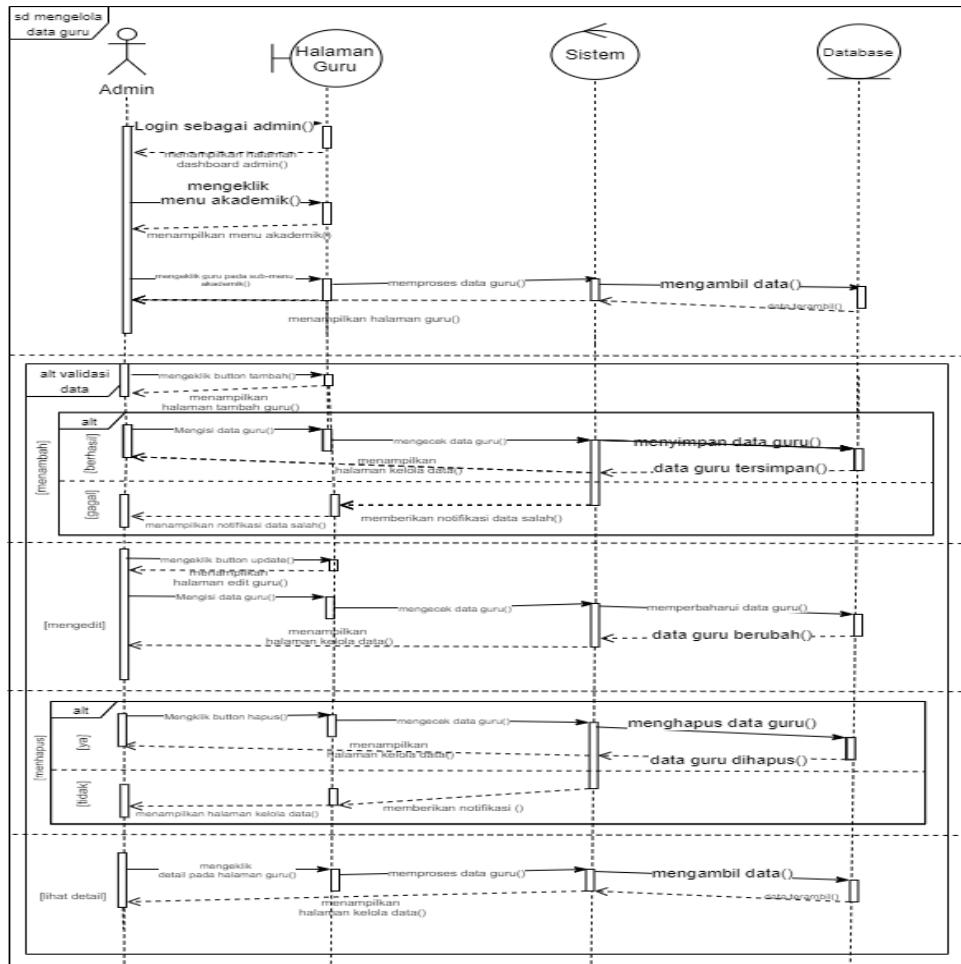
admin, lalu admin dapat menekan menu akademik, maka sistem akan menampilkan menu yang ada pada akademik. Kemudian admin menekan menu tahun akademik pada sub menu akademik. Selanjutnya sistem akan memproses data tahun akademik dan sistem akan mengambil data dari *database* setelah itu sistem akan menampilkan data tahun akademik pada halaman tahun akademik. Pada halaman ini, admin dapat menambah dengan menekan tombol tambah lalu sistem akan menampilkan halaman tambah tahun akademik. Selanjutnya admin dapat mengisi data tahun akademik, jika data berhasil maka sistem akan mengecek data tahun akademik serta menyimpan data tahun akademik kedalam *database* lalu sistem akan menampilkan data tahun akademik yang telah ditambah di halaman tahun akademik, namun jika data ada yang diisi kosong maka sistem akan menampilkan alert. Admin juga dapat mengedit dengan menekan tombol update lalu sistem akan menampilkan halaman edit tahun akademik. Selanjutnya admin dapat mengisi data tahun akademik yang akan diedit setelah itu sistem akan mengecek data tahun akademik yang sudah diedit serta memperbarui data tahun akademik pada *database* kemudian sistem menampilkan halaman tahun akademik yang sudah diedit. Admin dapat menghapus dengan menekan tombol hapus selanjutnya sistem menampilkan notifikasi jika menekan tombol iya maka sistem mengecek data tahun akademik yang akan dihapus, setelah itu menghapus data tahun akademik pada *database* kemudian sistem akan menampilkan halaman tahun akademik yang telah dihapus namun jika menekan tombol tidak maka sistem menampilkan halaman tahun akademik.



Gambar 4. 12 *Sequence Diagram* Mengelola Data Jadwal Pelajaran Admin

Gambar 4.12 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* mengelola data jadwal pelajaran oleh admin. Admin dapat melakukan *login* sebagai admin, kemudian sistem akan menampilkan halaman *dashboard*

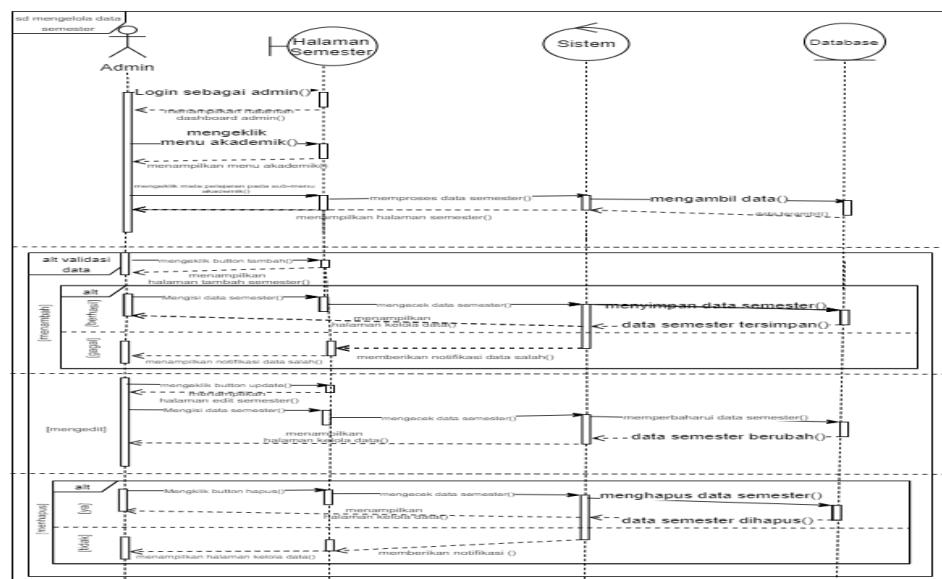
admin, lalu admin dapat menekan menu akademik, maka sistem akan menampilkan menu yang ada pada akademik. Kemudian admin menekan menu jadwal pada sub menu akademik. Selanjutnya sistem akan memproses data jadwal dan sistem akan mengambil data dari *database* dan setelah itu sistem akan menampilkan data jadwal pada halaman jadwal. Selanjutnya admin dapat menekan aksi pada halaman jadwal, kemudian sistem akan memproses data jadwal dan sistem akan mengambil data dari *database* setelah itu sistem akan menampilkan data jadwal pada halaman detail jadwal. Pada halaman ini, admin dapat menambah dengan menekan tombol tambah lalu sistem akan menampilkan halaman tambah jadwal. Selanjutnya admin dapat mengisi data jadwal, jika data berhasil maka sistem akan mengecek data jadwal serta menyimpan data jadwal kedalam *database* lalu sistem akan menampilkan data jadwal yang telah ditambah di halaman jadwal, namun jika data ada yang diisi kosong maka sistem akan menampilkan alert. Admin juga dapat mengedit dengan menekan tombol update lalu sistem akan menampilkan halaman edit jadwal. Selanjutnya admin dapat mengisi data jadwal yang akan diedit setelah itu sistem akan mengecek data jadwal yang sudah diedit serta memperbarui data jadwal pada *database* kemudian sistem menampilkan halaman jadwal yang sudah diedit. Admin dapat menghapus dengan menekan tombol hapus semua data selanjutnya sistem menampilkan notifikasi jika menekan tombol iya maka sistem mengecek data jadwal yang akan dihapus, setelah itu menghapus data jadwal pada database kemudian sistem akan menampilkan halaman jadwal yang telah dihapus namun jika menekan tombol tidak maka sistem menampilkan halaman jadwal yang ada pada website sistem informasi akademik SIAwarak.



Gambar 4. 13 *Squence Diagram* Mengelola Data Guru Admin

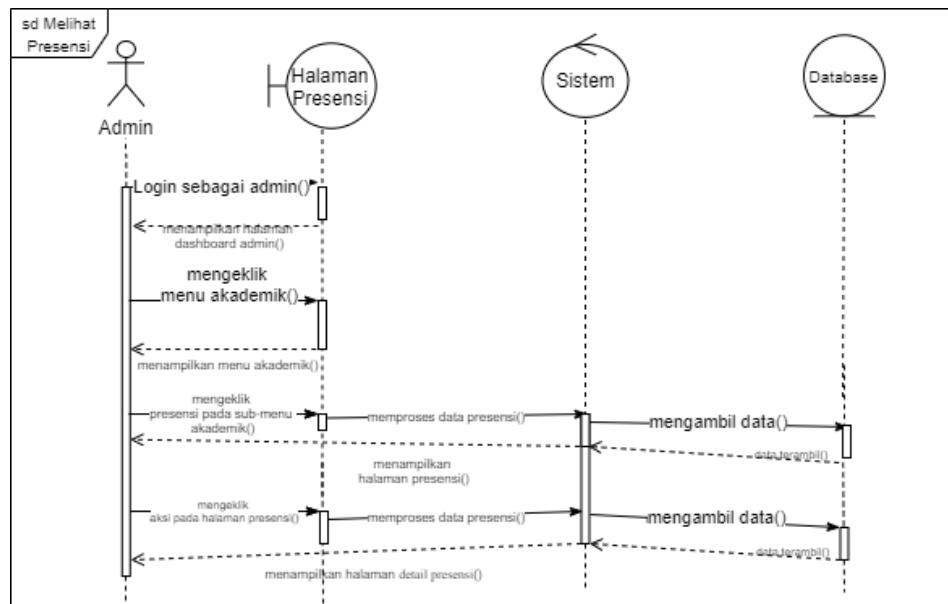
Gambar 4.13 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* mengelola data guru oleh admin. Admin dapat melakukan *login* sebagai admin, kemudian sistem akan menampilkan halaman *dashboard* admin, lalu admin dapat menekan menu akademik, maka sistem akan menampilkan menu yang ada pada akademik. Kemudian admin menekan menu guru pada sub menu akademik. Selanjutnya sistem akan memproses data guru dan sistem akan mengambil data dari *database* setelah itu sistem akan menampilkan data guru pada halaman guru. Pada halaman ini, admin dapat menambah dengan menekan tombol tambah lalu sistem akan menampilkan halaman tambah guru.

Selanjutnya admin dapat mengisi data guru, jika data berhasil maka sistem akan mengecek data guru serta menyimpan data guru kedalam *database* lalu sistem akan menampilkan data guru yang telah ditambah di halaman guru, namun jika data ada yang diisi kosong maka sistem akan menampilkan alert. Admin juga dapat mengedit dengan menekan tombol update lalu sistem akan menampilkan halaman edit guru. Selanjutnya admin dapat mengisi data guru yang akan diedit setelah itu sistem akan mengecek data guru yang sudah diedit serta memperbarui data guru pada *database* kemudian sistem menampilkan halaman guru yang sudah diedit. Admin dapat menghapus dengan menekan tombol hapus selanjutnya sistem menampilkan notifikasi jika menekan tombol iya maka sistem mengecek data guru yang akan dihapus, setelah itu menghapus data guru pada *database* kemudian sistem akan menampilkan halaman guru yang telah dihapus namun jika menekan tombol tidak maka sistem menampilkan halaman guru. Untuk melihat detail guru secara detail menekan tombol lihat guru detail kemudian sistem memproses data guru lalu mengambil data pada *database* kemudian ditampilkan dihalaman guru yang ada pada website sistem informasi akademik SIAwarak.



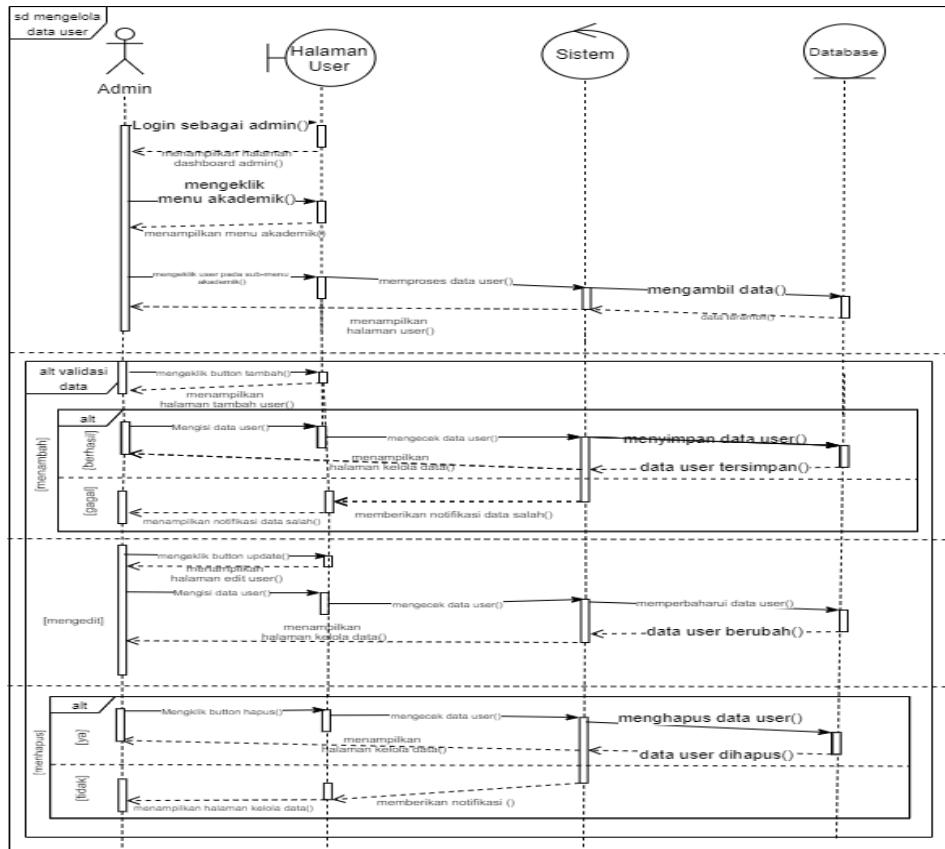
Gambar 4. 14 Squence Diagram Mengelola Data Semester Admin

Gambar 4.14 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* mengelola data semester oleh admin. Admin dapat melakukan *login* sebagai admin, kemudian sistem akan menampilkan halaman *dashboard* admin, lalu admin dapat menekan menu akademik, maka sistem akan menampilkan menu yang ada pada akademik. Kemudian admin menekan menu semester pada sub menu akademik. Selanjutnya sistem akan memproses data semester dan sistem akan mengambil data dari *database* setelah itu sistem akan menampilkan data semester pada halaman semester. Pada halaman ini, admin dapat menambah dengan menekan tombol tambah lalu sistem akan menampilkan halaman tambah semester. Selanjutnya admin dapat mengisi data semester, jika data berhasil maka sistem akan mengecek data semester serta menyimpan data semester kedalam *database* lalu sistem akan menampilkan data semester yang telah ditambah di halaman semester, namun jika data ada yang diisi kosong maka sistem akan menampilkan alert. Admin juga dapat mengedit dengan menekan tombol update lalu sistem akan menampilkan halaman edit semester. Selanjutnya admin dapat mengisi data semester yang akan diedit setelah itu sistem akan mengecek data semester yang sudah diedit serta memperbarui data semester pada *database* kemudian sistem menampilkan halaman semester yang sudah diedit. Admin dapat menghapus dengan menekan tombol hapus selanjutnya sistem menampilkan notifikasi jika menekan tombol iya maka sistem mengecek data semester yang akan dihapus, setelah itu menghapus data semester pada *database* kemudian sistem akan menampilkan halaman semester yang telah dihapus namun jika menekan tombol tidak maka sistem menampilkan halaman semester.



Gambar 4. 15 *Squence Diagram* Melihat Data Presensi Admin

Gambar 4.15 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* melihat data presensi oleh admin. Admin dapat melakukan *login* sebagai admin, kemudian sistem akan menampilkan halaman *dashboard* admin, lalu admin dapat menekan menu akademik, maka sistem akan menampilkan menu yang ada pada akademik. Kemudian admin menekan menu presensi pada sub menu akademik. Selanjutnya sistem akan memproses data presensi dan sistem akan mengambil data dari *database* dan setelah itu sistem akan menampilkan data presensi pada halaman presensi. Selanjutnya admin dapat menekan aksi pada halaman presensi untuk melihat detail presensi, kemudian sistem akan memproses data presensi dan sistem akan mengambil data dari *database* setelah itu sistem akan menampilkan data detail presensi pada halaman detail presensi. Pada halaman ini, admin dapat melihat data detail presensi.



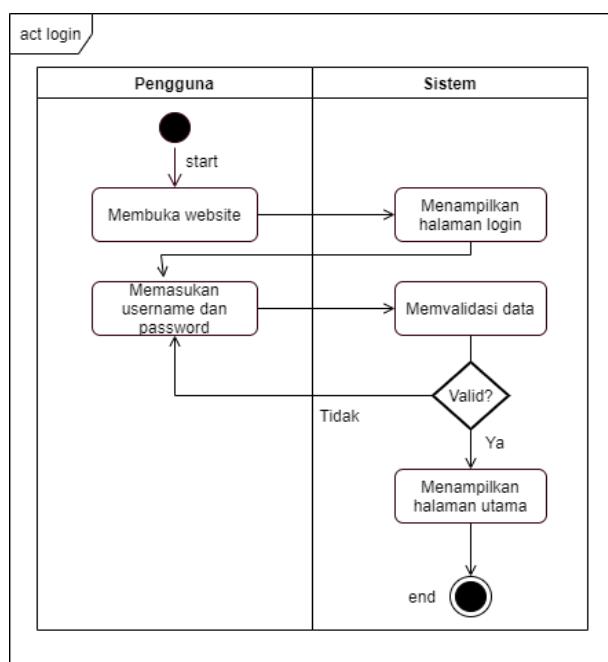
Gambar 4. 16 *Squence Diagram* Mengelola Data User Admin

Gambar 4.16 menjelaskan bagaimana *squence diagram* dari sebuah *Use Case* mengelola data *user* oleh admin. Admin dapat melakukan *login* sebagai admin, kemudian sistem akan menampilkan halaman *dashboard admin*, lalu admin dapat menekan menu akademik, maka sistem akan menampilkan menu yang ada pada akademik. Kemudian admin menekan menu *user* pada sub menu akademik. Selanjutnya sistem akan memproses data *user* dan sistem akan mengambil data dari *database* setelah itu sistem akan menampilkan data *user* pada halaman *user*. Pada halaman ini, admin dapat menambah dengan menekan tombol tambah lalu sistem akan menampilkan halaman tambah *user*. Selanjutnya admin dapat mengisi data *user*, jika data berhasil maka sistem akan mengecek data *user* serta menyimpan data *user* kedalam *database* lalu sistem akan menampilkan data *user* yang telah ditambah di halaman *user*,

namun jika data ada yang diisi kosong maka sistem akan menampilkan alert. Admin juga dapat mengedit dengan menekan tombol update lalu sistem akan menampilkan halaman edit *user*. Selanjutnya admin dapat mengisi data *user* yang akan diedit setelah itu sistem akan mengecek data *user* yang sudah diedit serta memperbaharui data *user* pada *database* kemudian sistem menampilkan halaman *user* yang sudah diedit. Admin dapat menghapus dengan menekan tombol hapus selanjutnya sistem menampilkan notifikasi jika menekan tombol iya maka sistem mengecek data *user* yang akan dihapus, setelah itu menghapus data *user* pada *database* kemudian sistem akan menampilkan halaman *user* yang telah dihapus namun jika menekan tombol tidak maka sistem menampilkan halaman *user*.

