

**PENERAPAN METODE EXTREME PROGRAMMING GUNA MEMBANGUN
MODUL DATA AKADEMIK DAN MODUL PEMBELAJARAN PADA SISTEM
INFORMASI AKADEMIK BERBASIS *WEBSITE* DENGAN *FRAMEWORK*
CODEIGNITER 4 DI SMPN 29 BANDAR LAMPUNG**

(Skripsi)

Oleh

**MUHAMMAD RAFI SATRIA
NPM 2117051046**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**PENERAPAN METODE EXTREME PROGRAMMING GUNA
MEMBANGUN MODUL DATA AKADEMIK DAN MODUL
PEMBELAJARAN PADA SISTEM INFORMASI AKADEMIK BERBASIS
WEBSITE DENGAN *FRAMEWORK* CODEIGNITER 4 DI SMPN 29
BANDAR LAMPUNG**

Oleh

MUHAMMAD RAFI SATRIA

Skripsi

**Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

PENERAPAN METODE EXTREME PROGRAMMING GUNA MEMBANGUN MODUL DATA AKADEMIK DAN MODUL PEMBELAJARAN PADA SISTEM INFORMASI AKADEMIK BERBASIS WEBSITE DENGAN FRAMEWORK CODEIGNITER 4 DI SMPN 29 BANDAR LAMPUNG

Oleh

MUHAMMAD RAFI SATRIA

Perkembangan teknologi informasi mendorong sekolah untuk mengadopsi sistem terkomputerisasi guna meningkatkan efisiensi pengelolaan data dan transparansi informasi akademik. SMPN 29 Bandar Lampung menghadapi tantangan dalam mengelola data akademik secara manual, yang berdampak pada keterlambatan informasi dan risiko kesalahan data. Penelitian ini bertujuan untuk mengembangkan sistem informasi akademik berbasis website dengan menggunakan metode *Extreme Programming (XP)* dan framework CodeIgniter 4. Sistem yang dibangun terdiri dari empat modul utama, yaitu modul data akademik, pembelajaran, informasi publik, dan evaluasi, dengan fokus pengembangan pada modul data akademik dan pembelajaran. Sistem ini memungkinkan pengelolaan data siswa, guru, dan pegawai secara terintegrasi serta memberikan akses informasi nilai dan kemajuan belajar secara real-time kepada siswa, guru, dan orang tua. Pengujian dilakukan menggunakan metode *Black-Box* dengan teknik *Equivalence Partitioning* untuk validasi fungsional dan *User Acceptance Testing (UAT)* untuk mengukur aspek non-fungsional. Hasil pengujian menunjukkan bahwa seluruh fungsi sistem berjalan sesuai spesifikasi, dengan tingkat kepuasan pengguna sebesar 86%. Hasil ini menunjukkan bahwa sistem telah memenuhi sebagian besar kebutuhan pengguna dan dapat meningkatkan efektivitas pengelolaan informasi akademik di lingkungan sekolah.

Kata kunci: Sistem Informasi Akademik, CodeIgniter 4, Extreme Programming, Black-Box, User Acceptance Testing.

ABSTRACT

IMPLEMENTATION OF EXTREME PROGRAMMING METHOD FOR DEVELOPING ACADEMIC DATA AND LEARNING MODULES IN A WEB- BASED ACADEMIC INFORMATION SYSTEM USING CODEIGNITER 4 FRAMEWORK AT SMPN 29 BANDAR LAMPUNG

By

MUHAMMAD RAFI SATRIA

The development of information technology encourages schools to adopt computerized systems to enhance the efficiency of data management and the transparency of academic information. SMPN 29 Bandar Lampung faces challenges in managing academic data manually, which leads to delays in information access and a high risk of data errors. This study aims to develop a web-based academic information system using the Extreme Programming (XP) method and the CodeIgniter 4 framework. The system consists of four main modules: academic data, learning, public information, and evaluation, with development focused on the academic data and learning modules. The system enables integrated management of student, teacher, and staff data, and provides real-time access to student grades and learning progress for students, teachers, and parents. Testing was conducted using the Black-Box method with the Equivalence Partitioning technique for functional validation, and User Acceptance Testing (UAT) to evaluate non-functional aspects. The test results show that all system functions operate according to specifications, with a user satisfaction level of 86%. These results indicate that the system meets most user needs and can improve the effectiveness of academic information management within the school environment.

Keywords: Academic Information System, CodeIgniter 4, Extreme Programming, Black-Box, User Acceptance Testing.

Judul Skripsi

: **PENERAPAN METODE EXTREME
PROGRAMMING GUNA MEMBANGUN
MODUL DATA AKADEMIK DAN
MODUL PEMBELAJARAN PADA
SISTEM INFORMASI AKADEMIK
BERBASIS *WEBSITE* DENGAN
FRAMEWORK CODEIGNITER 4 DI
SMPN 29 BANDAR LAMPUNG**

Nama Mahasiswa

: **Muhammad Rafi Satria**

Nomor Pokok Mahasiswa

: 2117051046

Program Studi

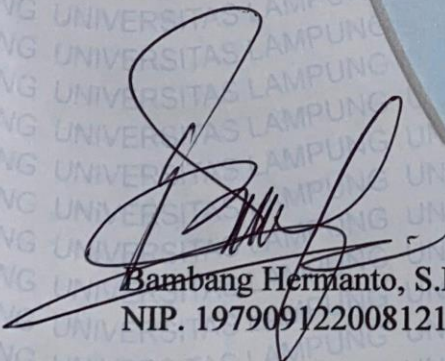
: Ilmu Komputer

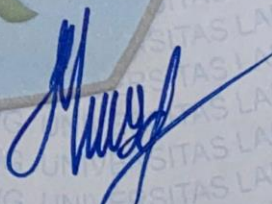
Fakultas

: Matematika dan Ilmu Pengetahuan Alam

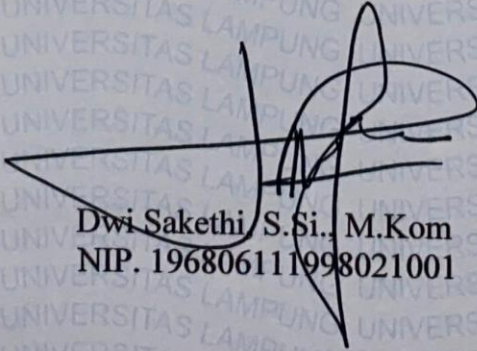
MENYETUJUI,

1. Komisi Pembimbing


Bambang Hermanto, S.Kom., M.Cs
NIP. 197909122008121002


Muhaqiqin, S.Kom., M.T.I.
NIP. 199305252022031009

2. Ketua Jurusan Ilmu Komputer


Dwi Sakethi, S.Si., M.Kom
NIP. 196806111998021001

MENGESAHKAN

1. Tim Penguji

Ketua : Bambang Hermanto, S.Kom., M.Cs

Sekretaris : Muhaqiqin, S.Kom., M.T.I.