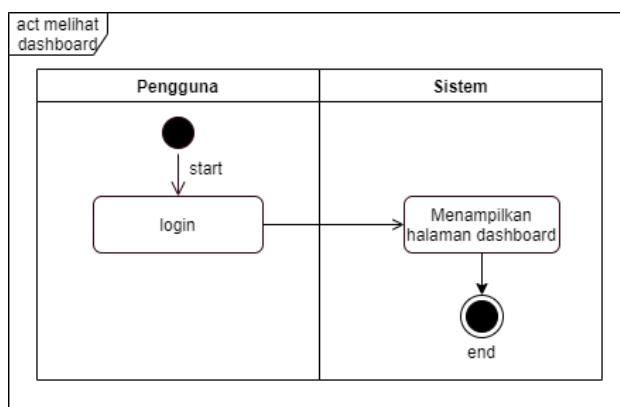
### c. Activity Diagram

*Activity Diagram* adalah sebuah representasi yang menjelaskan sebuah alir dari aktivitas suatu sistem yang berjalan. Berikut merupakan *activity diagram* dari website sistem informasi akademik SIAwarak :



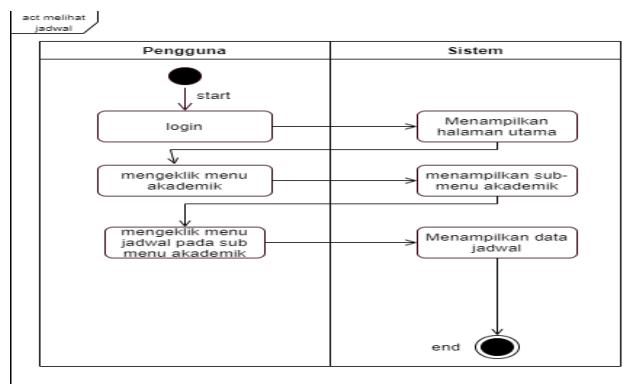
Gambar 4. 17 *Activity Diagram Login Pengguna*

Gambar 4.17 menunjukkan bagaimana *activity diagram* dari sebuah *Use Case login* pengguna oleh pengguna. Pengguna membuka website kemudian menampilkan halaman *login*, kemudian pengguna melakukan *login* pada halaman *login* dengan *username* dan *password*. Setelah itu sistem akan memvalidasi akun. Setelahnya dilakukan proses pemeriksaan akun apakah akun benar atau salah, jika benar maka akan menuju ke halaman utama, namun jika salah maka akan ada *alert* dan masukan kembali *username* dan *password*.



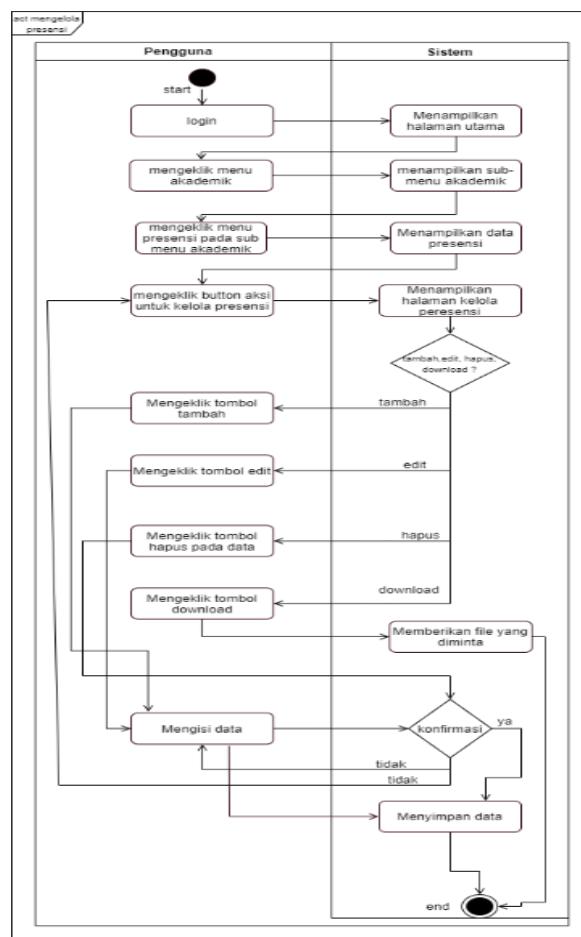
Gambar 4. 18 *Activity Diagram* Melihat Halaman *Dashboard* Pengguna

Gambar 4.18 menjelaskan bagaimana *activity* suatu *diagram* dari sebuah *Use Case* melihat menu utama atau dashboard oleh aktor pengguna. Pengguna dapat membuka website dan melakukan proses *login*, lalu setelah itu pengguna dapat membuka halaman utama.



Gambar 4. 19 *Activity Diagram* Melihat Jadwal Pelajaran Pengguna

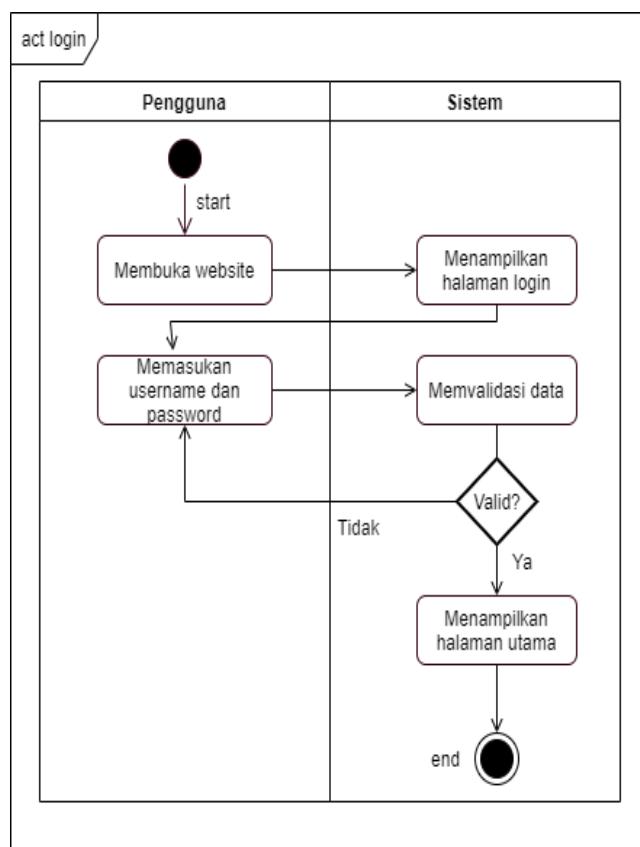
Gambar 4.19 menunjukkan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* melihat jadwal oleh aktor pengguna. Pengguna dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, pengguna menekan tombol akademik. Setelah itu pengguna menekan jadwal pada sub menu akademik, kemudian pengguna diarahkan ke halaman melihat jadwal pelajaran.



Gambar 4. 20 *Activity Diagram* Mengelola Data Presensi Pengguna

Gambar 4.20 menunjukkan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* mengelola data presensi oleh aktor pengguna. Pengguna dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, pengguna menekan tombol akademik. Setelah itu pengguna menekan presensi pada sub menu akademik, kemudian pengguna diarahkan ke halaman presensi siswa. Untuk

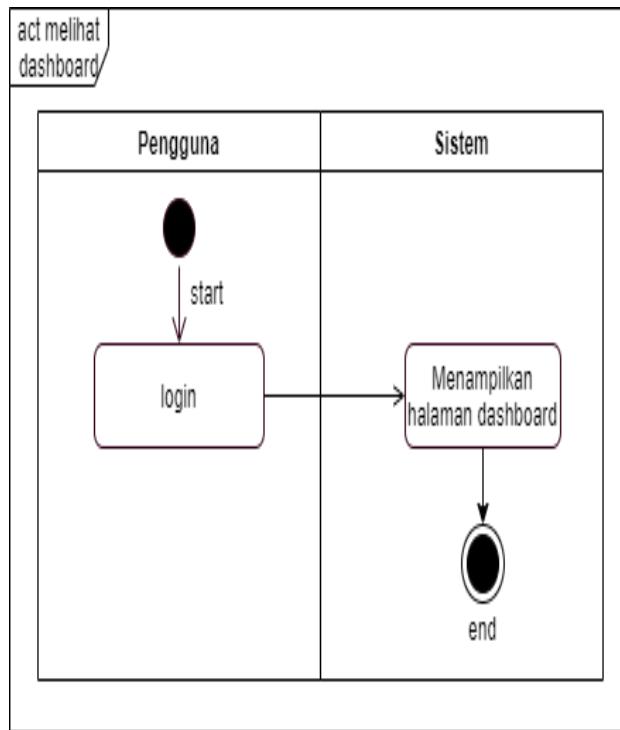
mengelola lebih detail presensi menekan tombol aksi. Pada halaman kelola presensi pengguna dapat menambah, mengedit, menghapus ataupun mendownload presensi siswa. Untuk bagian menambah, setelah presensi dikelola sistem akan mengkonfirmasi data yang telah dikelola. Setelah berhasil presensi, sistem dapat menyimpan hasil dari pengelolaan data yang dilakukan. Sementara itu untuk mengedit setelah presensi dikelola sistem akan menyimpan hasil dari data presensi. Serta untuk mengkonfirmasi data yang akan dihapus. Ketika tidak dihapus maka akan kembali ke halaman kelola presensi. Namun ketika iya, presensi dapat menyimpan hasil dari pengelolaan data yang dilakukan. Selanjutnya untuk mendownload data presensi sistem akan memberikan file yang diminta.



Gambar 4. 21 *Activity Diagram Login Admin*

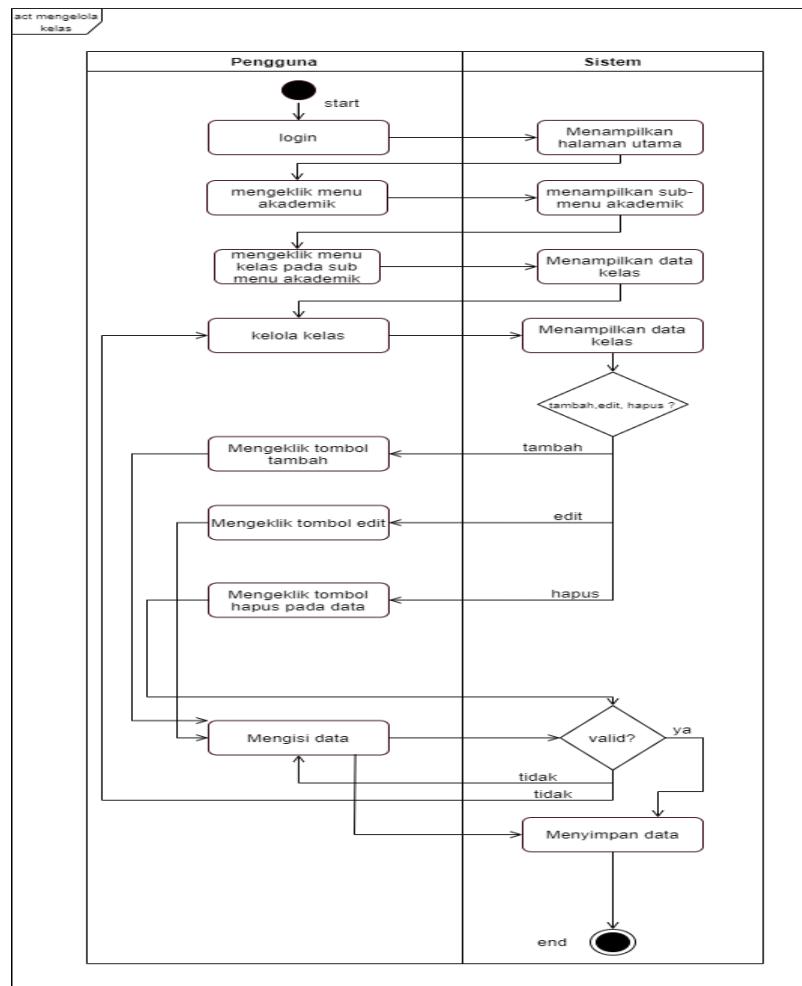
Gambar 4.21 menunjukkan bagaimana *activity diagram* dari sebuah *Use Case login* admin oleh admin. Admin membuka website kemudian menampilkan

halaman *login*, kemudian admin melakukan login pada halaman *login* dengan *username* dan *password*. Setelah itu sistem akan memvalidasi akun. Setelahnya dilakukan proses pemeriksaan akun apakah akun benar atau salah, jika benar maka akan menuju ke halaman utama, namun jika salah maka akan ada *alert* dan masukan kembali *username* dan *password*.



Gambar 4. 22 *Activity Diagram* Melihat Halaman *Dashboard* Admin

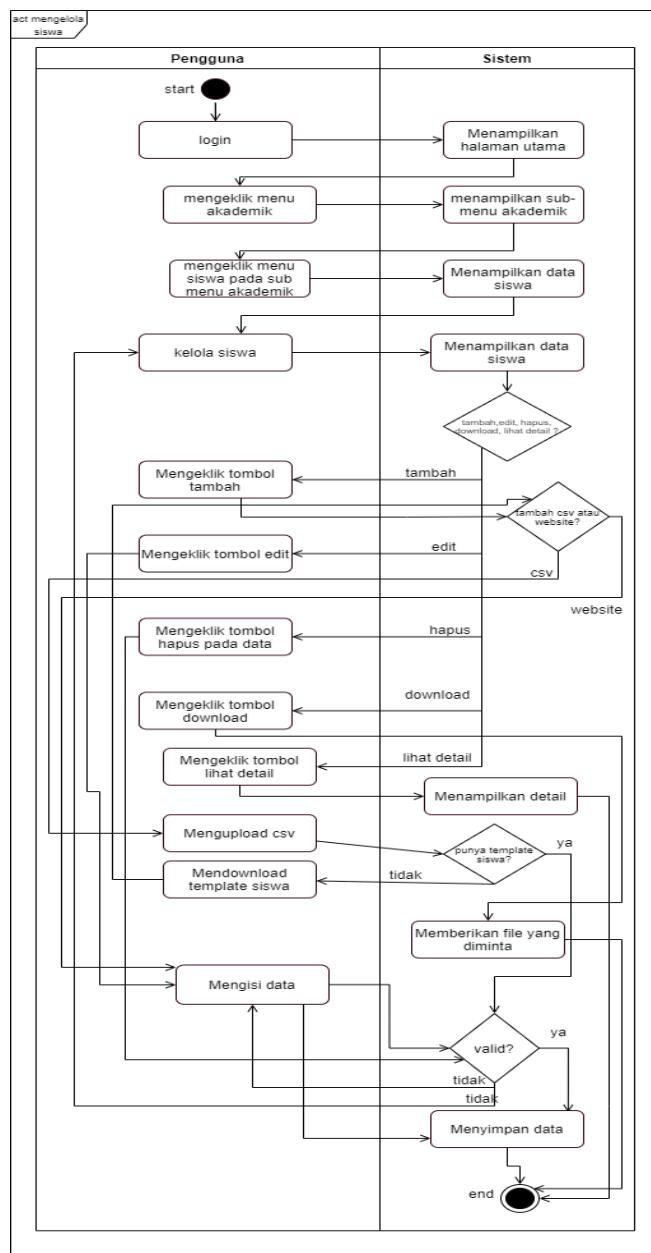
Gambar 4.22 menjelaskan bagaimana *activity* suatu *diagram* dari sebuah *Use Case* melihat menu utama atau dashboard oleh aktor admin. Admin dapat membuka website dan melakukan proses *login*, lalu setelah itu pengguna dapat membuka halaman utama.



Gambar 4. 23 *Activity Diagram* Mengelola Data Kelas Admin

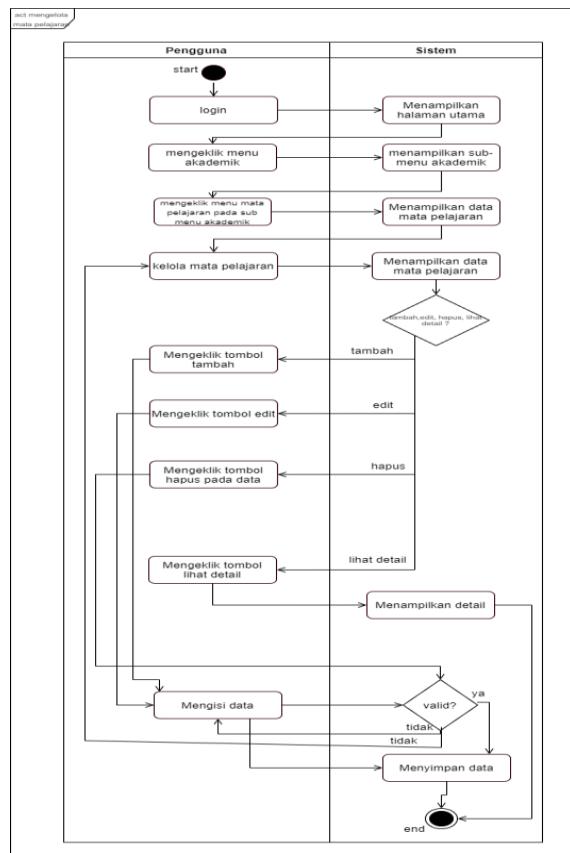
Gambar 4.23 menunjukkan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* mengelola data kelas oleh aktor admin. Admin dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, admin menekan tombol akademik. Setelah itu admin menekan tombol kelas pada sub menu akademik, kemudian admin diarahkan kehalaman kelola kelas. Pada halaman kelola kelas admin dapat menambah, mengedit, menghapus data kelas. Untuk bagian menambah, setelah kelas dikelola sistem akan mengkonfirmasi data yang telah dikelola. Setelah berhasil menambah kelas, sistem dapat menyimpan hasil dari pengelolaan data yang dilakukan. Sementara itu untuk mengedit, setelah kelas

dikelola sistem akan menyimpan hasil dari data kelas. Serta untuk mengkonfirmasi data yang akan dihapus. Ketika tidak dihapus maka akan kembali ke halaman kelola kelas. Namun ketika iya, data kelas dapat menyimpan hasil dari pengelolaan yang dilakukan.



Gambar 4. 24 *Activity Diagram Mengelola Data Siswa Admin*

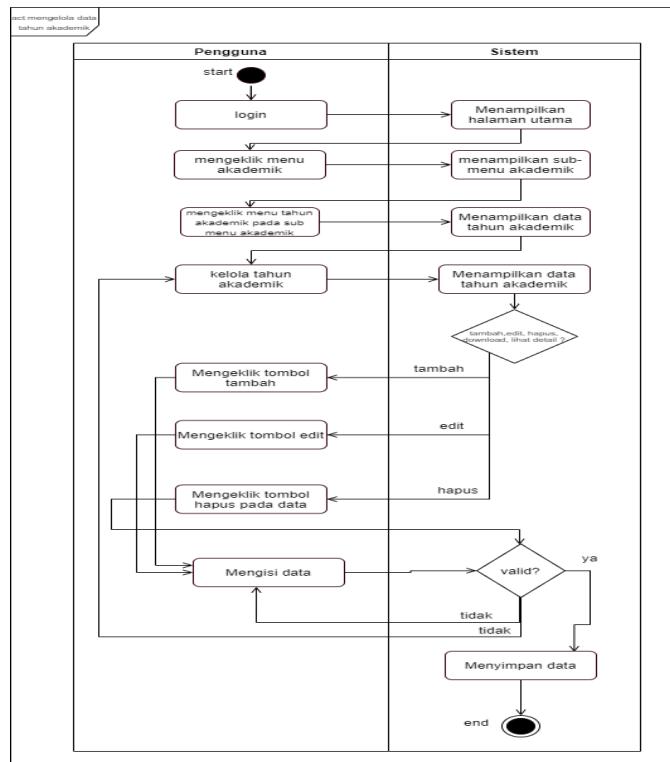
Gambar 4.24 menunjukan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* mengelola data siswa oleh aktor admin. Admin dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, admin menekan tombol akademik. Setelah itu admin menekan tombol siswa pada sub menu akademik, kemudian admin diarahkan kehalaman kelola siswa. Pada halaman kelola siswa admin dapat menambah, mengedit, menghapus, melihat detail serta mendownload data siswa. Untuk bagian menambah, admin dapat menambah dengan *csv* ataupun melalui website. Jika melalui website admin bisa mengisi data. Jika sudah mengisi data sistem akan mengkonfirmasi, jika data benar maka data siswa akan disimpan, namun jika salah maka admin akan mengisi data kembali. Sementara itu dengan *upload csv* yaitu admin dapat upload *csv*, sistem mempunyai sebuah *template csv* siswa. Jika admin belum mempunyai *template* siswa maka admin dapat mendownload *template* siswa terlebih dahulu, kemudian mengisi data yang ada pada *csv* lalu *upload* kembali selanjutnya sistem akan mengkonfirmasi apakah file *csv* benar atau tidak. Setelah *csv* benar maka data siswa yang ada di *csv* akan disimpan. Sementara itu untuk mengedit setelah siswa dikelola sistem akan menyimpan hasil dari data siswa. Serta untuk mengkonfirmasi data yang akan dihapus. Ketika tidak dihapus maka akan kembali ke halaman kelola siswa. Namun ketika iya, data siswa dapat menyimpan hasil dari pengelolaan yang dilakukan. Kemudian untuk lihat detail siswa sistem akan menampilkan detail siswa yang telah dimasukan. Selanjutnya untuk mendownload data siswa sistem akan memberikan *file* yang diminta.



Gambar 4. 25 *Activity Diagram* Mengelola Data Mata Pelajaran Admin

Gambar 4.25 menunjukkan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* mengelola data mata pelajaran oleh aktor admin. Admin dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, admin menekan tombol akademik. Setelah itu admin menekan tombol mata pelajaran pada sub menu akademik, kemudian admin diarahkan kehalaman kelola mata pelajaran. Pada halaman kelola mata pelajaran admin dapat menambah, mengedit, menghapus serta lihat detail data mata pelajaran. Untuk bagian menambah, setelah mata pelajaran dikelola sistem akan mengkonfirmasi data yang telah dikelola. Setelah berhasil menambah mata pelajaran, sistem dapat menyimpan hasil dari pengelolaan data yang dilakukan. Sementara itu untuk mengedit, setelah mata pelajaran dikelola sistem akan menyimpan hasil dari data mata pelajaran. Serta untuk mengkonfirmasi data yang akan dihapus. Ketika tidak dihapus maka akan kembali ke halaman kelola

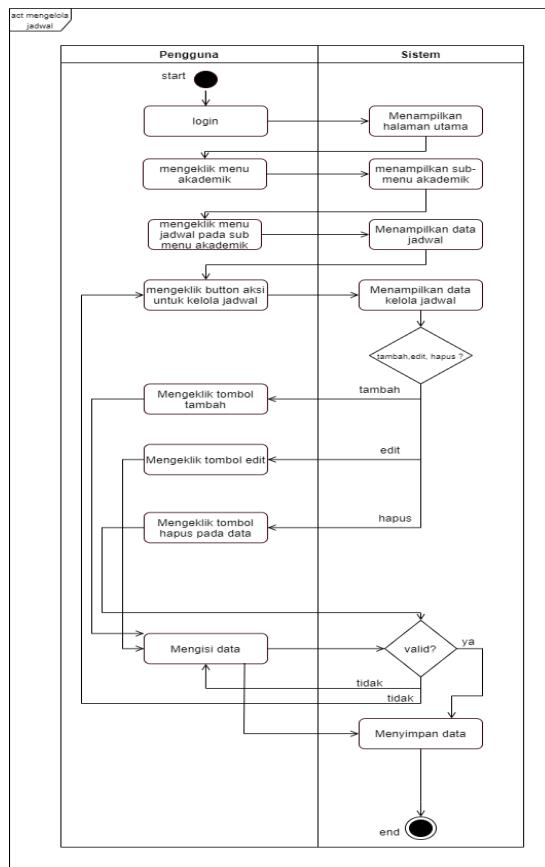
mata pelajaran. Namun ketika iya, data mata pelajaran dapat menyimpan hasil dari pengelolaan yang dilakukan. Kemudian untuk lihat detail mata pelajaran sistem akan menampilkan detail siswa yang telah dimasukan.



Gambar 4. 26 *Activity Diagram* Mengelola Data Tahun Akademik Admin

Gambar 4.26 menunjukkan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* mengelola data tahun akademik oleh aktor admin. Admin dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, admin menekan tombol akademik. Setelah itu admin menekan tombol tahun akademik pada sub menu akademik, kemudian admin diarahkan kehalaman kelola tahun akademik. Pada halaman kelola tahun akademik admin dapat menambah, mengedit, menghapus data tahun akademik. Untuk bagian menambah, setelah tahun akademik dikelola sistem akan mengkonfirmasi data yang telah dikelola. Setelah berhasil menambah tahun akademik, sistem dapat menyimpan hasil dari pengelolaan data yang dilakukan. Sementara itu untuk mengedit, setelah tahun akademik dikelola

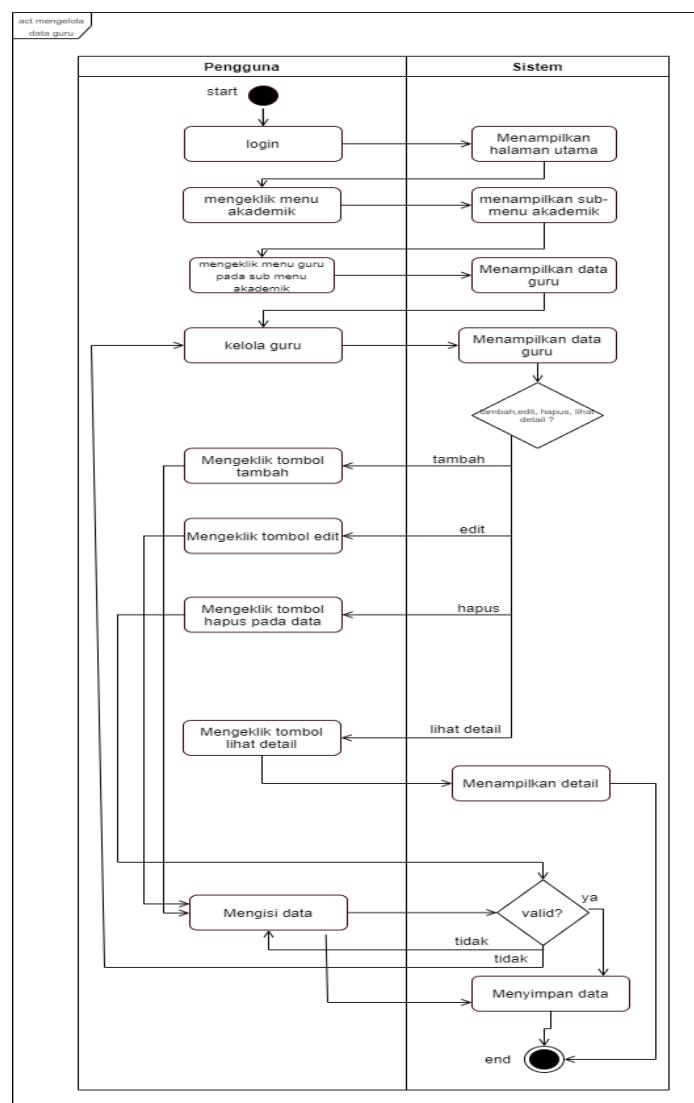
sistem akan menyimpan hasil dari data tahun akademik. Serta untuk mengkonfirmasi data yang akan dihapus. Ketika tidak dihapus maka akan kembali ke halaman kelola tahun akademik. Namun ketika iya, data tahun akademik dapat menyimpan hasil dari pengelolaan yang dilakukan.



Gambar 4. 27 *Activity Diagram* Mengelola Data Jadwal Pelajaran Admin

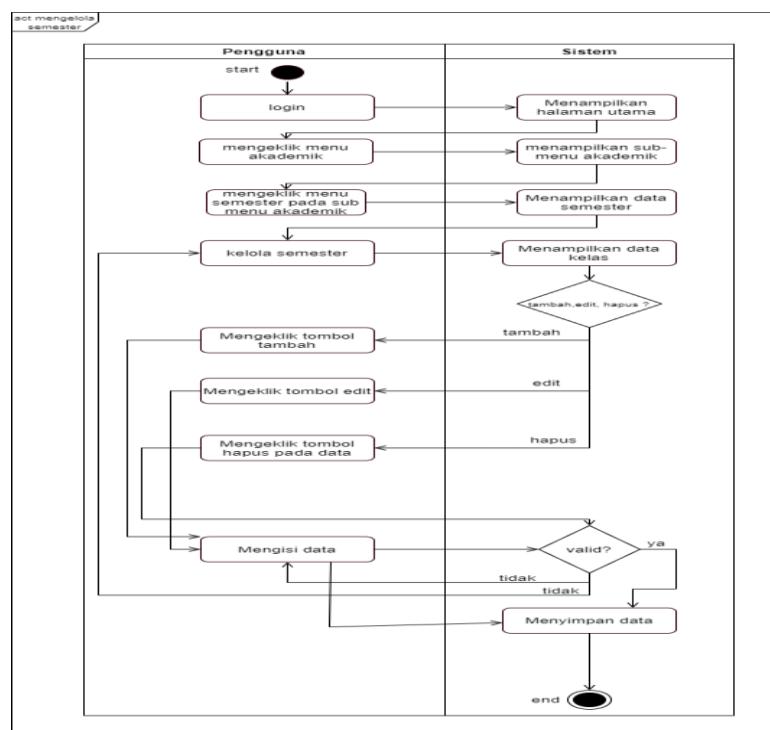
Gambar 4.27 menunjukkan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* mengelola data jadwal pelajaran oleh aktor admin. Admin dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, admin menekan tombol akademik. Setelah itu admin menekan jadwal pelajaran pada sub menu akademik, kemudian admin diarahkan kehalaman jadwal pelajaran. Untuk mengelola lebih detail jadwal menekan tombol aksi. Pada halaman kelola jadwal admin dapat menambah, mengedit, menghapus jadwal pelajaran kelas. Untuk

bagian menambah, setelah jadwal dikelola sistem akan mengkonfirmasi data yang telah dikelola. Setelah berhasil menambah jadwal, sistem dapat menyimpan hasil dari pengelolaan data yang dilakukan. Sementara itu untuk mengedit setelah jadwal dikelola sistem akan menyimpan hasil dari data jadwal. Serta untuk mengkonfirmasi data yang akan dihapus. Ketika tidak dihapus maka akan kembali ke halaman kelola jadwal. Namun ketika iya, jadwal dapat menyimpan hasil dari pengelolaan data yang dilakukan.



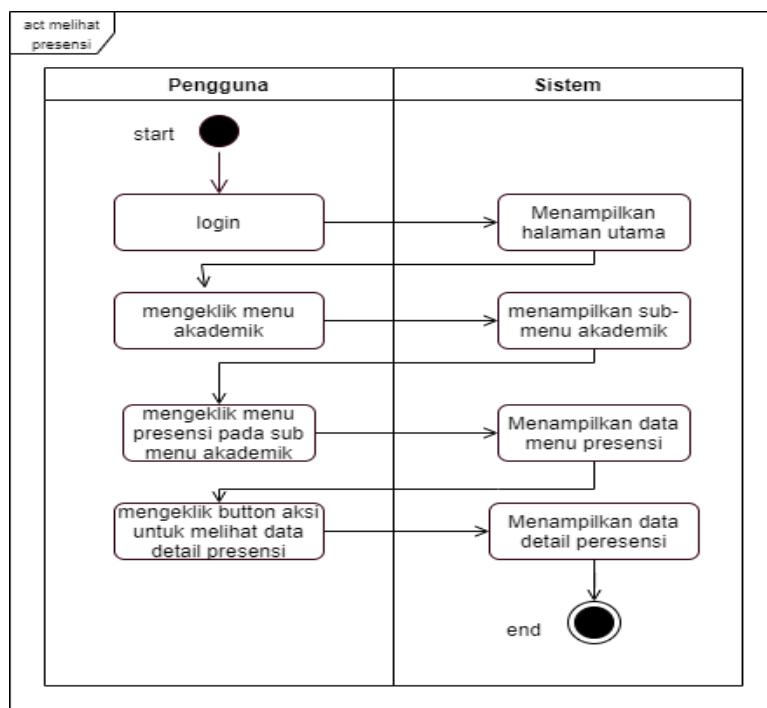
Gambar 4. 28 *Activity Diagram* Mengelola Data Guru Admin

Gambar 4.28 menunjukkan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* mengelola data guru oleh aktor admin. Admin dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, admin menekan tombol akademik. Setelah itu admin menekan tombol guru pada sub menu akademik, kemudian admin diarahkan kehalaman kelola data guru. Pada halaman kelola guru admin dapat menambah, mengedit, menghapus serta lihat detail data guru. Untuk bagian menambah, setelah data guru dikelola sistem akan mengkonfirmasi data yang telah dikelola. Setelah berhasil menambah guru, sistem dapat menyimpan hasil dari pengelolaan data yang dilakukan. Sementara itu untuk mengedit, setelah guru dikelola sistem akan menyimpan hasil dari data guru. Serta untuk mengkonfirmasi data yang akan dihapus. Ketika tidak dihapus maka akan kembali ke halaman kelola guru. Namun ketika iya, data guru dapat menyimpan hasil dari pengelolaan yang dilakukan. Kemudian untuk lihat detail guru sistem akan menampilkan detail guru yang telah dimasukan.



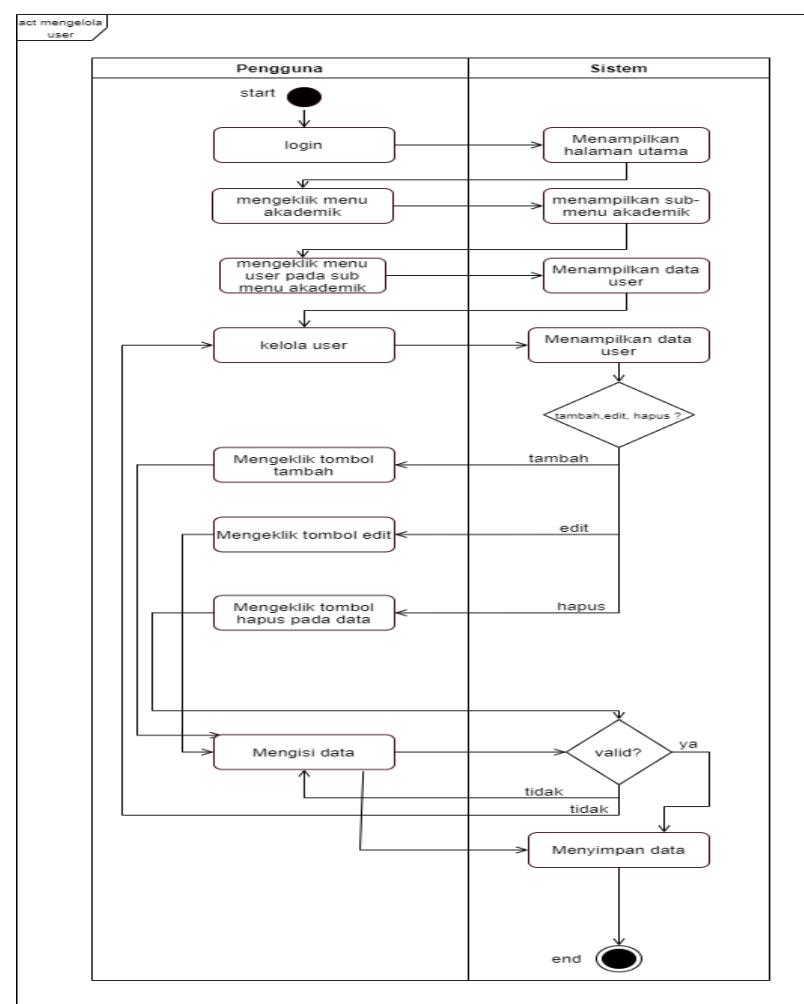
Gambar 4. 29 *Activity Diagram* Mengelola Data Semester Admin

Gambar 4.29 menunjukkan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* mengelola data semester oleh aktor admin. Admin dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, admin menekan tombol akademik. Setelah itu admin menekan tombol semester pada sub menu akademik, kemudian admin diarahkan kehalaman kelola data semester. Pada halaman kelola semester admin dapat menambah, mengedit, menghapus data semester. Untuk bagian menambah, setelah data semester dikelola sistem akan mengkonfirmasi data yang telah dikelola. Setelah berhasil menambah semester, sistem dapat menyimpan hasil dari pengelolaan data yang dilakukan. Sementara itu untuk mengedit, setelah semester dikelola sistem akan menyimpan hasil dari data semester. Serta untuk mengkonfirmasi data yang akan dihapus. Ketika tidak dihapus maka akan kembali ke halaman kelola semester. Namun ketika iya, data semester dapat menyimpan hasil dari pengelolaan yang dilakukan.