Penguji Utama : Ossy Dwi Endah Wulansari, S.SI, M.Si

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng Heri Satria S.Si., M.Si.
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: 19 Juni 2025

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Rafi Satria

NPM : 2117051046

Menyatakan bahwa skripsi saya yang berjudul **"Penerapan Metode Extreme Programming Guna Membangun Modul Data Akademik dan Modul Pembelajaran Pada Sistem Informasi Akademik Berbasis Website Dengan Framework Codeigniter 4 Di SMPN 29 Bandar Lampung"** merupakan karya saya sendiri dan bukan merupakan karya orang lain. Semua isi tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 30 Juni 2025



Muhammad Rafi Satria
NPM. 2117051046

RIWAYAT HIDUP



Penulis dilahirkan di Bandar Lampung pada tanggal 03 Juni 2003 sebagai anak kedua dari 2 bersaudara, dari Bapak Jazuli, S.Pd. dan Ibu Siti Aisyah, S.Pd. Penulis telah menyelesaikan pendidikan formal di SDN 1 Sukarame pada tahun 2015. SMPN 1 Bandar Lampung pada tahun 2018. Dan SMA Negeri 5 Bandar Lampung pada tahun 2021. Penulis terdaftar di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur masuk SBMPTN pada tahun 2021.

Selama menjadi mahasiswa, penulis aktif mengikuti beberapa kegiatan antara lain:

1. Menjadi Anggota bidang Keilmuan Himpunan Mahasiswa Ilmu Komputer Universitas Lampung pada tahun 2022.
2. Menjadi Koordinator Divisi Acara pada kegiatan Workshop Wawancara Kerja pada tahun 2022.
3. Menjadi Asisten Dosen mata kuliah Pemrograman Web Dasar Pada tahun 2024.
4. Mengikuti kegiatan kerja praktik di Bidang Perbendaharaan Badan Pengelola Keuangan dan Aset Daerah (BPKAD) pada tahun 2023.
5. Mengikuti kegiatan Kuliah Kerja Nyata (KKN) Universitas Lampung Periode 2 di Desa Tulung Balak pada tahun 2024.

MOTTO

"Ya Tuhanku, tambahkanlah kepadaku ilmu pengetahuan."

(Q.S Thaha: 114)

"Yoku ganbatta na. Ato wa otona ni makasero."

(Kau sudah berusaha dengan baik. Sisanya serahkan pada orang dewasa.)

(Kento Nanami, Jujutsu Kaisen)

"It always seems impossible until it's done."

(Nelson Mandela)

PERSEMBAHAN

Alhamdulillahillobbilalamin

Puji syukur kepada Allah SWT yang selalu memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan skripsi ini. Shalawat serta salam selalu disanjungkan kepada Nabi Muhammad SAW

Kupersembahkan karya ini kepada:

Kedua Orang Tuaku dan Kakak Tercinta

Yang selalu memberikan dukungan. terimakasih atas segala pengorbanan yang dilakukan, do'a yang selalu dipanjatkan, dan perhatian yang selalu diberikan.

Seluruh Keluarga Besar Ilmu Komputer 2021

Almamater Tercinta, Universitas Lampung

SANWACANA

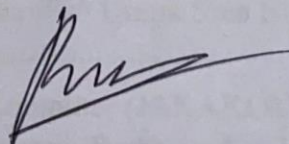
Puji syukur atas kehadiran Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Metode Extreme Programming Guna Membangun Modul Data Akademik dan Modul Pembelajaran Pada Sistem Informasi Akademik Berbasis Website Dengan Framework Codeigniter 4 Di SMPN 29 Bandar Lampung” telah diselesaikan. Tidak lupa shalawat dan salam yang senantiasa dicurahkan kepada baginda Nabi Muhammad SAW yang menjadi suri teladan kepada seluruh umatnya. Pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan skripsi ini, yaitu:

1. Allah SWT yang telah memberikan berkah, hidayah, rahmat, serta karunia-Nya selama proses penyelesaian skripsi ini.
2. Bapak Jazuli, S.Pd. dan Ibu Siti Aisyah, S.Pd. selaku kedua orang tua dan Kakak tercinta Alifa Ayu Rizki, S.Stat, yang senantiasa selalu memberikan semangat, dukungan, dan do’a serta membantu dalam segala hal yang tidak terhitung nilainya.
3. Bapak Dr. Eng. Heri Satria S.Si., M.Si. selaku Dekan Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
4. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Ibu Yunda Heningtyas, M.Kom. selaku Sekretaris Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Ibu Anie Rose Irawati, M.Cs. selaku Dosen Pembimbing Akademik yang telah memberikan saran dan dukungan kepada penulis.
7. Bambang Hermanto, S.Kom., M.Cs. selaku Dosen Pembimbing utama dan Bapak Muhaqiqin, S.Kom., M.T.I. atas ketersediannya dalam memberikan bimbingan, motivasi, kritik, serta saran dalam proses penyelesaian skripsi.