Gambar 4. 30 *Activity Diagram* Melihat Data Presensi Admin

Gambar 4. 30 menunjukan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* melihat data presensi oleh aktor admin. Admin dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, admin menekan tombol akademik. Setelah itu admin menekan presensi pada sub menu akademik, kemudian admin diarahkan kehalaman presensi siswa. Untuk melihat lebih detail presensi menekan tombol aksi selanjutnya sistem akan menampilkan data detail presensi siswa.



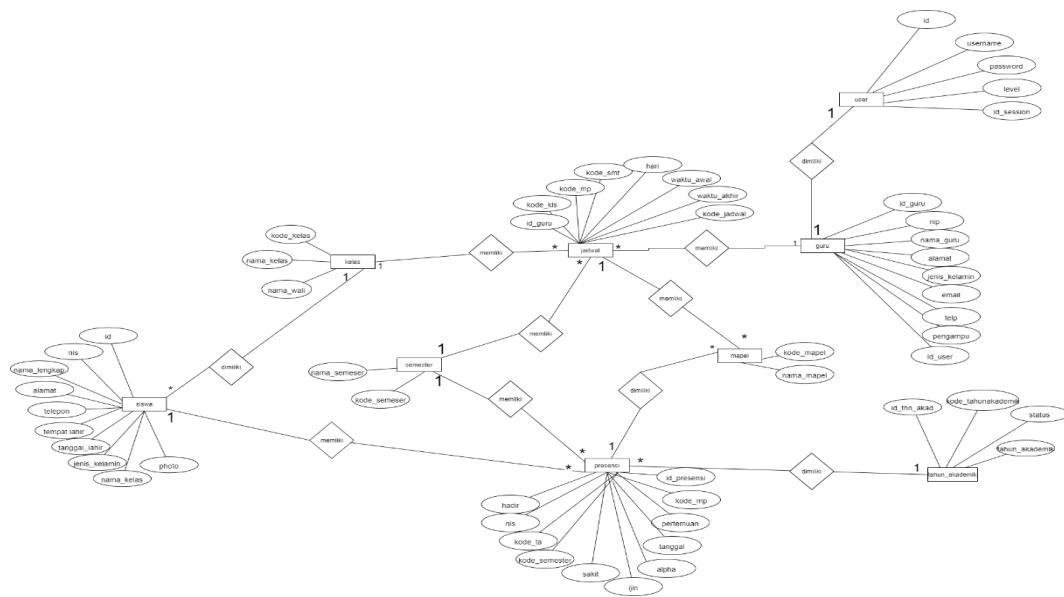
Gambar 4. 31 *Activity Diagram* Mengelola Data User Admin

Gambar 4.31 menunjukkan tentang bagaimana *activity* suatu *diagram* dari sebuah *Use Case* mengelola data *user* oleh aktor admin. Admin dapat melakukan *login*, kemudian setelah berhasil melakukan *login*, admin menekan tombol akademik. Setelah itu admin menekan tombol *user* pada sub menu akademik, kemudian admin diarahkan kehalaman kelola data *user*. Pada halaman kelola *user* admin dapat menambah, mengedit, menghapus data *user*. Untuk bagian menambah, setelah data *user* dikelola sistem akan mengkonfirmasi data yang telah dikelola. Setelah berhasil menambah *user*, sistem dapat menyimpan hasil dari pengelolaan data yang dilakukan. Sementara itu untuk mengedit, setelah *user* dikelola sistem akan menyimpan hasil dari data *user*. Serta untuk mengkonfirmasi data yang akan dihapus. Ketika tidak dihapus maka akan kembali ke halaman kelola *user*. Namun ketika iya, data *user* dapat menyimpan hasil dari pengelolaan yang dilakukan.

## 4.2.2 Desain

Pada tahap desain, membahas mengenai rancangan desain basisdata dengan menggunakan *ERD* (*Entity Relationship Diagram*).

### 4.2.2.1 ERD (*Entity Relationship Diagram*)



Gambar 4. 32 SIAwarak ERD (*Entity Relationship Diagram*)

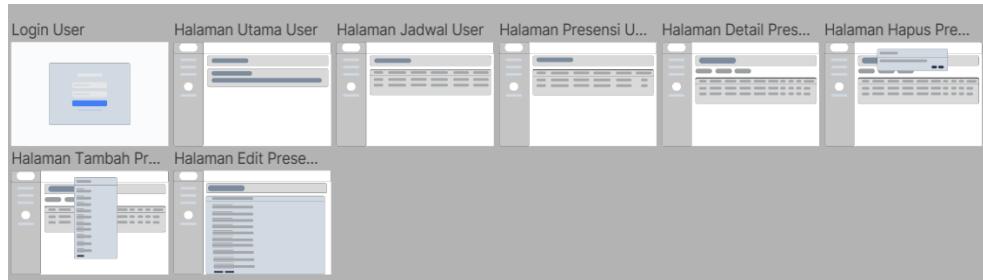
Pada gambar 4. 32, merupakan *Entity Relationship Diagram* dari sistem informasi akademik SIAwarak. Pada gambar tersebut menunjukkan hubungan antar tabel yang erdapat 9 tabel/entity dengan aksi dan atributnya masing masing.

#### 4.2.3 Implementasi

Setelah perencanaan selesai langkah berikutnya adalah implementasi atau aksi, ada beberapa *case* yang pertama peneliti membuat *Low Fidelity* dan setelah itu di aksikan ke dalam codingan sehingga menjadi website yang dapat dijalankan.

##### 4.2.3.1 Low Fidelity

*Low fidelity* merupakan design tingkat presisi yang rendah, yang digunakan hanya untuk menentukan tata letak.



Gambar 4. 33 *Low Fidelity* Sistem Informasi Akademik SIAwarak *User*

Pada gambar 4. 33 merupakan *low fidelity* dari pengguna yang meliputi *login*, halaman utama, halaman jadwal, halaman kelola presensi meliputi halaman detail presensi, halaman tambah presensi, halaman hapus presensi, halaman ubah presensi.

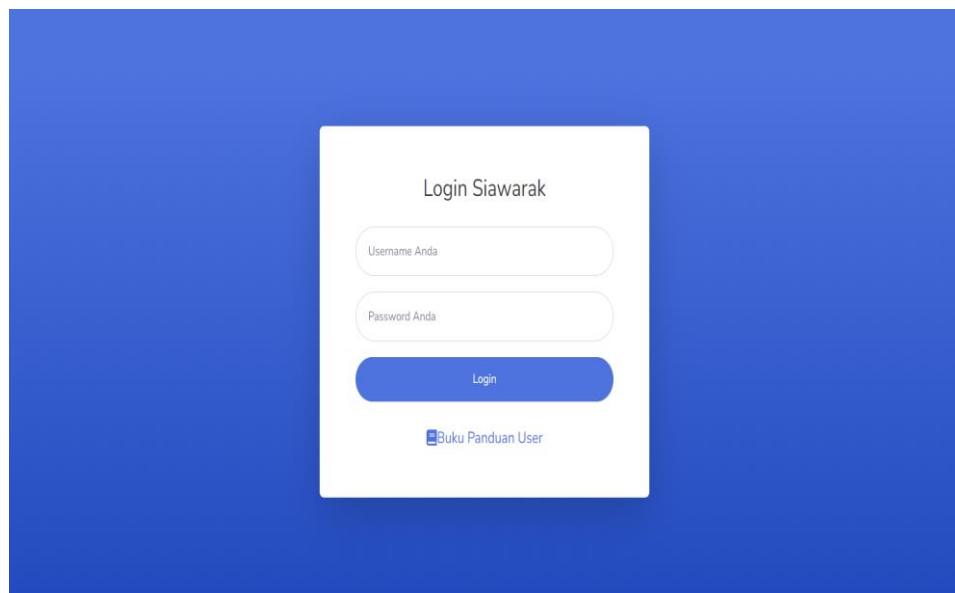


Gambar 4. 34 *Low Fidelity* Sistem Informasi Akademik SIAwarak *Admin*

Pada gambar 4. 34 merupakan *low fidelity* dari admin yang meliputi *login*, halaman utama, halaman kelola jadwal, halaman kelola kelas, halaman kelola guru, halaman kelola *user*, halaman kelola siswa, halaman kelola mata pelajaran, halaman kelola tahun akademik, halaman kelola semester, halaman presensi.

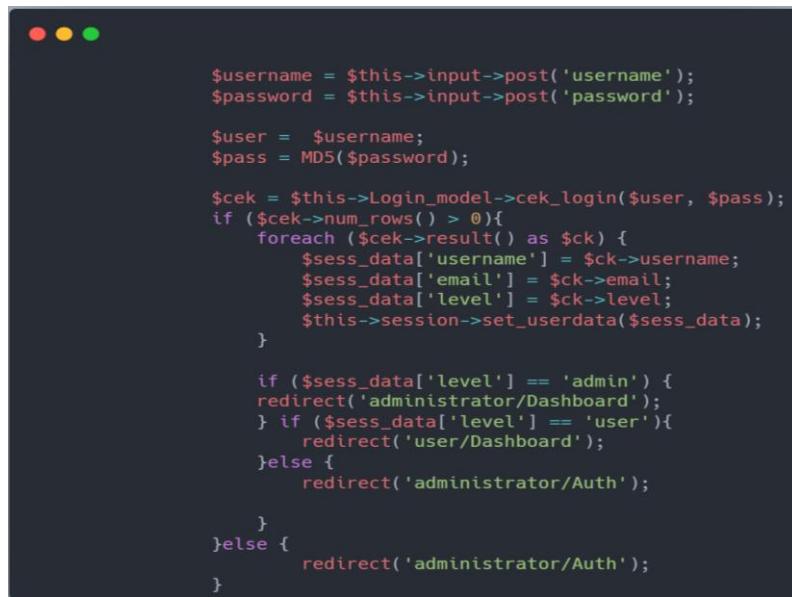
#### 4.2.3.2 *CodeIgniter & PHP*

*PHP (PHP: Hypertext Preprocessor)* merupakan *syntax* atau suatu bahasa pemrograman yang dapat digunakan sebagai mengembangkan aplikasi web, sedangkan *CodeIgniter* ialah sebuah *framework* dari *PHP* yang ditujukan untuk membuat aplikasi atau web dengan lebih mudah. Berikut tampilan web yang sudah di implementasikan.



Gambar 4. 35 Halaman *Login* Admin dan *User*

Pada gambar 4. 35 merupakan halaman *login* baik admin maupun *user*. Selanjutnya terdapat *code* untuk *login* yang dilihat pada gambar 4.36.



```

$username = $this->input->post('username');
$password = $this->input->post('password');

$user = $username;
$pass = MD5($password);

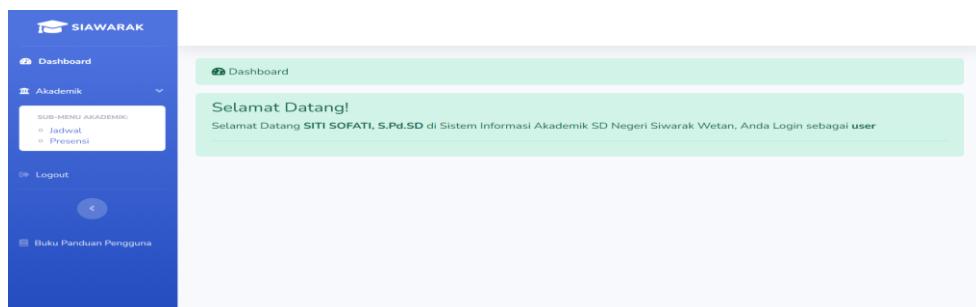
$cek = $this->Login_model->cek_login($user, $pass);
if ($cek->num_rows() > 0){
    foreach ($cek->result() as $ck) {
        $sess_data['username'] = $ck->username;
        $sess_data['email'] = $ck->email;
        $sess_data['level'] = $ck->level;
        $this->session->set_userdata($sess_data);
    }

    if ($sess_data['level'] == 'admin') {
        redirect('administrator/Dashboard');
    } if ($sess_data['level'] == 'user'){
        redirect('user/Dashboard');
    }else {
        redirect('administrator/Auth');
    }
} else {
    redirect('administrator/Auth');
}

```

Gambar 4. 36 Potongan *Code Login*

Pada gambar 4. 36 merupakan potongan *code login*. Dimulai dari mengambil *value username* dan *password* kemudian mengubah *password* menjadi *MD5*. Selanjutnya melakukan pengecekan kedalam *Login\_model*. Jika proses pengecekan berhasil maka sistem akan menyimpan data user kedalam sesi dan kemudian pengguna diarahkan ke *dashboard* sesuai role masing masing. Misal pengguna tersebut memiliki level admin maka akan menampilkan *dashboard admin*, namun jika level *user* maka menampilkan *dashboard user*. Akan tetapi jika pengecekan *login* gagal maka akan dilakukan proses memasukan *username* dan *password* Kembali.



Gambar 4. 37 Halaman Utama *User*

Pada gambar 4. 37 merupakan halaman utama *user*, pada halaman utama *user* menampilkan deskripsi. Selain itu pada halaman utama *user* disini merupakan halaman awal ketika telah melakukan *login* sebagai *user*. *User* memiliki dua menu yaitu menu jadwal serta presensi. Menu presensi disini merupakan menu utama dari sistem informasi akademik yang dimiliki oleh *user*.

NIS	Nama	Kelas	Aksi
48392	Andra eey	Kelas 1	
12456	pir	Kelas 1	
12234	Alia	Kelas 1	

Gambar 4. 38 Halaman Presensi *User*

Pada gambar 4. 38 merupakan halaman presensi *user*, pada halaman presensi *user* menampilkan presensi dari siswa. Jika *user* ingin mengelola presensi maka masuk ke halaman detail presensi dengan menekan tombol lihat detail yang berada pada aksi. Berikut *code* yang ada pada presensi.

```
public function presensi()
{
    $nama_guru = $this->session->userdata('username');
    $data['siswa'] = $this->Siswa_model->listSiswaWhrNamaGuru($nama_guru);

    $this->load->view('admin/header');
    $this->load->view('users/sidebar');
    $this->load->view('user/presensi', $data);
    $this->load->view('admin/footer');
}
```

Gambar 4. 39 Potongan *Code Presensi*

Pada gambar 4.39 ditulis dalam bahasa pemrograman *PHP* dan menggunakan *framework CodeIgniter*. Ini adalah sebuah fungsi yang disebut "presensi" yang terdiri dari beberapa bagian. Pertama, fungsi ini mengambil nama pengguna guru dari sesi yang sedang berjalan. Selanjutnya mendapatkan data siswa yang diajar oleh guru yang sedang *login*. Kemudian menampilkan *view*. Dengan demikian maka fungsi ini bertujuan untuk menampilkan daftar siswa dari seorang guru untuk melakukan presensi pada tampilan "presensi" pada halaman *website*.



```

public function listSiswaWhrNamaGuru($kode)
{
    $this->db->select('siswa.*', 'kelas.nama_kelas as nama');
    $this->db->from('siswa');
    $this->db->join('kelas', 'siswa.nama_kelas=kelas.nama_kelas');
    $this->db->join('guru', 'guru.nama_guru=kelas.nama_wali');
    $this->db->where('guru.nama_guru', $kode);
    $query = $this->db->get();
    return $query->result();
}

```

Gambar 4. 40 Potongan *Code Model* Siswa

Selanjutnya dengan *model* siswa yang ada pada gambar 4.40 dengan sebuah fungsi yang disebut "listSiswaWhrNamaGuru" yang terdiri dari beberapa bagian. Pertama fungsi ini menerima satu parameter, yaitu kode guru. Selanjutnya menentukan kolom yang akan dipilih dari tabel siswa dan tabel kelas menggunakan metode "*select*". Kemudian dijoinkan dengan menggabungkan tabel siswa dan tabel kelas dengan kolom "nama\_kelas", serta menggabungkan tabel guru dengan kolom "nama\_guru" pada tabel kelas. Setelah itu memfilter hasil *join* berdasarkan nama guru menggunakan metode "*where*". Lalu menjalankan *query* dan mengembalikan hasilnya sebagai *array* objek menggunakan metode "*result*". Dengan demikian maka fungsi ini bertujuan untuk mengambil daftar siswa beserta nama kelas mereka dari *database* yang diasuh oleh seorang guru yang sesuai dengan kode guru yang diberikan.