8. Ibu Ossy Dwi Endah Wulansari, S.SI, M.Si. selaku Dosen Pembahas yang telah memberikan kritik dan saran hingga masukan kepada penulis selama proses penyelesaian skripsi
9. Bapak dan Ibu Dosen Jurusan Ilmu Komputer Universitas Lampung yang telah memberikan ilmu, pengetahuan serta pengalamannya kepada penulis selama penulis menjadi mahasiswa.
10. Seluruh staf Jurusan Ilmu Komputer yang selalu membantu penulis dari awal hingga akhir masa perkuliahan.
11. Enjelita Aini Natasya yang telah membersamai penyelesaian skripsi dari awal penulisan judul hingga akhir sidang komprehensif.
12. Teman-teman "Kosan Ayah" yaitu Rimuru, Surya, Bagas, Iqbal, John, Qolby, Isal, Abdi, Ferry, Hanip, Kaje, Farel, Wangsa, Reza, dan Zidan selaku rekan seperjuangan dari awal masa perkuliahan.
13. Teman-teman "Tebu Tubaba" yaitu Enjel, Ipeh, Muca, Nathan, Nisa, Tasya, Wirda, Ijah, Peem, Alfa, Iqbal, Hanip, Qolby, Rimuru, Bagas, Reza, dan Zidan selaku rekan seperjuangan dari awal masa perkuliahan.
14. Teman-Teman SMA yang sudah bersama sampai saat ini, Hammam, Fitra
15. Teman-teman Ilmu Komputer yang menjadi keluarga besar Jurusan Ilmu Komputer selama menjalankan masa studi di Jurusan Ilmu Komputer Universitas Lampung

Proses dalam penyusunan skripsi ini tentunya terdapat banyak kekurangan dikarenakan keterbatasan pengetahuan serta pengalaman. Semoga skripsi ini dapat membawa manfaat dan keberkahan bagi perkembangan ilmu pengetahuan terutama bagi seluruh civitas jurusan Ilmu Komputer Universitas Lampung.

Bandar Lampung, 30 Juni 2025



Muhammad Rafi Satria
NPM. 2117051046

DAFTAR ISI

	Halaman
DAFTAR ISI	i
DAFTAR GAMBAR	iii
DAFTAR TABEL	vi
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
II. TINJAUAN PUSTAKA	6
2.1 Penelitian Terdahulu	6
2.1.1 Penerapan Extreme Programming Dalam Sistem Informasi Akademik SDN Kuala Teladas	6
2.1.2 Pengembangan Sistem Informasi Akademik berbasis <i>Web</i> di SMP Negeri 3 Sidikalang.....	7
2.1.3 Sistem Informasi Akademik Berbasis <i>Webiste</i> (Studi Kasus: Smpit Avicenna)	7
2.1.4 Pengembangan Sistem Informasi Akademik Berbasis <i>Website</i> Menggunakan Metode Rapid Application Development (Rad).....	8
2.1.5 Sistem Informasi Akademik Berbasis <i>Web</i> Untuk Sma Bina Warga 1 Palembang	8
2.1.6 Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM) Perancangan Sistem Informasi Akademik Berbasis <i>Web</i> Pada SMP N 1 Muaro Jambi.....	9
2.2 Uraian Landasan Teori.....	9
2.2.1 Pengertian Sistem Informasi Akademik.....	10

2.2.2	Pengertian PHP	10
2.2.3	Pengertian Extreme Programming (XP)	10
2.2.4	Pengertian Unified Modeling Language (UML)	12
2.2.5	Pengertian <i>Class Diagram</i>	15
2.2.6	Pengertian CodeIgniter 4.....	15
2.2.7.	Pengertian User Acceptance Testing (UAT).....	16
2.2.8.	Black Box Testing.....	16
III.	METODOLOGI PENELITIAN.....	18
3.1.	Waktu dan Tempat Penelitian	18
3.2.	Tahapan Penelitian	18
3.2.1.	Tahap Pengumpulan Data	19
3.2.2.	Tahap Pengembangan Sistem	21
3.2.3.	Tahap Penulisan Laporan.....	114
IV.	HASIL DAN PEMBAHASAN.....	115
4.1.	Hasil	115
4.2.	Pembahasan.....	116
4.2.1.	Iterasi.....	116
4.2.2.	Tahap Pengujian.....	195
V.	KESIMPULAN DAN SARAN.....	215
5.1.	Kesimpulan	215
5.2.	Saran.....	215
DAFTAR PUSTAKA.....		216

DAFTAR GAMBAR

Gambar	Halaman
1. Alur Tahapan Extreme Programming	11
2. Tahapan Penelitian	19
3. <i>Use Case Diagram</i>	27
4. <i>CRC Cards</i>	37
5. Halaman Login.....	39
6. Halaman Data Siswa	40
7. Halaman Data Guru dan Pegawai	40
8. Halaman Jadwal Mata Pelajaran	41
9. Halaman Riwayat Adminstrasi	41
10. Halaman Nilai	42
11. Halaman Kemajuan Belajar	42
12. Halaman Manajemen Akun.....	43
13. <i>Class Diagram</i>	44
14. <i>Activity Diagram</i> Melakukan Login.....	47
15. <i>Activity Diagram</i> Melihat Data Siswa.....	48
16. <i>Activity Diagram</i> Melihat Data Guru dan Pegawai.....	49
17. <i>Activity Diagram</i> Melihat Riwayat Administrasi.....	50
18. <i>Activity Diagram</i> Melihat Jadwal Mata Pelajaran	51
19. <i>Activity Diagram</i> Melihat Nilai.....	52
20. <i>Activity Diagram</i> Melihat Kemajuan Belajar.....	53
21. <i>Activity Diagram</i> Menambah Nilai	55
22. <i>Activity Diagram</i> Mengubah Nilai	57
23. <i>Activity Diagram</i> Menghapus Nilai	59
24. <i>Activity Diagram</i> Menambah Jadwal Mata Pelajaran.....	60

25. <i>Activity Diagram</i> Mengubah Jadwal mata Pelajaran	62
26. <i>Activity Diagram</i> Menghapus Jadwal Mata Pelajaran	63
27. <i>Activity Diagram</i> Menambah Data Siswa	65
28. <i>Activity Diagram</i> Mengubah Data Siswa	66
29. <i>Activity Diagram</i> Menghapus Data Siswa	67
30. <i>Activity Diagram</i> Menambah Data Guru dan Pegawai	69
31. <i>Activity Diagram</i> Mengubah Data Guru dan Pegawai	71
32. <i>Activity Diagram</i> Menghapus Data Guru dan Pegawai	72
33. <i>Activity Diagram</i> Menambah Riwayat Administrasi	74
34. <i>Activity Diagram</i> Mengubah Riwayat Administrasi	76
35. <i>Activity Diagram</i> Menghapus Riwayat Administrasi	78
36. <i>Activity Diagram</i> Melakukan Logout.....	79
37. <i>Activity Diagram</i> Menambah Akun	81
38. <i>Activity Diagram</i> Mengubah Akun	82
39. <i>Activity Diagram</i> Menghapus Akun	83
40. <i>Activity Diagram</i> Aktivasi Akun.....	85
41. <i>Activity Diagram</i> Mengatur Status Penilaian.....	86
42. <i>Activity Diagram</i> Reset Password.....	88
43. Halaman Nilai Siswa.....	117
44. Halaman Tambah Nilai	119
45. Halaman Ubah Nilai Siswa	121
46. Halaman Hapus Nilai Siswa	127
47. Halaman Riwayat Administrasi	130
48. Halaman Tambah Riwayat Administrasi	131
49. Halaman Ubah Riwayat Administrasi.....	133
50. Halaman Hapus Riwayat Administrasi	136
51. Halaman Data Siswa	139
52. Halaman Tambah Data Siswa	141
53. Halaman Ubah Data Siswa	143
54. Halaman Hapus Data Siswa	146
55. Halaman Data Guru.....	148

56. Halaman Tambah Data Guru	150
57. Halaman Ubah Data Guru	152
58. Halaman Hapus Data Guru	155
59. Halaman Data Pegawai	157
60. Halaman Tambah Pegawai	159
61. Halaman Ubah Data Pegawai	161
62. Halaman Hapus Data Pegawai	165
63. Halaman Manajemen Akun	167
64. Halaman Tambah Akun	169
65. Halaman Ubah Akun	171
66. Halaman Hapus Data Akun	174
67. Halaman Aktivasi Akun	177
68. Halaman Jadwal Mata Pelajaran	180
69. Halaman Tambah Jadwal Mata Pelajaran	181
70. Halaman Ubah Jadwal Mata Pelajaran	183
71. Halaman Hapus Jadwal Mata Pelajaran	186
72. Halaman Kemajuan Belajar	188
73. Halaman Login	191
74. Halaman Logout	194
75. Aktivitas Pengujian Sistem	195

DAFTAR TABEL

Tabel	Halaman
1. Simbol-simbol <i>Use Case Diagram</i>	13
2. Simbol-simbol <i>Activity Diagram</i>	14
3. User Stories, Acceptance Test Criteria dan Value	21
4. Iteration Plan	25
5. <i>Use Case Description</i> untuk Melakukan Login	88
6. <i>Use Case Description</i> untuk Melihat Data Siswa	89
7. <i>Use Case Description</i> untuk Melihat Data Guru dan Pegawai	89
8. <i>Use Case Description</i> untuk Melihat Riwayat Administrasi	90
9. <i>Use Case Description</i> untuk Melihat Jadwal Mata Pelajaran	90
10. <i>Use Case Description</i> untuk Melihat Nilai	90
11. <i>Use Case Description</i> untuk Melihat Kemajuan Belajar	91
12. <i>Use Case Description</i> untuk Menambah Nilai	91
13. <i>Use Case Description</i> untuk Mengubah Nilai	92
14. <i>Use Case Description</i> untuk Menghapus Nilai	92
15. <i>Use Case Description</i> untuk Menambah Jadwal Mata Pelajaran	93
16. <i>Use Case Description</i> untuk Mengubah Jadwal Mata Pelajaran	93
17. <i>Use Case Description</i> untuk Menghapus Jadwal Mata Pelajaran	94
18. <i>Use Case Description</i> untuk Menambah Data Siswa	94
19. <i>Use Case Description</i> untuk Mengubah Data Siswa	95
20. <i>Use Case Description</i> untuk Menghapus Data Siswa	95
21. <i>Use Case Description</i> untuk Menambah Data Guru dan Pegawai	95
22. <i>Use Case Description</i> untuk Mengubah Data Guru dan Pegawai	96
23. <i>Use Case Description</i> untuk Menghapus Data Guru dan Pegawai	96
24. <i>Use Case Description</i> untuk Menambah Riwayat Administrasi	97

25. <i>Use Case Description</i> untuk Mengubah Riwayat Administrasi	97
26. <i>Use Case Description</i> untuk Menghapus Riwayat Administrasi.....	98
27. <i>Use Case Description</i> untuk Melakukan Logout	98
28. <i>Use Case Description</i> untuk Menambah Akun.....	99
29. <i>Use Case Description</i> untuk Mengubah Data Akun	99
30. <i>Use Case Description</i> untuk Menghapus Data Akun	100
31. <i>Use Case Description</i> untuk Aktivasi Akun	100
32. <i>Use Case Description</i> Mengatur Status Penilaian	101
33. <i>Use Case Description</i> Reset Password	101
34. Matriks Kebutuhan Fungsional <i>As-Is</i> dan <i>To-Be</i>	102
35. Matriks Kebutuhan Non-Fungsional <i>As-Is</i> dan <i>To-Be</i>	104
36. Skenario Black Box testing	106
37. Skenario User Acceptance Testing	113
38. Hasil Pengujian Black-Box	196
39. Skenario User Acceptance Testing	206
40. Hasil Jawaban Responden.....	207
41. Hasil <i>Acceptance Testing</i>	208

I. PENDAHULUAN

1.1 Latar Belakang

Dunia pada saat ini tengah menghadapi era teknologi atau bisa disebut dengan revolusi industri 4.0 (Putri Irianti et al., 2023). Perkembangan teknologi informasi saat ini telah memberi manfaat yang besar dalam dunia pendidikan, pembangunan website sekolah dan pembangunan sistem informasi akademik berbasis komputer adalah sebagian pemanfaatan teknologi informasi (Setiawan et al., 2023). Sistem Informasi Akademik merupakan sebuah aplikasi yang dirancang untuk memenuhi kebutuhan pengelolaan data administratif di lingkungan sekolah. Tujuannya adalah untuk memastikan data akademik dapat dikelola dengan lebih efektif, sehingga mendukung pengambilan keputusan yang lebih baik. (Muhamad Solahudin, 2021). Sistem ini tidak hanya mempermudah pengelolaan data, tetapi juga mempercepat distribusi informasi yang berkaitan dengan kegiatan akademik, sehingga menciptakan lingkungan pembelajaran yang lebih efisien dan terstruktur.