No	Matapelajaran	Tahun Akademik	Tanggal	Pertemuan	Hadir	Alpha	Ijin	Sakit	Aksi
1	Matematika	2022/2023	2023-03-28	1	0	0	1	0	
2	Geologi	2022/2023	2023-03-30	1	0	1	0	0	
Total					0	1	1	0	

Gambar 4. 41 Halaman Detail Presensi *User*

Pada gambar 4. 41 merupakan halaman detail presensi *user*, pada halaman detail presensi *user* ini terdapat tambah presensi, ubah data presensi, serta hapus data presensi, menampilkan data presensi dari siswa. Berikut ini *code* yang ada pada detail presensi siswa. Serta kode untuk mengunduh data presensi dapat dilihat pada gambar 4. 51.

```

public function detailpresensi($nis)
{
    $tahun_akademik = $this->Tahunakademik_model->get_tahun_aktif();

    $data['nis'] = $nis;
    $data['presensi'] = $this->Presensi_model->get_presensi_by_nis($nis, $tahun_akademik);
    $data['mapel'] = $this->Mapel_model->listMapel();
    $data['semester'] = $this->Semester_model->listSemester();
    $data['tahun_akademik'] = $this->Tahunakademik_model->listTahun_akademik();

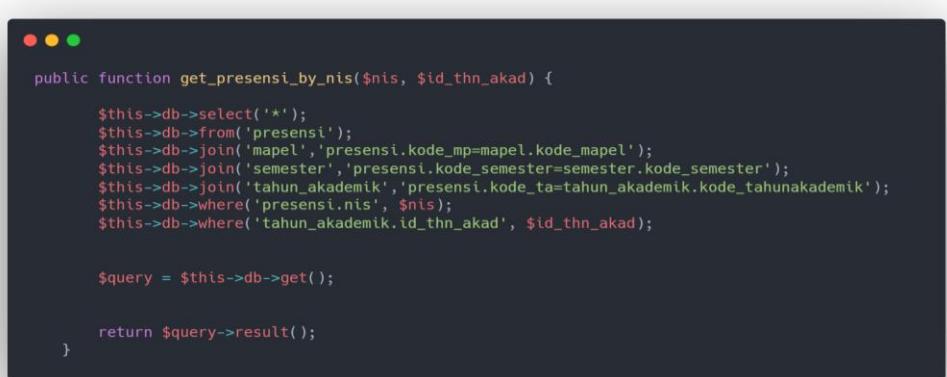
    $this->load->view('admin/header');
    $this->load->view('users/sidebar');
    $this->load->view('user/detail_presensi', $data);
    $this->load->view('admin/footer');
}

```

Gambar 4. 42 Potongan *Code* Detail Presensi

Pada gambar 4. 42 merupakan potongan *code controller* dengan fungsi "detailpresensi". Dimana pada *code* tersebut mengambil data presensi berdasarkan NIS, tahun akademik yang aktif, data daftar mata pelajaran, data daftar semester,

data daftar tahun akademik. Selanjutnya akan menampilkan tampilan detail presensi dengan menggunakan data yang telah diambil. Dengan demikian maka fungsi ini bertujuan untuk menampilkan detail presensi siswa, termasuk daftar mata pelajaran, semester, dan tahun akademik, pada tahun akademik aktif. Kode ini digunakan dalam halaman *web* yang menampilkan detail presensi siswa untuk seorang guru.



```

public function get_presensi_by_nis($nis, $id_thn_akad) {
    $this->db->select('*');
    $this->db->from('presensi');
    $this->db->join('mapel','presensi.kode_mp=mapel.kode_mapel');
    $this->db->join('semester','presensi.kode_semester=semester.kode_semester');
    $this->db->join('tahun_akademik','presensi.kode_ta=tahun_akademik.kode_tahunakademik');
    $this->db->where('presensi.nis', $nis);
    $this->db->where('tahun_akademik.id_thn_akad', $id_thn_akad);

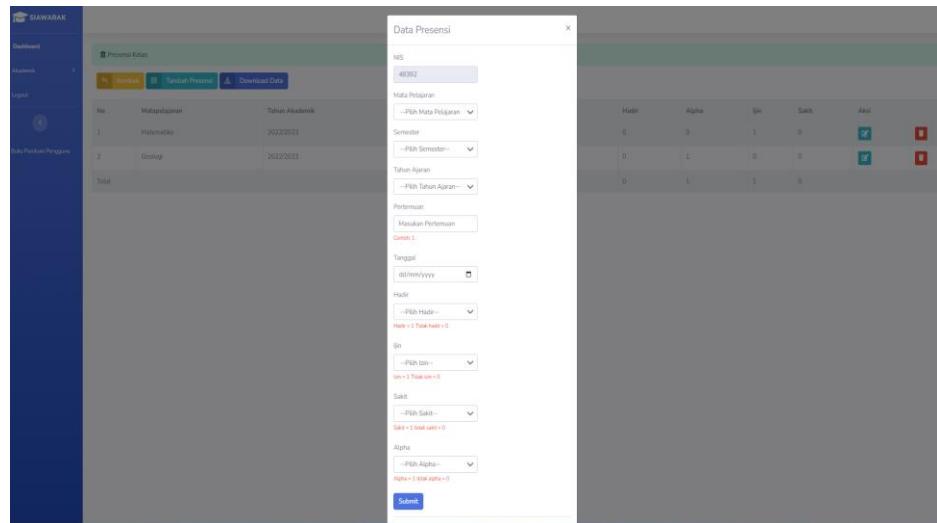
    $query = $this->db->get();

    return $query->result();
}

```

Gambar 4. 43 Potongan *Code Model* Presensi

Pada gambar 4. 43 merupakan potongan kode *model* presensi yang diberi nama sebuah fungsi "get\_presensi\_by\_nis" yang terdiri dari beberapa bagian. Pada potongan kode ini menerima dua parameter yaitu NIS (Nomor Induk Siswa) dan ID tahun akademik, serta menggunakan *model* untuk mengambil data presensi siswa berdasarkan NIS dan ID tahun akademik. Kemudian dijoinkan dengan tabel mata pelajaran, semester, dan tahun akademik. Kemudian menambahkan dua klausa "*where*" untuk memfilter data berdasarkan NIS siswa dan ID tahun akademik. Lalu menjalankan *query* dan mengembalikan hasilnya sebagai *array* objek menggunakan metode "*result*". Dengan demikian maka fungsi ini bertujuan untuk mengambil data presensi siswa berdasarkan NIS dan ID tahun akademik. Kode ini digunakan dalam *model* untuk mengambil data presensi siswa dari *database* dan mengirimkannya ke tampilan dalam sebuah halaman *web* yang menampilkan detail presensi siswa.



Gambar 4. 44 Halaman Tambah Presensi *User*

Pada gambar 4. 44 merupakan halaman tambah presensi *user*, pada halaman tambah presensi *user* ini terdapat *form* tambah presensi yang akan dimasukan, selanjutnya ketika data telah dimasukan lalu tekan *submit*. Dengan kode tambah jadwal pada gambar 4. 45.

```

public function savepresensi()
{
    $kode_mp = $this->input->post('kode_mp');
    $pertemuan = $this->input->post('pertemuan');
    $stanggal = $this->input->post('tanggal');
    $alpha = $this->input->post('alpha');
    $ijin = $this->input->post('ijin');
    $sakit = $this->input->post('sakit');
    $nts = $this->input->post('nis');
    $kode_semester = $this->input->post('kode_semester');
    $kode_ta = $this->input->post('kode_ta');
    $hadir = $this->input->post('hadir');

    if (!is_numeric($pertemuan)) {
        $this->session->set_flashdata('pesan', '<div class="alert alert-danger alert-dismissible fade show" role="alert"> Pertemuan harus berupa angka! <button type="button" class="close" data-dismiss="alert" aria-label="Close"> <span aria-hidden="true">&times;</span> </button> </div>');
        redirect('user/presensi/detailpresensi/' . $nts);
    } else {
        $data = array(
            'kode_mp' => $kode_mp,
            'pertemuan' => $pertemuan,
            'alpha' => $alpha,
            'ijin' => $ijin,
            'tanggal' => $stanggal,
            'sakit' => $sakit,
            'nts' => $nts,
            'kode_semester' => $kode_semester,
            'kode_ta' => $kode_ta,
            'hadir' => $hadir,
        );
        $this->Presensi_model->insertPresensi($data, 'presensi');
        $this->session->set_flashdata('pesan', '<div class="alert alert-success alert-dismissible fade show" role="alert"> Data Presensi Berhasil Ditambahkan! <button type="button" class="close" data-dismiss="alert" aria-label="Close"> <span aria-hidden="true">&times;</span> </button> </div>');
        redirect('user/presensi/detailpresensi/' . $data['nis']);
    }
}

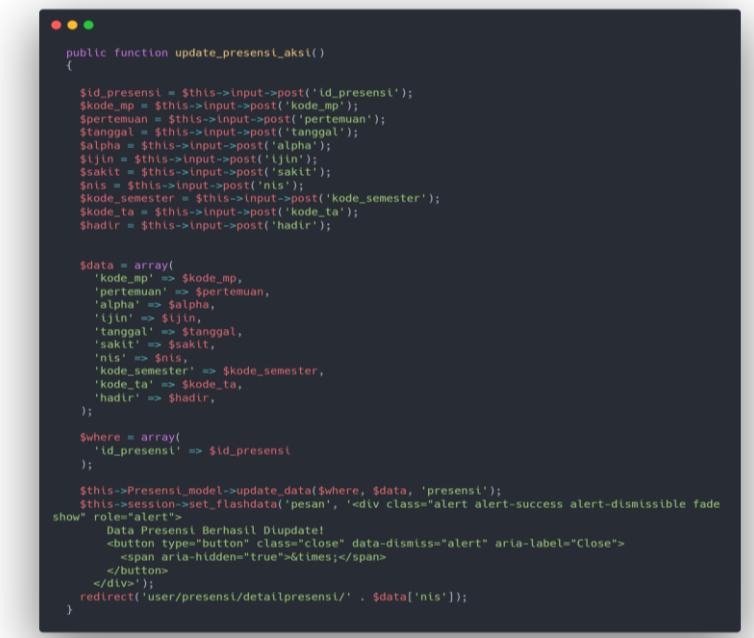
```

Gambar 4. 45 Potongan Code Tambah Presensi

Pada gambar 4. 45 merupakan *controller* dari presensi yang diberi nama sebuah fungsi "savepresensi" yang terdiri dari beberapa bagian. Pada kode ini fungsi ini akan dipanggil ketika tombol "Submit" pada form presensi ditekan. Pertama tama mengambil nilai melalui metode *POST* dari form inputan. Kemudian jika pertemuan bukan angka maka akan ada *alert*. Selanjutnya jika pertemuan adalah angka maka akan menyimpan data presensi tersebut kedalam *array*, serta memasukan data presensi ke dalam tabel presensi di *database* dengan memanggil fungsi “*insertPresensi()*” dari *model* “*Presensi\_model*”. Kemudian mengeset pesan berhasil ditambahkan ke dalam *session*, lalu melakukan *redirect* ke halaman detail presensi dengan parameter “\$nis”.

Gambar 4. 46 Halaman Mengubah Data Presensi *User*

Pada gambar 4. 46 merupakan halaman ubah presensi *user*, pada halaman ubah presensi *user* ini terdapat *form* ubah presensi yang akan dimasukan, selanjutnya ketika data telah dimasukan lalu tekan simpan. Dengan kode ubah data presensi pada gambar 4. 47.



```

public function update_presensi_aksi()
{
    $id_presensi = $this->input->post('id_presensi');
    $kode_mp = $this->input->post('kode_mp');
    $pertemuan = $this->input->post('pertemuan');
    $stanggal = $this->input->post('stanggal');
    $alpha = $this->input->post('alpha');
    $ijin = $this->input->post('ijin');
    $sakit = $this->input->post('sakit');
    $nis = $this->input->post('nis');
    $kode_semester = $this->input->post('kode_semester');
    $kode_ta = $this->input->post('kode_ta');
    $hadir = $this->input->post('hadir');

    $data = array(
        'kode_mp' => $kode_mp,
        'pertemuan' => $pertemuan,
        'alpha' => $alpha,
        'ijin' => $ijin,
        'stanggal' => $stanggal,
        'sakit' => $sakit,
        'nis' => $nis,
        'kode_semester' => $kode_semester,
        'kode_ta' => $kode_ta,
        'hadir' => $hadir,
    );

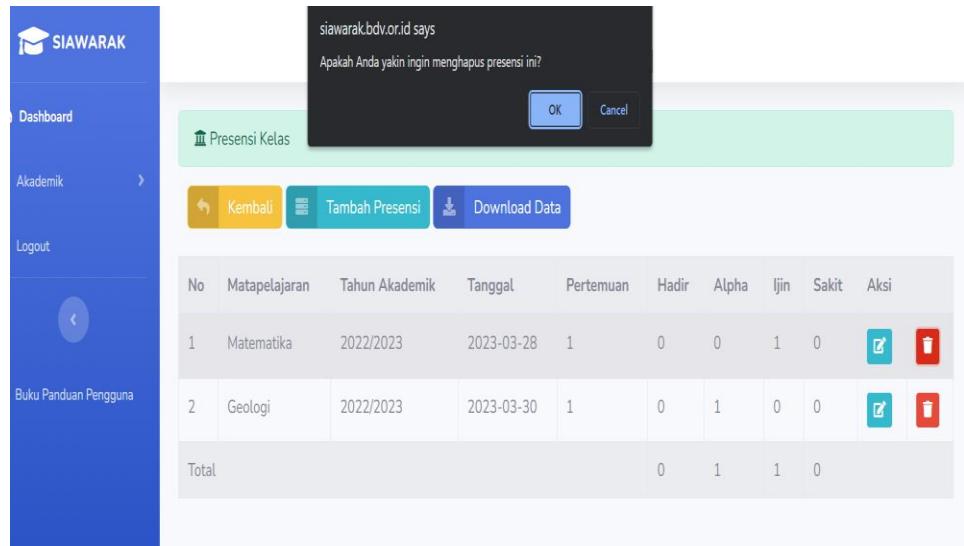
    $where = array(
        'id_presensi' => $id_presensi
    );

    $this->Presensi_model->update_data($where, $data, 'presensi');
    $this->session->set_flashdata('pesan', '<div class="alert alert-success alert-dismissible fade show" role="alert">
        Data Presensi Berhasil Diupdate!
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">&amptimes</span>
        </button>
    </div>');
    redirect('user/presensi/detailpresensi/' . $data['nis']);
}

```

Gambar 4. 47 Potongan Code Ubah Data Presensi

Gambar 4. 47 tersebut adalah sebuah fungsi yang digunakan untuk meng-update data presensi di database. Fungsi ini akan dipanggil ketika tombol "*Update*" pada form presensi ditekan. Pada awal fungsi, mengambil nilai melalui metode *POST* dari form inputan. Kemudian, nilai-nilai presensi tersebut disimpan ke dalam *array* *\$data*. Selanjutnya, dilakukan pengaturan kondisi dengan menggunakan *where clause* untuk memastikan data yang di ubah tepat sesuai dengan id presensi yang diberikan. Kemudian data yang baru diperoleh disimpan ke dalam tabel 'presensi' melalui method *update\_data()* dari model *Presensi\_model*. Jika proses update berhasil dilakukan, akan muncul pesan sukses dan pengguna akan dialihkan ke halaman detail presensi dengan parameter *\$nis*. Pesan tersebut disimpan dalam *session* berhasil *update* data.



Gambar 4. 48 Halaman Hapus Data Presensi *User*

Pada gambar 4. 48 merupakan halaman hapus presensi *user*, pada halaman jadwal terdapat notifikasi hapus presensi *user* ini. Lalu tekan ok untuk menghapus. Dengan kode hapus presensi pada gambar 4. 49.

```

public function hapuspresensi($id, $nis)
{
    $where = array('id_presensi' => $id);
    $this->Presensi_model->hapus_data($where, 'presensi');
    $this->session->set_flashdata('pesan', '<div class="alert alert-danger alert-dismissible fade show" role="alert">
        Data Presensi Berhasil Dihapus!
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>');
}

redirect('user/presensi/detailpresensi/' . $nis);
}

```

Gambar 4. 49 Potongan Code Hapus Data Presensi

Gambar 4.49 adalah sebuah fungsi yang digunakan untuk menghapus data presensi berdasarkan id presensi yang diterima sebagai parameter pertama dan nilai NIS yang diterima sebagai parameter kedua. Fungsi ini memanggil fungsi `hapus_data()` dari model `Presensi_model` dengan memberikan nilai dengan kondisi `where` yang sesuai dengan id presensi yang akan dihapus. Setelah itu, `session`

*flashdata* digunakan untuk menampilkan pesan berhasil dihapus dan pengguna akan diarahkan kembali ke halaman detail presensi siswa dengan nomor induk *\$nis*. Untuk menampilkan *alert* dalam hapus data presensi yaitu ketika tombol hapus tersebut di klik, maka akan menjalankan sebuah fungsi dengan memanggil *URL* dari method "hapuspresensi" dengan parameter "id\_presensi" dan "nis". Fungsi ini akan menampilkan konfirmasi untuk memastikan apakah pengguna yakin untuk menghapus presensi tersebut. Jika pengguna menekan tombol "OK" pada konfirmasi, maka presensi akan dihapus dari database seperti pada gambar 4. 50



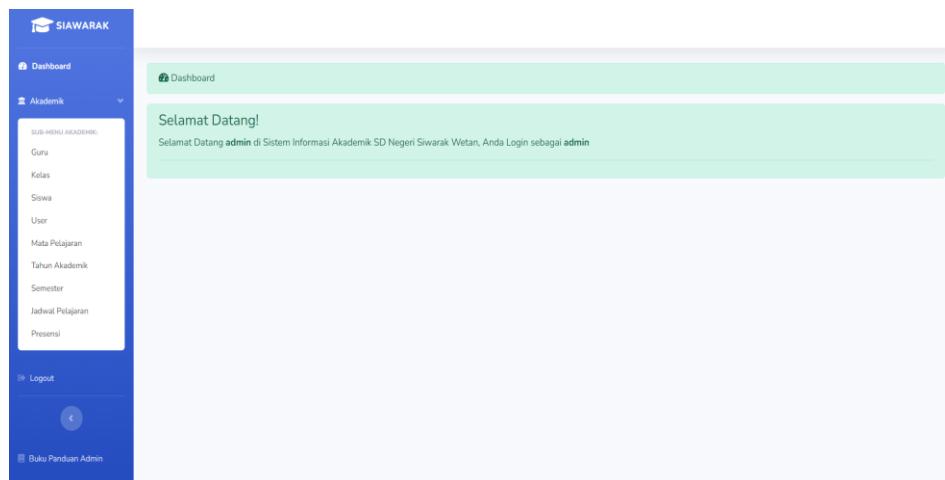
Gambar 4. 50 Potongan *Code Hapus Data Presensi Alert*



Gambar 4. 51 Potongan *Code Unduh Data Presensi*

Pada gambar 4. 51 terdapat kode implementasi dari sebuah fungsi untuk mengunduh data presensi dalam format *CSV*. Pertama-tama, fungsi ini memanggil *model* untuk mendapatkan data presensi dari *database*. Setelah itu, fungsi ini membuat *objek Spreadsheet* dan *Worksheet* baru untuk menulis data ke dalam *file*

CSV. Urutan kolom pada *file CSV* ditentukan oleh variabel *\$columnOrder*, yang berisi nama-nama *field* pada *database*. Kemudian, *header* kolom ditulis pada baris pertama, dan data presensi ditulis pada baris selanjutnya. Setelah itu, *objek writer* dibuat untuk menulis *file CSV*. Selanjutnya, *header* untuk *file download* ditentukan dan *output* dari *writer* diarahkan ke browser. Terakhir, properti dari *objek writer CSV* diatur, seperti penggunaan *BOM* (*Byte Order Mark*) untuk membantu aplikasi membaca karakter khusus pada *file CSV*, penggunaan *delimiter* (tanda pemisah) untuk memisahkan antara kolom dalam *file CSV*, penggunaan *enclosure* (tanda pembuka dan penutup) untuk melindungi nilai kolom yang mengandung *delimiter*, dan penggunaan *line ending* (tanda akhir baris) untuk menandakan akhir baris dalam *file CSV*. Sehingga *output* dari *objek writer CSV* disimpan di *php://output* sehingga dapat dikirimkan ke browser untuk diunduh.



Gambar 4. 52 Halaman Utama Admin

Pada gambar 4. 52 merupakan halaman utama admin, pada halaman utama admin menampilkan deskripsi. Halaman utama admin merupakan halaman awal ketika telah melakukan login sebagai admin. Terdapat beberapa menu yang berada pada halaman utama admin yaitu guru, kelas, siswa, *user*, mata pelajaran, tahun akademik, semester, jadwal pelajaran dan presensi. Untuk admin memiliki fitur

utama yaitu jadwal pelajaran untuk mengelola jadwal yang dipakai pada saat mengajar.

NO	KODE KELAS	NAMA KELAS	NAMA WALI	AKSI
1	K1	Kelas 1	SITI SOFATI, S.Pd.SD	
2	K2	Kelas 2	ARIF ROHMAN HAKIM, S.Pd	
3	K3	Kelas 3	AGUS LATIF AZIS FAUZI	
4	K4	Kelas 4	SRI SUHARYATI, S.Pd	
5	K5	Kelas 5	NOWO SUHENDRO,S.Pd	
6	K6	Kelas 6	DVVI SETYANINGSIH, S.Pd	

Gambar 4. 53 Halaman Jadwal Pelajaran Admin

Pada gambar 4. 53 merupakan halaman jadwal pelajaran admin., Pada halaman jadwal pelajaran menampilkan data jadwal pelajaran.

```
public function index()
{
    $data['kelas'] = $this->Kelas_model->tampil_data()->result();
    $this->load->view('admin/header');
    $this->load->view('admin/sidebar');
    $this->load->view('administrator/jadwal', $data);
    $this->load->view('admin/footer');
}
```

Gambar 4. 54 Potongan Code Jadwal

Gambar 4. 54 merupakan bagian dari sebuah *controller* di aplikasi *web* yang digunakan untuk menampilkan halaman jadwal pada tampilan admin. Fungsi ini digunakan untuk menampilkan halaman jadwal yang terdiri dari data kelas yang diambil dari *database* melalui *model Kelas\_model*. Data kelas tersebut akan dimasukkan ke dalam *array \$data* dengan *key 'kelas'* dan akan digunakan pada *view* jadwal. Kemudian, fungsi *load view* digunakan untuk memuat *view* yang diperlukan

yaitu *header*, *sidebar*, *jadwal*, dan *footer*. Sehingga, dengan fungsi ini, halaman jadwal dapat ditampilkan dengan tampilan yang sudah terstruktur dan menggunakan data yang diambil dari *database*.

NO	KELAS	HARI	NAMA MATA PELAJARAN	SEMESTER	JAM	AKSI
1	Kelas 1	Selasa	Bahasa Indonesia	ganjil	16:33:00 - 16:33:00	<input checked="" type="checkbox"/>
2	Kelas 1	Senin	Bahasa Indonesia	ganjil	07:00:00 - 08:30:00	<input checked="" type="checkbox"/>

Gambar 4. 55 Halaman Detail Jadwal Pelajaran Admin

Pada gambar 4. 55 merupakan halaman detail jadwal pelajaran admin. Pada halaman detail jadwal pelajaran menampilkan data jadwal pelajaran secara mendetail.

```

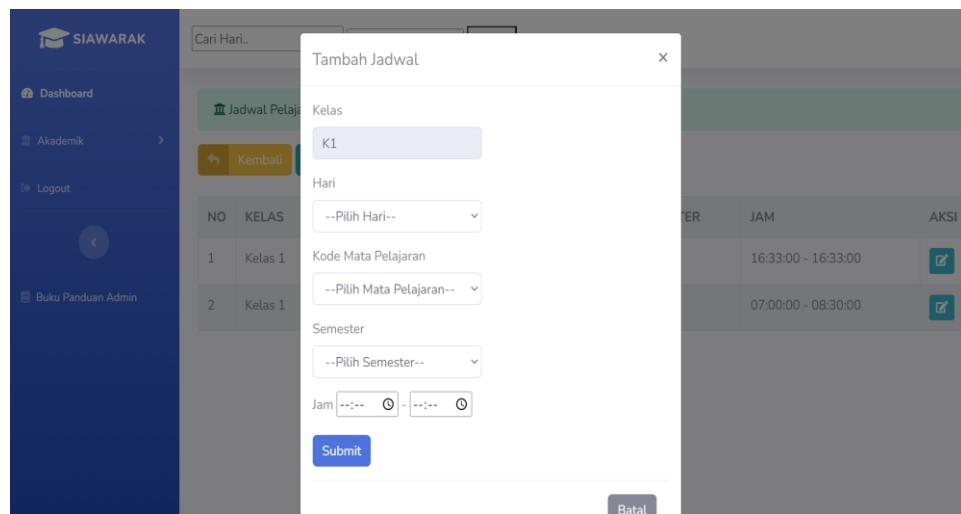
public function listJadwal($kode_kls)
{
    $data['kode_kls'] = $kode_kls;
    $data['mapel'] = $this->Mapel_model->listMapel();
    $data['semester'] = $this->Semester_model->listSemester();
    $data['jadwal'] = $this->Jadwal_model->getJadwalWhereId($kode_kls);

    $this->load->view('admin/header');
    $this->load->view('admin/sidebar');
    $this->load->view('administrator/jadwal_detail', $data);
    $this->load->view('admin/footer');
}

```

Gambar 4. 56 Potongan Code Detail Jadwal

Pada gambar 4. 56 merupakan potongan kode detail jadwal merupakan sebuah function yang bertujuan untuk menampilkan jadwal pelajaran berdasarkan kode kelas yang dipilih. Pada baris pertama *function*, diinisialisasikan variabel “\$kode\_kls” yang diisi dengan parameter yang dikirimkan ke *function* tersebut. Baris selanjutnya digunakan untuk memanggil beberapa data dari *model* seperti list mapel dan list semester yang akan digunakan pada tampilan nanti. Selain itu, juga dipanggil *function* “getJadwalWhereId()” dari *model* “Jadwal\_model” yang akan mengambil data jadwal pelajaran dari kelas yang ditentukan. Data tersebut kemudian dimasukkan ke dalam variabel \$data. Terakhir, dilakukan pemanggilan *view* untuk menampilkan halaman "jadwal\_detail" dengan *parameter* \$data. View tersebut akan ditampilkan dengan menggunakan *template header, sidebar, dan footer* dari admin.



Gambar 4. 57 Halaman Tambah Jadwal Pelajaran Admin

Pada gambar 4. 57 merupakan halaman tambah jadwal pelajaran yang ada pada admin. Pada halaman tambah jadwal pelajaran menampilkan *form* input jadwal pelajaran. Klik simpan jika sudah mengisi *form*. Pada gambar 4. 57 merupakan potongan *code* pada *controller* dari tambah jadwal.



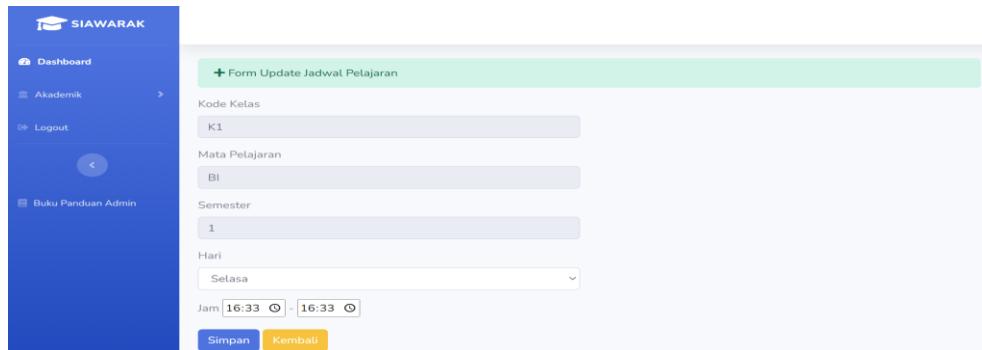
```

public function savejadwal()
{
    $data = [
        'kode_kls' => $this->input->post('kode_kls'),
        'hari' => $this->input->post('hari'),
        'kode_smt' => $this->input->post('kode_smt'),
        'kode_mp' => $this->input->post('kode_mp'),
        'waktu_awal' => $this->input->post('waktu_awal'),
        'waktu_akhir' => $this->input->post('waktu_akhir')
    ];
    $this->Jadwal_model->insert($data);
    $this->session->set_flashdata('pesan', '<div class="alert alert-success alert-dismissible fade show" role="alert">
        Jadwal Berhasil Ditambahkan!
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>');
    return redirect('administrator/Jadwal/listJadwal/' . $data['kode_kls']);
}

```

Gambar 4. 58 Potongan *Code* Tambah Jadwal

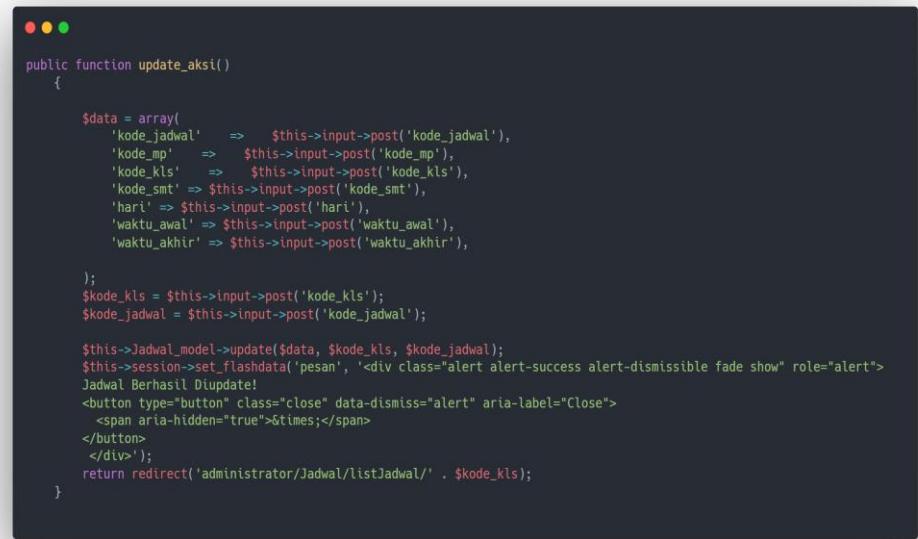
Pada gambar 4. 58 merupakan *controller* dari jadwal yang diberi nama sebuah fungsi "savejadwal" yang terdiri dari beberapa bagian. Pada kode ini fungsi ini akan dipanggil ketika tombol "Submit" pada form jadwal ditekan. Pertama, kode mengambil inputan data jadwal dari form dan memasukkannya ke dalam *array* \$data. Kemudian, *array* \$data disimpan ke dalam *database* melalui *model* "Jadwal\_model" menggunakan "*method insert()*". Selanjutnya, sebuah *flash data* diset untuk memberikan pesan sukses setelah jadwal berhasil ditambahkan. Terakhir, kode mengarahkan pengguna ke halaman "listJadwal()" dengan mengirimkan kode kelas sebagai parameter.



The screenshot shows a part of an administrator dashboard for 'SIAWARAK'. On the left, there's a sidebar with 'Dashboard', 'Akademik' (selected), 'Logout', and 'Buku Panduan Admin'. The main area has a green header bar with '+ Form Update Jadwal Pelajaran'. Below it, there are several input fields: 'Kode Kelas' (K1), 'Mata Pelajaran' (BI), 'Semester' (1), 'Hari' (Selasa), and a time range 'Jam' (16:33 - 16:33). At the bottom are two buttons: 'Simpan' (blue) and 'Kembali' (orange).

Gambar 4. 59 Halaman Ubah Jadwal Admin

Pada gambar 4. 59 merupakan halaman ubah jadwal admin, pada halaman ubah jadwal menampilkan *form* yang akan di ubah diantaranya yaitu hari dan jam. Pada gambar 4. 60 merupakan potongan kode dari ubah jadwal.



```

● ● ●

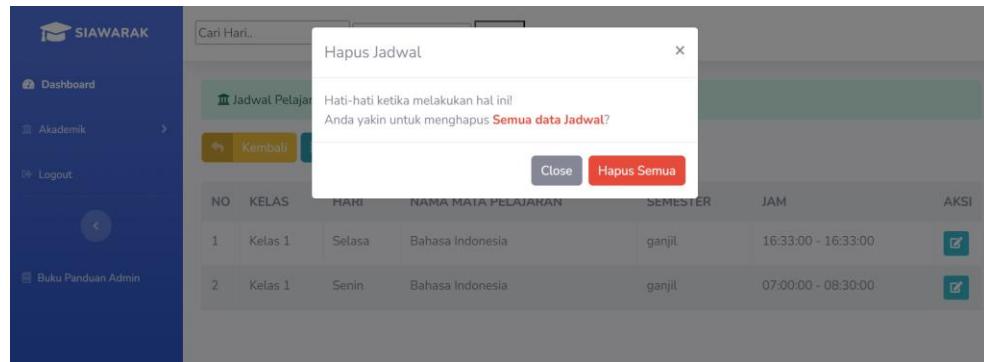
public function update_aksi()
{
    $data = array(
        'kode_jadwal' => $this->input->post('kode_jadwal'),
        'kode_mp' => $this->input->post('kode_mp'),
        'kode_kls' => $this->input->post('kode_kls'),
        'kode_smt' => $this->input->post('kode_smt'),
        'hari' => $this->input->post('hari'),
        'waktu_awal' => $this->input->post('waktu_awal'),
        'waktu_akhir' => $this->input->post('waktu_akhir'),
    );
    $kode_kls = $this->input->post('kode_kls');
    $kode_jadwal = $this->input->post('kode_jadwal');

    $this->Jadwal_model->update($data, $kode_kls, $kode_jadwal);
    $this->session->set_flashdata('pesan', '<div class="alert alert-success alert-dismissible fade show" role="alert">
Jadwal Berhasil Diupdate!
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>');
    return redirect('administrator/Jadwal/listJadwal/' . $kode_kls);
}

```

Gambar 4. 60 Potongan *Code* Ubah Jadwal

Pada gambar 4. 60 merupakan *controller* dari jadwal dengan nama fungsi "update\_aksi" yang terdiri dari beberapa bagian fungsi. Fungsi ini akan dipanggil ketika tombol "Simpan" pada form jadwal ditekan. Pada baris pertama, variabel `\$data` dibuat sebagai sebuah *array* yang berisi data-data yang akan diubah. Kemudian, nilai dari "\$kode\_kls" dan "\$kode\_jadwal" ditentukan melalui *input* yang diterima oleh fungsi. Selanjutnya, fungsi *update* pada "Jadwal\_model" dipanggil dengan parameter "\$data", "\$kode\_kls", dan "\$kode\_jadwal". Kemudian, pesan sukses diset dengan menggunakan *session* dan fungsi *redirect* digunakan untuk mengarahkan admin ke halaman daftar jadwal dengan parameter \$kode\_kls.



Gambar 4. 61 Halaman Hapus Jadwal Admin

Pada gambar 4. 61 merupakan halaman hapus jadwal admin, pada halaman hapus jadwal menampilkan kotak dialog. Klik hapus semua. Untuk kode dapat dilihat pada gambar 4. 62

```
public function delete($id)
{
    $where = array('kode_kls' => $id);
    $this->Jadwal_model->hapus_data($where, 'jadwal');
    $this->session->set_flashdata('pesan', '<div class="alert alert-danger alert-dismissible fade show" role="alert">
Data Jadwal Berhasil Dihapus!
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>');
    return redirect('administrator/Jadwal/listJadwal/' . $id);
}
```

Gambar 4. 62 Potongan Code Hapus Semua Data Jadwal

Gambar 4.62 adalah sebuah fungsi yang ada dalam *controller* jadwal digunakan untuk menghapus data jadwal dengan parameter id (kode\_kls). Pertama, fungsi ini akan membuat sebuah *array* Bernama “\$where” yang berisi kondisi untuk menghapus data yang memiliki “kode\_kls” sama dengan nilai parameter “\$id”. Kemudian, fungsi ini akan memanggil fungsi "hapus\_data" dari *model* "Jadwal\_model" dengan argumen “\$where” dan 'jadwal' sebagai nama tabel yang akan dihapus. Setelah itu, fungsi ini akan membuat sebuah pesan *flash data* dengan status *alert-danger* yang berisi informasi bahwa data jadwal berhasil dihapus. Terakhir, fungsi ini akan melakukan redirect ke halaman "listJadwal" dengan parameter “\$id”. Untuk menampilkan *alert* dalam hapus semua data jadwal yaitu

ketika tombol hapus tersebut di klik, maka akan menjalankan sebuah fungsi dengan menampilkan sebuah *modal* untuk mengonfirmasi penghapusan semua data jadwal. *Modal* ini memiliki header yang berisi judul, "Hapus Jadwal", serta tombol close di pojok kanan atas. Isi modal berisi pesan peringatan untuk memastikan user yakin akan menghapus semua data jadwal. Di bagian bawah modal terdapat tombol "Close" untuk membatalkan penghapusan dan tombol "Hapus Semua" untuk menghapus data jadwal. Tombol "Hapus Semua" menggunakan fungsi "base\_url()" untuk mengarahkan ke fungsi "delete()" pada *controller* Jadwal, dengan parameter "\$kode\_kls" sebagai id data jadwal yang akan dihapus. Dapat lihat kodennya pada gambar 4. 63.



```


##### Hapus Jadwal



Hati-hati ketika melakukan hal ini!



Anda yakin untuk menghapus <b>Semua data Jadwal</b>?