SMPN 29 Bandar Lampung sebagai salah satu sekolah menengah pertama yang terus berusaha meningkatkan kualitas pendidikan, menghadapi tantangan dalam pengelolaan data akademik dan pembelajaran. Dengan jumlah siswa yang mencapai 934 orang, proses pengelolaan data siswa, guru, dan pegawai yang masih dilakukan secara manual dan terpisah-pisah menjadi semakin kompleks, memerlukan waktu yang lama, serta berisiko menimbulkan kesalahan data. Pada saat ini, pengelolaan data akademik masih dilakukan secara manual dan terpisah-pisah menggunakan buku administrasi dan *spreadsheet* yang berbeda-beda. Proses penginputan data memerlukan waktu lama dan rawan terjadi kesalahan atau duplikasi data. Pemantauan

kemajuan belajar siswa bergantung pada laporan berkala yang disampaikan secara tertulis, sehingga informasi tidak dapat diakses secara langsung dan *real-time* oleh siswa, guru, maupun orang tua. Hal ini mengakibatkan koordinasi antar pihak kurang optimal dan pengambilan keputusan menjadi kurang cepat dan tepat. Selain itu, sistem yang ada belum mendukung pemantauan kemajuan belajar siswa secara *real-time*, yang merupakan elemen penting dalam pembelajaran modern. Oleh karena itu, diperlukan sebuah sistem informasi akademik berbasis *website* yang mampu mengintegrasikan data akademik dan pembelajaran dalam satu platform, sehingga informasi dapat dikelola dengan lebih efektif, transparan, dan mudah diakses oleh seluruh pihak yang berkepentingan. Selain itu, sistem ini juga memungkinkan keterlibatan orang tua dalam memantau perkembangan pembelajaran siswa secara *real-time*, sehingga mereka dapat lebih aktif dalam mendukung proses pendidikan anaknya.

Metode Extreme Programming (XP) dipilih sebagai pendekatan dalam pembangunan sistem ini karena metode ini dikenal fleksibel, iteratif, dan menekankan pada kolaborasi antara tim pengembang dan pengguna. Extreme Programming (XP) merupakan sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel (Ardiansah et al., 2023). XP juga memiliki siklus pengembangan yang cepat dan berfokus pada kebutuhan pengguna, sehingga dapat menghasilkan sistem yang sesuai dengan kebutuhan sekolah dan dapat disesuaikan seiring berjalannya waktu. Dalam proyek ini, sistem informasi akademik berbasis *website* dibangun menggunakan *framework* CodeIgniter 4 yang terkenal ringan dan mendukung pengembangan aplikasi berbasis *web* secara cepat. *Framework* CodeIgniter merupakan aplikasi sumber terbuka yang berupa kerangka kerja PHP dengan model MVC (Model, View, Controller) untuk membangun situs *web* dinamis dengan menggunakan PHP (Dristyan et al., 2021).

Sistem yang dikembangkan terdiri dari empat modul utama, yaitu modul data akademik, modul pembelajaran, modul informasi publik, dan modul evaluasi. Disini penulis bertanggungjawab atas modul data akademik dan modul pembelajaran. Modul data akademik akan mencakup pengelolaan data siswa, guru dan pegawai sekolah, jadwal pelajaran, dan riwayat administrasi yang lebih terintegrasi, efisien, terkomputerisasi, serta mudah diakses secara *real-time* oleh pihak yang berkepentingan, guna meningkatkan transparansi dalam pengelolaan data sekolah, sedangkan modul pembelajaran akan mencakup fitur nilai siswa dan pemantauan kemajuan belajar yang terkomputersisasi dan memungkinkan akses informasi terkait pembelajaran yang cepat, mudah bagi siswa, guru, dan orang tua di SMPN 29 Bandar Lampung, agar dapat mendukung proses pembelajaran yang lebih transparan. Setelah penerapan sistem informasi akademik berbasis *website*, seluruh proses pengelolaan data akademik menjadi terintegrasi dalam satu platform digital yang dapat diakses secara online oleh seluruh pihak terkait. Data siswa, guru, pegawai, jadwal pelajaran, dan nilai siswa dapat diinput dan diperbarui secara cepat dan *real-time*, mengurangi risiko kesalahan data dan duplikasi. Sistem ini memungkinkan pemantauan perkembangan belajar siswa secara langsung, sehingga guru dapat memberikan *feedback* secara cepat dan orang tua dapat memantau kemajuan belajar anaknya kapan saja. Proses komunikasi menjadi lebih efisien dan transparan, mendukung pengambilan keputusan yang tepat waktu, serta meningkatkan kualitas layanan pendidikan di SMPN 29 Bandar Lampung. Dengan adanya sistem ini, diharapkan sekolah dapat mengelola informasi akademik dengan lebih baik, serta memantau perkembangan belajar siswa secara efektif dan efisien.

Melalui penelitian ini, diharapkan penerapan metode Extreme Programming dapat mempercepat proses pengembangan dan memaksimalkan kualitas sistem yang dibangun, sehingga memberikan dampak positif bagi pengelolaan pendidikan di SMPN 29 Bandar Lampung.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, rumusan masalah yang dapat diajukan adalah sebagai berikut.

1. Bagaimana mengembangkan sistem terkomputerisasi untuk mempermudah akses informasi pembelajaran yang cepat bagi siswa, guru, dan orang tua di SMPN 29 Bandar Lampung guna mendukung transparansi belajar, atau keterbukaan data yang memungkinkan semua pihak dapat mengakses informasi pembelajaran secara jelas dan *real-time*.
2. Bagaimana mengembangkan sistem terkomputerisasi yang terintegrasi dan *real-time* untuk meningkatkan pengelolaan serta pemantauan data akademik di SMPN 29 Bandar Lampung guna mendukung transparansi data sekolah, atau keterbukaan informasi akademik yang memungkinkan pemantauan dan akses data secara terbuka bagi seluruh pemangku kepentingan.

1.3 Batasan Masalah

Berdasarkan latar belakang yang telah diberikan, berikut adalah batasan masalah yang dapat ditetapkan untuk fokus penelitian.