```

Gambar 4. 63 Potongan Code Hapus Semua Jadwal *Modal*

#### 4.2.4 Pengujian

Pada langkah berikutnya setelah implementasi yaitu peneliti mengetes sistem informasi akademik apakah sudah berjalan secara fungsional atau belum dan input ataupun outputnya diharapkan sesuai keinginan. Tes yang digunakan adalah testing *blackbox*, pengujian perangkat lunak menggunakan *blackbox* adalah suatu metode pengujian perangkat lunak dimana *tester* tidak memiliki akses ke sumber kode atau implementasi internal dari sistem yang diuji. Testing di lakukan oleh peneliti untuk menemukan kesalahan pada sistem yang di bangun. Peneliti membuat table rencana kelas uji dan butir uji pada Tabel 4.16

Tabel 4. 16 Rencana Kelas Uji dan Butir Uji

No	Kelas Uji	Butir Uji
1	Halaman Pengguna	Login, halaman jadwal pengguna, halaman presensi pengguna
2	Halaman Admin	Login, halaman guru admin, halaman kelas admin, halaman siswa admin, halaman <i>user</i> admin, halaman mata pelajaran admin, halaman tahun akademik admin, halaman semester admin, halaman jadwal pelajaran admin, halaman presensi admin

Kemudian tes dilakukan untuk masing-masing menguji item berupa membuat tabel tes fungsional yang berisi dari kolom: kelas uji, uji batas, uji kasus dan juga hasil yang diharapkan dari hasil uji. Akan dijelaskan pada Tabel 4.17 dibawah ini:

Tabel 4. 17 Task Pegujian Sistem Informasi Akademik Admin

No Task	Form Uji	Uji Batasan	Skenario Uji	Hasil yang diharapkan	Hasil Uji
T1	<b>Login</b>	Admin	Memasukan username dan password pada sistem informasi akademik dengan benar	Ketika memasuki kedalam menu login akan muncul form login diisi dengan akun username dan password yang benar akan berhasil masuk kedalam Menu Dashboard untuk admin	[√] Berhasil [] Gagal
T2		Admin	Masukan username dan passwoard yang salah pada sistem informasi akademik	Ketika masuk kedalam menu login dan terdapat form pengisian, dan apabila dimasukan username dan passwoard yang salah akan gagal masuk dan kembali lagi ke menu form login	[] Gagal [√] Berhasil
T3	<b>Halaman Guru</b>	Admin	Button tambah guru untuk menambah data guru	Ketika admin telah berada di halaman guru kemudian admin menekan tombol tambah guru lalu	[√] Berhasil [] Gagal

				mengisi semua form pada data tambah guru lalu tekan simpan, maka akan menampilkan data berhasil ditambah dan muncul pada halaman guru	
<b>T4</b>		Admin	<i>Button</i> tambah guru untuk menambah data guru	Ketika admin telah berada di halaman guru kemudian admin menekan tombol tambah guru lalu mengisi sebagian form pada data tambah guru lalu tekan simpan, maka akan menampilkan <i>alert</i> wajib diisi	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
<b>T5</b>		Admin	<i>Button</i> ubah untuk mengubah data guru	Ketika admin mengubah data guru dengan menggunakan tombol ubah pada bagian aksi, kemudian ubah data guru lalu tekan simpan maka akan menampilkan data berhasil diupdate	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
<b>T6</b>		Admin	<i>Button</i> hapus untuk menghapus data guru	Ketika admin telah menambah data kemudian hapus dengan tombol hapus pada bagian aksi. Akan memunculkan <i>alert</i> , selanjutnya tekan iya selanjutnya menampilkan data guru berhasil dihapus	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
<b>T7</b>		Admin	<i>Button</i> melihat detail data guru untuk melihat secara detail guru	Ketika admin telah menambah data kemudian cek detail data guru dengan tombol cek detail pada bagian aksi selanjutnya akan menampilkan detail	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

				guru	
<b>T8</b>	<b>Halaman Kelas</b>	Admin	<i>Button tambah kelas menambah pada data kelas</i>	Ketika admin telah berada di halaman kelas kemudian admin menekan tombol tambah kelas lalu mengisi semua form pada data tambah kelas lalu tekan simpan, maka akan menampilkan data berhasil ditambah dan muncul pada halaman kelas	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
<b>T9</b>		Admin	<i>Button tambah kelas untuk menambah pada data kelas</i>	Ketika admin telah berada di halaman kelas kemudian admin menekan tombol tambah kelas lalu mengisi sebagian form pada data tambah kelas lalu tekan simpan, maka akan menampilkan <i>alert</i> wajib diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
<b>T10</b>		Admin	<i>Button ubah untuk mengubah data kelas</i>	Ketika admin mengubah data kelas dengan menggunakan tombol ubah pada bagian aksi, kemudian ubah data kelas lalu tekan simpan maka akan menampilkan data berhasil diupdate	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
<b>T11</b>		Admin	<i>Button hapus untuk menghapus data kelas</i>	Ketika admin telah menambah data kemudian hapus dengan tombol hapus pada bagian aksi. Akan memunculkan alert, selanjutnya tekan iya selanjutnya akan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

				menampilkan data kelas berhasil dihapus	
T12	Halaman Siswa	Admin	Button tambah siswa menambah pada data siswa	Ketika admin telah berada di halaman siswa kemudian admin menekan tombol tambah siswa lalu mengisi semua form pada data tambah siswa lalu tekan simpan, maka akan menampilkan data berhasil ditambah dan muncul pada halaman siswa	[ <input type="checkbox"/> ] Gagal [ <input checked="" type="checkbox"/> ] Berhasil
T13		Admin	Button tambah siswa utnuk menambah data siswa	Ketika admin telah berada di halaman siswa kemudian admin menekan tombol tambah siswa lalu mengisi sebagian form pada data tambah siswa lalu tekan simpan, maka akan menampilkan <i>alert</i> wajib diisi	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
T14		Admin	Button upload csv utnuk menambah data siswa	Ketika admin telah berada di halaman siswa kemudian admin menekan tombol upload siswa lalu <i>upload file csv</i> yang sudah diubah kemudian <i>csv</i> disimpan akan memunculkan <i>alert</i> berhasil	[ <input type="checkbox"/> ] Gagal [ <input checked="" type="checkbox"/> ] Berhasil
T15		Admin	Button upload csv untuk menambah data siswa	Ketika admin mengunggah <i>file</i> bukan <i>csv</i> maka akan menampilkan peringatan	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
T16		Admin	Button ubah untuk mengubah data siswa	Ketika admin mengubah data siswa dengan menggunakan tombol ubah pada bagian aksi, kemudian ubah data guru lalu tekan simpan	[ <input type="checkbox"/> ] Gagal [ <input checked="" type="checkbox"/> ] Berhasil

				maka akan menampilkan data berhasil diupdate	
T17		Admin	<i>Button hapus semua siswa untuk menghapus data semua siswa</i>	Ketika admin telah menambah data kemudian hapus semua siswa dengan tombol hapus semua siswa. Akan memunculkan alert, selanjutnya tekan hapus semua maka data akan dihapus semua	[√] Berhasil [] Gagal
T18		Admin	<i>Button download data untuk download data dengan keluaran csv</i>	Ketika admin telah menambah data kemudian menkan tombol download data maka akan download semua data siswa yang ada pada halaman siswa	[] Gagal [√] Berhasil
T19		Admin	<i>Button download gambar untuk mendownload gambar data siswa</i>	Ketika admin telah menambah data kemudian menkan tombol download gambar maka akan download semua gambar siswa yang ada pada halaman siswa	[√] Berhasil [] Gagal
T20		Admin	<i>Button melihat detail data siswa untuk melihat secara detail siswa</i>	Ketika admin telah menambah data kemudian cek detail data siswa dengan tombol cek detail pada bagian aksi selanjutnya akan menampilkan detail siswa	[] Gagal [√] Berhasil
T21		Admin	<i>Button hapus untuk menghapus data guru</i>	Ketika admin telah menambah data kemudian hapus dengan tombol hapus pada bagian aksi. Akan memunculkan alert, selanjutnya	[√] Berhasil [] Gagal

				tekan iya	
T22	<b>Halaman User</b>	Admin	<i>Button tambah user</i> menambah pada data <i>user</i>	Ketika admin telah berada di halaman <i>user</i> kemudian admin menekan tombol tambah <i>user</i> lalu mengisi semua form pada data tambah <i>user</i> lalu tekan simpan, maka akan menampilkan data berhasil ditambah dan muncul pada halaman <i>user</i>	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
T23		Admin	<i>Button tambah user</i> untuk menambah pada data <i>user</i>	Ketika admin telah berada di halaman <i>user</i> kemudian admin menekan tombol tambah <i>user</i> lalu mengisi sebagian form pada data tambah <i>user</i> lalu tekan simpan, maka akan menampilkan <i>alert wajib</i> diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
T24		Admin	<i>Button ubah</i> untuk mengubah data <i>user</i>	Ketika admin mengubah data <i>user</i> dengan menggunakan tombol ubah pada bagian aksi, kemudian ubah data <i>user</i> lalu tekan simpan maka akan menampilkan data berhasil diupdate	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
T25		Admin	<i>Button hapus</i> untuk menghapus data <i>user</i>	Ketika admin telah menambah data kemudian hapus dengan tombol hapus pada bagian aksi. Akan memunculkan <i>alert</i> , selanjutnya tekan iya selanjutnya akan menampilkan data <i>user</i> berhasil dihapus	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

T26	<b>Halaman Mata Pelajaran</b>	Admin	<i>Button</i> tambah mata pelajaran menambah pada data mata pelajaran	Ketika admin telah berada di halaman mata pelajaran kemudian admin menekan tombol tambah mata pelajaran lalu mengisi semua form pada data tambah mata pelajaran lalu tekan simpan, maka akan menampilkan data berhasil ditambah dan muncul pada halaman mata pelajaran	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
T27		Admin	<i>Button</i> tambah mata pelajaran untuk menambah pada data mata pelajaran	Ketika admin telah berada di halaman mata pelajaran kemudian admin menekan tombol tambah mata pelajaran lalu mengisi sebagian form pada data tambah mata pelajaran lalu tekan simpan, maka akan menampilkan <i>alert</i> wajib diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
T28		Admin	<i>Button</i> ubah untuk mengubah data mata pelajaran	Ketika admin mengubah data mata pelajaran dengan menggunakan tombol ubah pada bagian aksi, kemudian ubah data mata pelajaran lalu tekan simpan maka akan menampilkan data berhasil diupdate	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
T29		Admin	<i>Button</i> hapus untuk menghapus data mata pelajaran	Ketika admin telah menambah data kemudian hapus dengan tombol hapus pada bagian aksi. Akan memunculkan <i>alert</i> , selanjutnya tekan iya selanjutnya akan menampilkan data mata pelajaran berhasil dihapus	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

T30		Admin	<i>Button</i> melihat detail data mata pelajaran untuk melihat secara detail mata pelajaran	Ketika admin telah menambah data kemudian cek detail data mata pelajaran dengan tombol cek detail pada bagian aksi selanjutnya akan menampilkan detail mata pelajaran	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
T31	<b>Halaman Tahun Akademik</b>	Admin	<i>Button</i> tambah tahun akademik untuk menambah data tahun akademik	Ketika admin telah berada di halaman tahun akademik kemudian admin menekan tombol tambah tahun akademik lalu mengisi semua form pada data tambah tahun akademik lalu tekan simpan, maka akan menampilkan data berhasil ditambah dan muncul pada halaman tahun akademik	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
T32		Admin	<i>Button</i> tambah tahun akademik utnuk menambah data tahun akademik	Ketika admin telah berada di halaman tahun akademik kemudian admin menekan tombol tambah tahun akademik lalu mengisi sebagian form pada data tambah tahun akademik lalu tekan simpan, maka akan menampilkan <i>alert</i> wajib diisi	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
T33		Admin	<i>Button</i> ubah untuk mengubah data tahun akademik	Ketika admin mengubah data tahun akademik dengan menggunakan tombol ubah pada bagian aksi, kemudian ubah data tahun akademik lalu tekan simpan maka akan menampilkan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

				data berhasil diupdate	
T34		Admin	<i>Button hapus untuk menghapus data tahun akademik</i>	Ketika admin telah menambah data kemudian hapus dengan tombol hapus pada bagian aksi. Akan memunculkan alert, selanjutnya tekan iya selanjutnya menampilkan data tahun akademik berhasil dihapus	[ <input type="checkbox"/> ] Gagal [ <input checked="" type="checkbox"/> ] Berhasil
T35	<b>Halaman Semester</b>	Admin	<i>Button tambah semester untuk menambah data semester</i>	Ketika admin telah berada di halaman semester kemudian admin menekan tombol tambah semester lalu mengisi semua form pada data tambah semester lalu tekan simpan, maka akan menampilkan data berhasil ditambah dan muncul pada halaman semester	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
T36		Admin	<i>Button tambah semester utnuk menambah data semester</i>	Ketika admin telah berada di halaman semester kemudian admin menekan tombol tambah semester lalu mengisi sebagian form pada data tambah semester lalu tekan simpan, maka akan menampilkan <i>alert</i> wajib diisi	[ <input type="checkbox"/> ] Gagal [ <input checked="" type="checkbox"/> ] Berhasil
T37		Admin	<i>Button ubah untuk mengubah data semester</i>	Ketika admin mengubah data semester dengan menggunakan tombol ubah	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal

				pada bagian aksi, kemudian ubah data semester lalu tekan simpan maka akan menampilkan data berhasil diupdate	
<b>T38</b>		Admin	<i>Button</i> hapus untuk menghapus data semester	Ketika admin telah menambah data kemudian hapus dengan tombol hapus pada bagian aksi. Akan memunculkan alert, selanjutnya tekan iya selanjutnya menampilkan data semester berhasil dihapus	[ <input type="checkbox"/> ] Gagal [ <input checked="" type="checkbox"/> ] Berhasil
<b>T39</b>	<b>Halaman Jadwal Pelajaran</b>	Admin	<i>Button</i> tambah jadwal untuk menambah data jadwal	Ketika admin telah berada di halaman jadwal kemudian admin dapat menekan tombol aksi untuk melihat detail jadwal dari tiap kelas kemudian tambah jadwal dengan menekan tombol tambah jadwal lalu mengisi semua form pada data tambah jadwal lalu tekan simpan, maka akan menampilkan data berhasil ditambah dan muncul pada halaman detail jadwal	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
<b>T40</b>		Admin	<i>Button</i> tambah jadwal utnuk menambah data jadwal	kemudian admin dapat menekan tombol aksi untuk melihat detail jadwal dari tiap kelas kemudian tambah jadwal dengan menekan tombol tambah	[ <input type="checkbox"/> ] Gagal [ <input checked="" type="checkbox"/> ] Berhasil

				jadwal lalu mengisi sebagian form pada data tambah jadwal lalu tekan simpan, maka akan menampilkan <i>alert</i> wajib diisi	
<b>T41</b>		Admin	<i>Button ubah</i> untuk mengubah data jadwal	Ketika admin mengubah data jadwal dengan menggunakan tombol ubah pada bagian aksi, kemudian ubah data jadwal lalu tekan simpan maka akan menampilkan data berhasil diupdate	[√] Berhasil [] Gagal
<b>T42</b>		Admin	<i>Button hapus</i> semua jadwal untuk menghapus data semua jadwal	Ketika admin telah menambah data kemudian hapus semua jadwal dengan tombol hapus semua jadwal. Akan memunculkan <i>alert</i> , selanjutnya tekan hapus semua maka data akan dihapus semua	[] Gagal [√] Berhasil
<b>T43</b>	<b>Halaman Presensi</b>	Admin	<i>Button lihat</i> detail presensi untuk melihat data presensi	Ketika admin telah berada di halaman presensi kemudian admin dapat menekan tombol aksi untuk melihat detail presensi dari tiap siswa kemudian menampilkan detail presensi siswa	[√] Berhasil [] Gagal

Tabel 4. 18 Task Pengujian Sistem Informasi Akademik Pengguna

<u>No Task</u>	<u>Form Uji</u>	<u>Uji Batasan</u>	<u>Skenario Uji</u>	<u>Hasil yang diharapkan</u>	<u>Hasil Uji</u>
----------------	-----------------	--------------------	---------------------	------------------------------	------------------

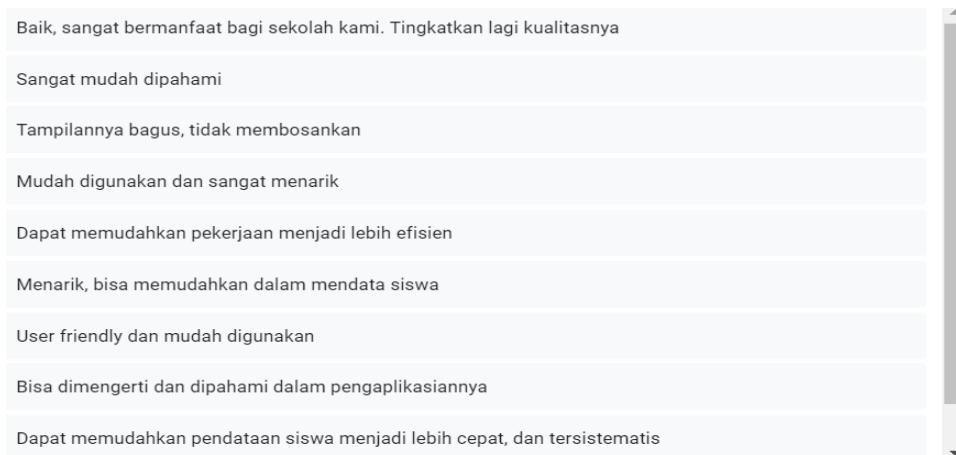
TP1	Login	Pengguna	Memasukan username dan password pada sistem informasi akademik dengan benar	Ketika memasuki kedalam menu login akan muncul form login diisi dengan akun username dan password yang benar akan berhasil masuk kedalam Menu Dashboard untuk pengguna	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
TP2		Pengguna	Masukan username dan passwoard yang salah pada sistem informasi akademik	Ketika masuk kedalam menu login dan terdapat form pengisian, dan apabila dimasukan username dan passwoard yang salah akan gagal masuk dan kembali lagi ke menu form login	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
TP3	<b>Halaman Jadwal</b>	Pengguna	Menampilkan halaman jadwal pelajaran	Ketika pengguna masuk dengan menggunakan login kemudian menekan tombol jadwal pada sub menu akademik, selanjutnya akan menampilkan data jadwal pelajaran yang akan diajarkan kepada siswa	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
TP4	<b>Halaman Presensi</b>	Pengguna	<i>Button</i> tambah presensi untuk menambah data presensi	Ketika pengguna telah berada di halaman presensi kemudian admin dapat menekan tombol aksi untuk melihat detail presensi dari tiap siswa kemudian menampilkan detail presensi siswa pengguna dapat tambah presensi dengan menekan tombol tambah presensi lalu mengisi semua form pada data tambah presensi lalu tekan simpan, maka akan	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil

				menampilkan data berhasil ditambah dan muncul pada halaman detail presensi	
TP5		Pengguna	<i>Button tambah presensi utnuk menambah data presensi</i>	Ketika pengguna telah berada di halaman presensi kemudian admin dapat menekan tombol aksi untuk melihat detail presensi dari tiap siswa kemudian menampilkan detail presensi siswa pengguna dapat tambah presensi dengan menekan tombol tambah presensi lalu mengisi sebagian form pada data tambah jadwal lalu tekan simpan, maka akan menampilkan <i>alert wajib diisi</i>	[√] Berhasil [] Gagal
TP6		Pengguna	<i>Button ubah untuk mengubah data presensi</i>	Ketika pengguna mengubah data presensi dengan menggunakan tombol ubah pada bagian aksi, kemudian ubah data presensi lalu tekan simpan maka akan menampilkan data berhasil diupdate	[] Gagal [√] Berhasil
TP7		Pengguna	<i>Button hapus untuk menghapus data presensi</i>	Ketika pengguna telah menambah data kemudian hapus dengan tombol hapus pada bagian aksi. Akan memunculkan alert, selanjutnya tekan iya selanjutnya akan menampilkan data presensi berhasil dihapus	[√] Berhasil [] Gagal

<b>TP8</b>		Pengguna	<i>Button download</i> data untuk <i>download</i> data dengan keluaran <i>csv</i>	Ketika pengguna telah menambah data kemudian menkan tombol download data maka akan download semua data presensi siswa yang ada pada halaman detail presensi siswa	<input type="checkbox"/> Gagal <input checked="" type="checkbox"/> Berhasil
------------	--	----------	---	---	--

Telah dilakukan testing sesuai task yang sudah di buat diatas dan pada pengujian kali ini memilih seluruh tenaga pendidik, operator sekolah dan kepala sekolah sampel yang menguji sistem informasi akademik SIAwarak, dengan 2 sebagai Admin yaitu Kepala Sekolah dan Operator Sekolah serta seluruh tenaga pendidik yaitu guru berjumlah 8 Guru SD Negeri Siwarak Wetan sebagai *user*.

Setelah dilakukan tes para responden pun diminta mengisi sebuah data diri dan *feedback* tentang sistem informasi akademik SIAwarak, dan dari semua *feedback* yang diterima pada gambar 4. 64.



Baik, sangat bermanfaat bagi sekolah kami. Tingkatkan lagi kualitasnya

Sangat mudah dipahami

Tampilannya bagus, tidak membosankan

Mudah digunakan dan sangat menarik

Dapat memudahkan pekerjaan menjadi lebih efisien

Menarik, bisa memudahkan dalam mendata siswa

User friendly dan mudah digunakan

Bisa dimengerti dan dipahami dalam pengaplikasianya

Dapat memudahkan pendataan siswa menjadi lebih cepat, dan tersistematis

Gambar 4. 64 *Feedback* Sistem Informasi Akademik

Dapat disimpulkan bahwa tidak ada masalah pada sistem dan kebanyakan responden mendapatkan *feedback*, mudah digunakan dan user friendly.