1. Sistem informasi yang dikembangkan berbasis *website* dengan menggunakan Framework CodeIgniter 4.
2. Sistem ini hanya digunakan di SMPN 29 Bandar Lampung.
3. Sistem ini dibangun menyesuaikan permintaan dan kebutuhan dari pihak SMPN 29 Bandar Lampung.

1.4 Tujuan Penelitian

Adapun tujuan dilakukannya penelitian ini adalah sebagai berikut.

1. Untuk mengembangkan dan menyediakan sistem informasi pembelajaran terkomputerisasi yang memungkinkan akses yang cepat dan mudah bagi siswa, guru, dan orang tua di SMPN 29 Bandar

Lampung, guna mendukung terciptanya proses pembelajaran yang lebih transparan, atau keterbukaan data yang memungkinkan semua pihak memperoleh informasi pembelajaran secara jelas, terbuka, dan *real-time* sehingga mendukung efektivitas dan efisiensi proses pembelajaran.

2. Untuk meningkatkan sistem pengelolaan dan pemantauan data akademik berbasis komputer di SMPN 29 Bandar Lampung, termasuk data guru, pegawai, siswa, dan riwayat administrasi, agar lebih terintegrasi dan efisien, serta dapat diakses secara *real-time* oleh pihak yang berkepentingan, guna meningkatkan transparansi dan akuntabilitas, atau keterbukaan informasi yang memungkinkan pemantauan dan akses data secara terbuka oleh seluruh pemangku kepentingan dalam pengelolaan data sekolah.

1.5 Manfaat Penelitian

Adapun manfaat yang didapatkan dari penelitian ini adalah sebagai berikut.

1. Memudahkan siswa mengakses informasi nilai dan perkembangan belajarnya secara *online*. Hal ini dapat mendorong siswa untuk lebih bertanggung jawab dan termotivasi dalam proses pembelajaran.
2. Membantu sekolah dalam mengelola data akademik seperti data siswa, guru, dan pegawai secara lebih efisien dan terintegrasi. Selain itu, sistem ini juga mempermudah pemantauan nilai dan kemajuan belajar siswa, sehingga dapat mendukung proses pengambilan keputusan yang lebih cepat dan akurat.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian ini tidak terlepas dari penelitian-penelitian sebelumnya yang bertujuan untuk mendukung penelitian ini. Berikut ini merupakan skripsi dan jurnal penelitian yang digunakan sebagai referensi dalam penelitian.

2.1.1 Penerapan Extreme Programming Dalam Sistem Informasi Akademik SDN Kuala Teladas

Penelitian ini bertujuan untuk merancang bangun sistem informasi akademik berbasis website di SD Negeri Kuala Teladas guna membantu mengelola dan menyajikan data akademik dengan mudah, cepat, tepat, dan dapat diakses kapan saja tanpa harus bertanya langsung ke sekolah. Penelitian ini menggunakan metode Extreme Programming (XP) yang dikenal fleksibel dalam menghadapi perubahan kebutuhan pengguna. Sistem ini dibangun dengan menggunakan bahasa pemrograman PHP dan SQLyog untuk manajemen basis data. Pengumpulan data dilakukan melalui metode kuisisioner dengan menerapkan pemodelan Technology Acceptance Model (TAM) untuk mengukur kelayakan sistem. Hasil analisis pengujian menunjukkan bahwa sistem informasi akademik yang dibangun mendapatkan rata-rata persentase kelayakan sebesar 92,71% dan dikategorikan sangat layak untuk meningkatkan efektivitas sekolah dalam menyajikan data akademik dengan mudah, cepat, dan tepat (Ardiansah et al., 2023).

2.1.2 Pengembangan Sistem Informasi Akademik berbasis *Web* di SMP Negeri 3 Sidikalang

Penelitian ini bertujuan untuk mengembangkan sistem informasi akademik berbasis *web* di SMP Negeri 3 Sidikalang untuk mempermudah pihak sekolah dalam mengelola data dan mempercepat penyebaran informasi. Penelitian ini menggunakan metode Waterfall, yang terdiri dari tahap pengumpulan kebutuhan, desain, pengembangan, ujicoba pemakaian, dan pemeliharaan. Pengujian sistem menggunakan metode System Usability Scale (SUS) untuk menguji tingkat kepuasan pengguna. Implementasi sistem dilakukan dengan menggunakan bahasa pemrograman PHP dengan *framework* CodeIgniter dan perancangan *database* menggunakan MySQL. Hasil pengujian sistem informasi akademik menunjukkan bahwa sistem ini layak digunakan. Berdasarkan hasil penelitian, sistem informasi akademik yang dikembangkan dapat mempermudah pengelolaan data dan penyebaran informasi akademik di lingkungan SMP Negeri 3 Sidikalang (Sahat & Sihombing, 2023).

2.1.3 Sistem Informasi Akademik Berbasis *Webiste* (Studi Kasus: Smpit Avicenna)

Penelitian ini bertujuan untuk mengembangkan sistem informasi akademik berbasis *web* di SMPIT Avicenna guna mengatasi kelemahan dari sistem konvensional yang masih digunakan, seperti pencatatan manual dalam buku besar yang mempersulit pencarian data dan memakan waktu serta tempat lebih banyak. Penelitian ini menggunakan metode Waterfall, yang meliputi tahap pengumpulan kebutuhan, desain, pengembangan, ujicoba pemakaian, dan pemeliharaan. Sistem informasi akademik ini dibangun menggunakan bahasa pemrograman PHP dan basis data MySQL. Pengujian sistem menggunakan metode Black Box Testing untuk menguji fungsi-fungsi sistem. Hasil penelitian menunjukkan bahwa sistem informasi akademik ini mampu membantu

mempermudah kinerja aktor di SMPIT Avicenna, yaitu guru, siswa, kurikulum, kepala sekolah, admin, dan tata usaha, dalam pengolahan data akademik dengan lebih efisien dan efektif (Hakim & Meilina, 2022).