#### **4.2.5 Pemeliharaan**

Tahapan selanjutnya yaitu pemeliharaan atau *maintenance* sebuah sistem, langkah ini perlu dilakukan karena untuk meminimalisir terjadinya bug atau error sehingga mengganggu kenyamanan pengguna. Pada penelitian kali ini peneliti memaintance website sistem informasi akademik selama 4 bulan.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil dari perancangan dan pengujian terhadap sistem informasi akademik menggunakan metode *waterfall*, maka dapat disimpulkan sebagai berikut :

1. Sistem informasi akademik di SD Siwarak Wetan Kecamatan Tambak Kabupaten Banyumas dibuat dengan metode waterfall, melalui tahap-tahap pengembangan seperti analisis kebutuhan, desain, implementasi, pengetesan perangkat lunak, dan pemeliharaan. Penelitian dilakukan dengan identifikasi masalah, studi literatur, pengumpulan data, dan analisis dengan membuat *use case*, *squence diagram* dan *activity diagram* sebagai kerangka alur jalannya sistem. Desain dilakukan dengan merancang basis data, lalu dilanjutkan dengan implementasi melibatkan pembuatan *wireframe*, pemrograman, dan penghostingan agar dapat diakses melalui internet. Pengetesan diambil dengan *purposive sampling*, melakukan pengujian atau evaluasi akhir ketika implementasi berhasil diuji semua menggunakan metode *blackbox testing* pada masing masing rancangan dengan berhasil semua, dan pemeliharaan dilakukan selama 3 bulan. Dengan demikian sistem informasi akademik menggunakan metode *waterfall* dapat memenuhi kebutuhan pengguna.
2. Hasil dari pengujian pada sistem informasi akademik dengan menggunakan *User Acceptance Test (UAT)* sudah tinggi. Perhitungan skala *likert* pada Tabel 4. 20, diketahui bahwa dari 10 responden adalah sistem yang diuji dapat diterima dengan total persentase yang dihasilkan 77,00% dan berada didalam kategori setuju berdasarkan pada skor kriteria Tabel 2.6. Sehingga sistem informasi akademik diterima oleh pengguna.

## 5.2 Saran

Berdasarkan hasil analisis yang dilakukan, penelitian ini menyarankan beberapa hal untuk meningkatkan kualitas dan pengembangan penelitian selanjutnya sebagai berikut :

1. Pada penelitian selanjutnya dalam melakukan pengetesan perangkat lunak dengan menggunakan teknik pengujian non fungsional seperti *usability testing* atau teknik pengujian non fungsional lainnya.
2. Sistem telah dibangun masih memiliki ruang untuk pengembangan lebih lanjut, seperti ditambahkannya fitur-fitur baru yang dapat membantu proses kerja.
3. Hasil aplikasi yang terdapat dipenelitian ini dapat menjadi referensi dasar untuk penelitian selanjutnya.

## DAFTAR PUSTAKA

- [1] Wainah, “Pemanfaatan Website Sebagai Media Pengembangan Loyalitas Wali Murid Di Madrasah Ibtidaiyah Muhammadiyah Pasirmuncang,” *QALAM J. Pendidik. Islam*, vol. 1, no. 1, pp. 125–139, 2020, [Online]. Available: <http://ejournal.stais.ac.id/index.php/qlm/article/view/25>
- [2] N. P. E. Merliana and P. B. A. A. Putra, “Sistem Informasi Akademik dalam pengelolaan pendidikan di Institut Agama Hindu Negeri Tampung Penyang Palangka Raya,” *Satya Sastraharing J. Manaj.*, pp. 47–56, 2021, doi: 10.33363/satya-sastraharing.v5i2.777.
- [3] H. Simanjuntak, B. toni Endaryono, and Balyan, “Bakti Peran Teknologi Informasi dalam Proses Kegiatan Belajar Mengajar di Sekolah Dasar,” *Inventa*, vol. 4, no. 1, pp. 1–10, 2020, doi: 10.36456/inventa.4.1.a2122.
- [4] T. A. Lestari and S. Mariah, “Strategi Pengembangan Sistem Informasi Akademik Dalam Upaya Peningkatan Mutu Pendidikan,” *Kelola J. Manaj. Pendidik.*, vol. 5, no. 1, pp. 15–23, 2018, doi: 10.24246/j.jk.2018.v5.i1.p15-23.
- [5] D. Provinsi Pendidikan Jawa Tengah, “Profil SD Negeri Siwarak Wetan,” *sekolah.data.kemdikbud.go.id*, 2023.  
<https://sekolah.data.kemdikbud.go.id/index.php/chome/profil/b0997d55-2df5-e011-a6ac-e5f24bda0488> (accessed Jun. 19, 2022).
- [6] E. S. Honggara, D. D. Purwanto, and H. Junaedi, “Membangun Sistem POS Supermarket Dengan Tim tanpa Pengalaman Dalam Metodologi SCRUM,” *J. Inf. Syst. Hosp. Technol.*, vol. 3, no. 02, pp. 64–69, 2021, doi: 10.37823/insight.v3i02.171.
- [7] T. Pricillia and Zulfachmi, “Survey Paper : Perbandingan Metode Pengembangan Perangkat Lunak,” vol. X, no. 01, pp. 6–12, 2021.
- [8] Rizaldi, “Penerapan Waterfall Dalam Membangun Sistem Informasi

- Pengolahan Data,” *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. IV, no. 1, 2018.
- [9] I. D. L. - AMIK BSI Purwokerto and F. R. - AMIK BSI Purwokerto, “Sistem Informasi Penjualan Merchandise Berbasis Web Pada PT Come Indonusa Dengan Metode Waterfall,” *Evolusi J. Sains dan Manaj.*, vol. 6, no. 2, pp. 91–97, 2018, doi: 10.31294/evolusi.v6i2.4469.
- [10] L. D. Saputri, “Rancang Bangun Sistem Informasi Akademik Menggunakan Metode Waterfall pada SMA Kosgoro Tangerang,” no. 1, pp. 64–70, 2019.
- [11] K. Fahrezi, A. R. Mulana, S. Melinda, N. Nurhaliza, and S. Mulyati, “Penerapan Model Waterfall dalam Pengembangan Sistem Informasi Akademik Berbasis Web sebagai Sistem Pengolahan Nilai Siswa,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 4, no. 2, p. 98, 2021, doi: 10.32493/jtsi.v4i2.10196.
- [12] Marijan and S. Nurajizah, “Perancangan Sistem Informasi Akademik Berbasis Web Pada SD Islam Luqmanul Hakim Bekasi,” vol. VI, no. 1, pp. 71–78, 2019.
- [13] F. Nadhira, M. I. Wahyuddin, and R. T. K. Sari, “Penerapan Metode Agile Scrum Pada Rancangan SisIAM4,” *J. Media Inform. Budidarma*, vol. 6, no. 1, p. 560, 2022, doi: 10.30865/mib.v6i1.3525.
- [14] D. Triasanti, “Pembuatan Sistem Informasi Akademik Sekolah Menengah Atas Berbasis Website,” 2021.
- [15] Y. N. Fitria Risyda, “Rancang Bangun Sistem Informasi Akademik Pada Sekolah SMK PGRI 28 Dengan Metode Waterfall,” 2018, pp. 189–200.
- [16] D. Wahyudi, A. P. Juledi, and Irmayanti, “Penerapan Framework Codeigniter pada Sistem Absensi QR Code Diskominfo Kabupaten

- Labuhanbatu Selatan,” *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. VII, no. 3, pp. 303–310, 2021.
- [17] A. Saripudin and M. Ardhiansyah, “Sistem Informasi Akademik Sekolah Berbasis Web Menggunakan Model Prototype ( Studi Kasus : Smk Bina Mandiri Depok),” *Pros. SINTAK*, vol. 5, no. 1, pp. 86–100, 2020.
  - [18] T. Haryati, J. Banten No, and J. Barat, “Metode Waterfall pada Sistem Informasi Akademik SMPIT Boarding School Thariq Bin Ziyad Cikarang,” *Bina Insa. Ict J.*, vol. 6, no. 2, pp. 155–164, 2019.
  - [19] K. Basuki, “Rancang Bangun Aplikasi Pengiriman Pesan Pada Jaringan Komputer Berbasis Java Pada Perpustakaan SMK Negeri 9 Medan,” *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, pp. 1689–1699, 2019, [Online]. Available: [www.journal.uta45jakarta.ac.id](http://www.journal.uta45jakarta.ac.id)
  - [20] H. Winardi , Farida & Dicky, “Rancang Bangun Sistem Informasi Manajemen Bengkel (Studi Kasus : CV . Anugrah Bogor),” *IJSE – Indones. J. Softw. Eng.*, vol. 3, no. 2, pp. 8–14, 2017, [Online]. Available: <https://ejournal.bsi.ac.id/ejurnal/index.php/ijse/article/view/2813>
  - [21] I. Pangaribuan and F. Subakti, “Sistem Informasi Akademik Berbasis Web pada SMK (Sekolah Menengah Kejuruan) Teknologi Industri Pembangunan Cimahi,” *J. Teknol. dan Inf.*, vol. 9, no. 2, pp. 128–137, 2019, doi: 10.34010/jati.v9i2.1836.
  - [22] M. Imelda ST. and S. K. Muhammad Erik., “Perancangan Sistem Informasi Akademik Pada Sekolah Dasar Negeri Sukajadi 9 Bandung,” *Angew. Chemie Int. Ed.* 6(11), 951–952., no. 5, pp. 2013–2015, 2021.
  - [23] I. N. Rafi Zakaria, “Penerapan Framework Codeigniter Dan Bootstrap Pada Sistem Pengelolaan Pajak Bumi Dan Bangunan Di Desa Sridadi

- Application of Codeigniter and Bootstrap Frameworks To the Land and Building Tax Management System in Sridadi Village,” *J. Informatics Comput.*, vol. 01, no. 01, pp. 17–26, 2022.
- [24] Uminingsih, M. Nur Ichsanudin, M. Yusuf, and S. Suraya, “Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula,” *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 1, no. 2, pp. 1–8, 2022, doi: 10.55123/storage.v1i2.270.
- [25] D. Mallisza, H. S. Hadi, and A. T. Aulia, “Implementasi Model Waterfall Dalam Perancangan Sistem Surat Perintah Perjalanan Dinas Berbasis Website Dengan Metode SDLC,” *J. Tek. Komputer, Agroteknologi Dan Sains*, vol. 1, no. 1, pp. 24–35, 2022, doi: 10.56248/marostek.v1i1.9.
- [26] M Teguh Prihandoyo, “Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web,” *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 126–129, 2018.
- [27] Redaksi Jagoan Hosting, “Pengertian Use Case Diagram, Simbol, Contoh & Cara Buatnya,” <https://www.jagoanhosting.com>, 2022. <https://www.jagoanhosting.com/blog/use-case-diagram/> (accessed Feb. 19, 2023).
- [28] N. L. Junaedi, “Sequence diagram: Pengertian, manfaat, 14 komponen, dan contohnya,” *ekrut.com*, 2022. <https://www.ekrut.com/media/sequence-diagram-adalah> (accessed Feb. 19, 2023).
- [29] M. Rizky, “UML Diagram Activity Diagram,” *Https://Socs.Binus.Ac.Id/*, pp. 20–22, 2019. <https://soc.s.binus.ac.id/2019/11/22/uml-diagram-activity-diagram/> (accessed Feb. 19, 2023).
- [30] Soegiyono, *Metode Penelitian Kuantitatif, Kualitatif dan R&D*, 27th ed. B. 2018.

- [31] P. Permata and W. D. Rahmawati, “Pengembangan Media Pembelajaran Berbasis Multimedia Pada Materi Kalkulus,” *UNION J. Ilm. Pendidik. Mat.*, vol. 6, no. 3, pp. 277–286, 2018, doi: 10.30738/union.v6i3.2985.
- [32] B. Priyatna, A. Lia Hananto, M. Nova, P. Studi Sistem Informasi, and U. Buana Perjuangan Karawang, “Application of UAT (User Acceptance Test) Evaluation Model in Minggon E-Meeting Software Development,” *Systematics*, vol. 2, no. 3, pp. 110–117, 2020.
- [33] D. Taluke, R. S. M. Lakat, A. Sembel, E. Mangrove, and M. Bahwa, “Analisis Preferensi Masyarakat Dalam Pengelolaan Ekosistem Mangrove Di Pesisir Pantai Kecamatan Loloda Kabupaten Halmahera Barat,” *Spasial*, vol. 6, no. 2, pp. 531–540, 2019.

## **LAMPIRAN**

### **Lampiran 1 Pertanyaan wawancara**

#### **Lampiran Daftar Pertanyaan**

1. Apakah sekolah sudah memiliki Sistem Informasi Akademik?
2. Apakah perlu dilakukan digitalisasi di SDN Siwarak Wetan dengan menerapkan Sistem Informasi Akademik?
3. Apakah Bapak/Ibu guru tahu tentang Sistem Informasi Akademik?
4. Mengapa guru SDN Siwarak Wetan membutuhkan Sistem Informasi Akademik?
5. Bagaimana akses internet untuk menjalankan Sistem Informasi Akademik agar berjalan dengan lancar?
6. Apakah penggunaan Sistem Informasi Akademik diperlukan oleh guru di SDN Siwarak Wetan?
7. Seberapa penting penggunaan Sistem Informasi Akademik bagi guru di SDN Siwarak Wetan?
8. Bagaimana efektivitas dari penggunaan Sistem Informasi Akademik yang akan digunakan oleh guru di SDN Siwarak Wetan?
9. Apakah penggunaan Sistem Informasi Akademik dapat berjalan dengan baik di SDN Siwarak Wetan?
10. Apa saja faktor yang mendukung digunakannya Sistem Informasi Akademik untuk guru di SDN Siwarak Wetan?
11. Bagaimana tanggapan guru SDN Siwarak Wetan dengan adanya Sistem Informasi Akademik?
12. Apakah Sistem Informasi Akademik dapat membantu kinreja guru dalam input nilai?
13. Apakah guru SDN Siwarak Wetan setuju adanya Sistem Informasi Akademik?
14. Apa yang menjadi faktor penghambat guru dalam menggunakan Sistem Informasi Akademik?
15. Apa yang akan dilakukan pertama kali oleh guru SDN Siwarak Wetan ketika menggunakan Sistem Informasi Akademik untuk pertama kali?
16. Mengapa Sistem Informasi Akademik ini menjadi penting untuk guru?
17. Kapan Sistem Informasi Akademik ini akan mulai efektif digunakan di SDN Siwarak Wetan?
18. Apakah semua guru mata pelajaran di SDN Siwarak Wetan akan menggunakan Sistem ini?
19. Bersediakah guru SDN Siwarak Wetan mulai menggunakan Sistem Informasi Akademik untuk mengelola nilai hasil belajar siswa?
20. Apakah guru SDN Siwarak Wetan akan dibentuk tim khusus untuk menjadi pengelola Sistem Informasi Akademik?
21. Bagaimana persiapan guru SDN Siwarak Wetan akan diterapkannya digitalisasi dalam proses akademik yang berlangsung?
22. Apakah penggunaan Sistem Informasi Akademik ini akan dilakukan secara berkelanjutan?
23. Seberapa membantu Sistem Informasi Akademik ini untuk guru SDN Siwarak Wetan dalam mengelola hasil belajar siswa?
24. Bagaimana perubahan digitalisasi yang akan terjadi apakah akan secara cepat?
25. Bagaimana Kepala sekolah mengevaluasai penggunaan Sistem Informasi Akademik oleh guru SDN Siwarak Wetan?

## Lampiran 2 Hasil Jawaban Pertanyaan

2 November 2022

No.

Date

1. Belum
2. Bisa dimungkinkan
3. Belum
4. ~~Untuk~~ Untuk mendukung sistem pembelajaran yg menerapkan digitalisasi
5. Ya
6. Ya
7. Penting, tutukan zaman
8. Sangat efektif mendukung
9. Ya
10. Menghantui anggapan yg positif dan dapat membantu sdm tidak nulis manual
11. Bada, Paling SDM.
12. Setuju
13. Tak ada faktor penghambat
14. Sosialisasi menggunakan intranasi digitalisasi dalam pembelajaran sehari-hari
15. Adminstrasi sekolah
16. Sekolah Sosialisasi
17. Ya
18. Harusnya ~~adalah~~ ada
19. Memperbaiki infrastruktur yg ada untuk efisiensi teknologi
20. Sangat bant karena proses lebih cepat
21. Terutama input penenerima (guru)
22. Memerlukan teknologi, serta masukan agar menjadi bahan refleksi untuk lebih baik lagi.

### Lampiran 3 Hasil Survey Google Form

SUYATI, S.Pd.SD  
 ARIF ROHMAN HAKIM, S.Pd  
 NOWO SUHENDRO,S.Pd  
 AGUS SETIONO  
 NASIRI, S.Pdi  
 SRI SUHARYATI, S.Pd  
 JATMIKO, S.Pd.SD  
 AGUS LATIF AZIS FAUZI  
 DWI SETYANINGSIH, S.Pd

**Mengampu disekolah sebagai ?**

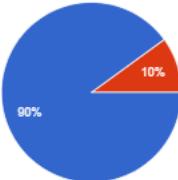
10 jawaban



Kategori	Persentase
Guru Kelas 1	10%
Guru Kelas 2	10%
Guru Kelas 3	10%
Guru Kelas 4	10%
Guru Kelas 5	10%
Guru Kelas 6	10%
Operator Sekolah	10%
Guru Mapel Pendidikan Agama Islam	10%

**Apakah perlu dilakukan digitalisasi di SD Negeri Siwarak Wetan dengan menerapkan Sistem Informasi Akademik berbasis website?**

10 jawaban



Kategori	Persentase
Perlu	90%
Tidak Perlu	10%

**Jika perlu, mengapa alasan diadakannya Sistem Informasi Akademik berbasis website?**

9 jawaban

Dengan adanya sistem informasi akademik dapat membuat sesuatu menjadi lebih cepat dan praktis  
 ingin pembaharuan disekolah  
 Untuk mempermudah mengelola data sekolah khususnya data siswa  
 Supaya menginput data sekolah terutama jadwal pelajaran jadi lebih efisien  
 Alasan saya memilih perlu diadakannya sistem informasi akademik berbasis website untuk mempermudah dalam perekapan absensi secara spesifik  
 Alasannya untuk memudahkan guru dalam pencatatan administrasi sekolah berbasis website  
 Dapat mencoba hal baru seperti pengaplikasian penggunaan website sistem akademik  
 Untuk mendukung adanya peralihan dari manual ke website dengan memanfaatkan teknologi digital  
 Untuk memudahkan dalam meninntu data karena membutuhkan waktu yang lama pada proses

Lampiran 4 Dokumentasi Wawancara kepada kepala sekolah



Lampiran 5 Dokumentasi Pengisian Google Form



Lampiran 6 Dokumentasi Pengujian Sistem Informasi Akademik