2.1.4 Pengembangan Sistem Informasi Akademik Berbasis *Website* Menggunakan Metode Rapid Application Development (Rad)

Penelitian ini bertujuan untuk mengembangkan sistem informasi akademik berbasis web di SMA YATPI Godong guna mengatasi keterbatasan sistem akademik konvensional yang data akademiknya hanya tersimpan di komputer dan tidak dapat diakses oleh seluruh akademisi maupun orang tua wali. Penelitian ini menggunakan metode Rapid Application Development (RAD), yang menekankan pengembangan sistem secara iteratif dengan waktu yang relatif singkat. Sistem informasi akademik ini dibangun menggunakan bahasa pemrograman PHP, JavaScript, HTML, dan basis data MySQL. Pengujian sistem dilakukan dengan metode Black Box Testing untuk memastikan fungsi-fungsi sistem berjalan sesuai harapan. Hasil penelitian menunjukkan bahwa sistem ini dapat memberikan kemudahan akses informasi akademik secara akurat, cepat, dan tepat bagi siswa, guru, administrator, dan orang tua, sehingga diharapkan dapat meningkatkan kualitas layanan akademik di SMA YATPI Godong (Lukman Santoso & Juni Amanullah, 2022).

2.1.5 Sistem Informasi Akademik Berbasis *Web* Untuk Sma Bina Warga 1 Palembang

Penelitian ini bertujuan untuk mengembangkan sistem informasi akademik berbasis web di SMA Bina Warga 1 Palembang sebagai solusi atas keterbatasan pengelolaan data akademik yang masih dilakukan secara manual menggunakan Microsoft Excel dan Word. Sistem ini dirancang untuk mengelola data siswa, guru, mata pelajaran, jadwal pelajaran, absensi, nilai, dan laporan akademik secara

terintegrasi dan dapat diakses dengan mudah oleh admin, guru, siswa, dan kepala sekolah. Penelitian ini menggunakan metode pengembangan perangkat lunak yang berfokus pada analisis kebutuhan, perancangan, implementasi, serta pengujian sistem dengan metode Black Box Testing. Sistem dibangun menggunakan *framework* Laravel dan basis data MySQL. Hasil pengujian menunjukkan bahwa sistem informasi akademik ini dapat membantu mempercepat dan mempermudah proses pengolahan data akademik serta meningkatkan efektivitas pelayanan informasi di SMA Bina Warga 1 Palembang (Sutabri, 2021).

2.1.6 Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM) Perancangan Sistem Informasi Akademik Berbasis *Web* Pada SMP N 1 Muaro Jambi

Penelitian ini bertujuan untuk merancang sistem informasi akademik berbasis *web* di SMP N 1 Muaro Jambi guna menggantikan metode pengolahan data akademik yang masih menggunakan cara konvensional dan aplikasi Microsoft Excel yang belum terintegrasi, sehingga menyulitkan pencarian data dan memperlambat proses penyampaian informasi. Sistem ini dirancang menggunakan bahasa pemrograman PHP dengan basis data MySQL, dan dikembangkan dengan metode Waterfall. Sistem informasi akademik ini menyediakan fitur seperti *login* pengguna dan admin, pengelolaan data sekolah, guru, siswa, mata pelajaran, kelas, jadwal pelajaran, dan nilai. Hasil penelitian menunjukkan bahwa sistem ini mampu membantu mempermudah pengolahan data akademik dan penyampaian informasi secara cepat dan akurat, serta dapat diakses oleh siswa dan guru kapan saja dan di mana saja, sehingga mengatasi permasalahan yang terjadi di sekolah tersebut (Rahmadani et al., 2022).

2.2 Uraian Landasan Teori

Beberapa teori yang berkaitan dengan penelitian ini adalah sebagai berikut.

2.2.1 Pengertian Sistem Informasi Akademik

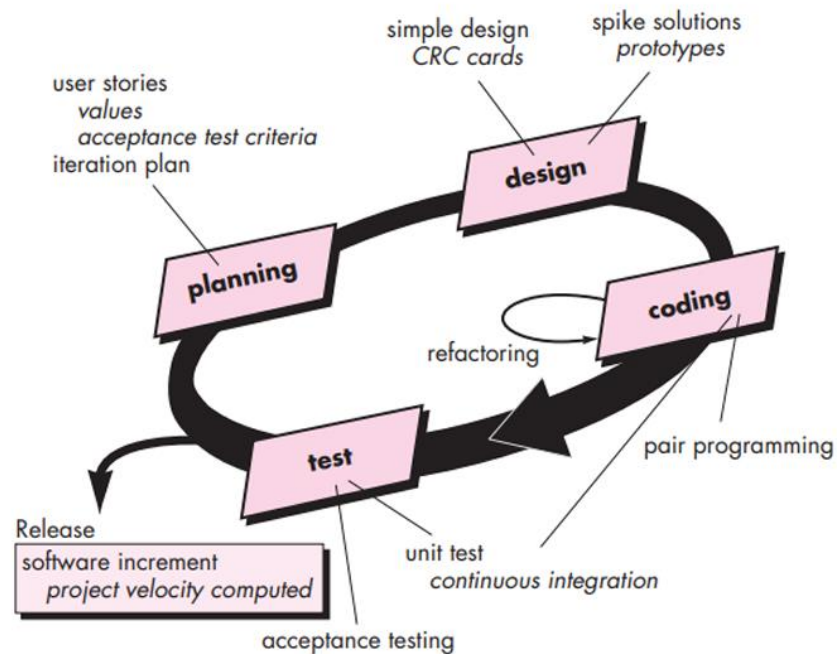
Menurut Muhamad Solahudin pada jurnalnya yang berjudul “Rancang Bangun Sistem Informasi Akademik Sekolah (SIAS) Berbasis *Website*” pada tahun 2021, mengatakan bahwa Sistem Informasi Akademik adalah salah-satu aplikasi yang dirancang untuk kebutuhan pengolahan data administratif sekolah dengan tujuan supaya data akademik lebih terkelola dengan baik dalam pengambilan keputusan di lingkungan sekolah (Muhamad Solahudin, 2021).

2.2.2 Pengertian PHP

Menurut Sitanggang Rianto dan kawan kawan pada tahun 2022 dalam jurnal penelitiannya yang berjudul “Rancang Bangun Sistem Penjualan Tanaman Hias Berbasis *Web* Menggunakan Php Dan Mysql” mengatakan bahwa, PHP, atau PHP *Hypertext Preprocessor*, adalah PHP merupakan script yang terintegrasi dengan HTML dan berada pada *server (serverside HTML embedded scripting)* (Sitanggang Rianto et al., 2022).

2.2.3 Pengertian Extreme Programming (XP)

Menurut Nurwahyuni Sari dan Dwi Cahyani pada tahun 2022 dalam jurnal nya yang berjudul “Perancangan Sistem Informasi *Monitoring* Sertifikat Menggunakan Extreme Programming” menyatakan bahwa, Metode Extreme Programming, atau sering disebut metode XP, adalah sebuah metodologi pengembangan perangkat lunak yang termasuk dalam kelompok Agile. XP menekankan fleksibilitas dalam menghadapi perubahan persyaratan, dengan menggunakan pendekatan iteratif (Sari & Cahyani, 2022). Adapun alur tahapan Extreme Programming dapat dilihat pada Gambar 1.



Gambar 1. Alur Tahapan Extreme Programming
(Kustiawan et al., 2022)

1. *Planning* (Perencanaan)

Tahap pertama adalah pertemuan antara *client* dengan tim *developer* guna presentasi gambaran hasil yang diinginkan oleh *client*. Tim kemudian memperkirakan gambaran yang diinginkan oleh *client* dan membuat rencana pengembangan yang dibagi menjadi beberapa tahapan yang diperlukan untuk memenuhi fungsionalitas yang dibutuhkan. Jika satu atau lebih dari gambaran *client* tidak dapat diperkirakan, atau biasa disebut dengan *spike*, maka diperlukan penelitian yang lebih lanjut.

2. *Design* (Perancangan)

Tahap *designing* sebenarnya merupakan bagian dari proses *planning* tetapi dapat dipisahkan untuk mengoptimalkan kedua proses ini. Hal ini terkait dengan salah satu nilai utama XP yaitu kesederhanaan. Desain yang baik membawa logika dan struktur ke dalam sistem

sehingga dapat menghindari kompleksitas dan redundansi yang tidak diperlukan.

3. *Coding* (Pengkodean)

Tahapan ini merupakan fase di mana kode dibuat dengan menerapkan praktik-praktik Extreme Programming tertentu seperti standar pengkodean, pemrograman berpasangan, integrasi berkelanjutan, dan kepemilikan kode kolektif.

4. *Testing* (Pengujian)

Tahap ini adalah inti *dari* Extreme Programming. Fase *testing* adalah aktivitas rutin yang melibatkan pengujian unit (pengujian otomatis untuk menentukan apakah fitur yang dikembangkan berfungsi dengan baik) dan pengujian penerimaan (pengujian pelanggan untuk memverifikasi bahwa keseluruhan sistem dibuat sesuai dengan persyaratan awal).

2.2.4 Pengertian Unified Modeling Language (UML)

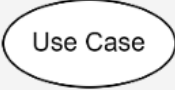
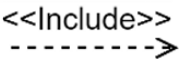
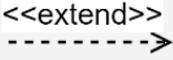
Unified Modeling Language (UML) adalah bahasa pemodelan visual tujuan umum yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sebuah sistem perangkat lunak. Bahasa ini menangkap keputusan dan pemahaman tentang sistem yang harus dibangun. UML digunakan untuk memahami, merancang, menjelajah, mengonfigurasi, memelihara, dan mengontrol informasi tentang sistem tersebut (Jacobson & Booch, 2021).

1. *Use Case Diagram*

Use case view menangkap perilaku dari sebuah sistem, subsistem, atau kelas sebagaimana yang terlihat oleh pengguna luar. *View* ini membagi fungsionalitas sistem menjadi transaksi-transaksi yang bermakna bagi aktor—pengguna ideal dari sebuah sistem. Potongan-

potongan fungsi interaktif ini disebut *use case*. *Use case* menggambarkan interaksi dengan aktor sebagai rangkaian pesan antara sistem dan satu atau lebih aktor. Istilah aktor mencakup manusia, serta sistem komputer dan proses lainnya (Jacobson & Booch, 2021). Adapun simbol-simbol *Use Case Diagram* disajikan pada Tabel 1.

Tabel 1. Simbol-simbol *Use Case Diagram*

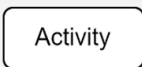
No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Aktor berperan untuk berinteraksi dengan sistem.
2		<i>Association</i>	<i>Association</i> merupakan relasi yang menghubungkan antara satu objek dengan objek lainnya.
3		<i>Use Case</i>	<i>Use Case</i> menggambarkan bagaimana aktor menggunakan sistem.
4		Relasi <i>Include</i>	Suatu relasi yang menghubungkan suatu <i>Use Case</i> dengan <i>Use Case</i> tambahan, dimana <i>Use Case</i> tambahan tersebut tidak dapat menjalankan fungsinya tanpa <i>Use Case</i> utama.
5		Relasi <i>Extend</i>	Relasi yang menghubungkan satu <i>Use Case</i> dengan <i>Use Case</i> tambahan, dimana <i>Use Case</i> tambahan tersebut dapat

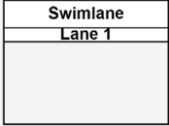
menjalankan fungsi meskipun tidak ada *Use Case* utama.

2. Activity Diagram

Activity diagram adalah notasi untuk grafik aktivitas. Diagram ini mencakup beberapa simbol singkatan khusus untuk kemudahan. Simbol-simbol ini sebenarnya bisa digunakan pada diagram *statechart* mana pun, meskipun pencampuran notasi mungkin terlihat kurang bagus sebagian besar waktu. *Activity state* ditampilkan sebagai kotak dengan ujung melengkung yang berisi deskripsi aktivitas. Transisi penyelesaian sederhana ditunjukkan dengan panah. Cabang-cabang ditunjukkan sebagai kondisi pengawal pada transisi atau sebagai berlian dengan beberapa panah keluar berlabel. *Fork* atau *join* kontrol ditampilkan sama seperti pada *statechart*, dengan beberapa panah yang masuk atau keluar melewati batang sinkronisasi yang berat (Jacobson & Booch, 2021). Adapun simbol-simbol *Activity Diagram* disajikan pada Tabel 2.

Tabel 2. Simbol-simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Initial State</i>	Indikasi awal dimulainya operasi yang dilakukan oleh sistem.
2		<i>Finale Node</i>	Indikasi akhir penyelesaian operasi yang dilakukan oleh sistem.
3		<i>Activity</i>	Simbol aktivitas yang dilakukan dalam sistem dan dimulai dengan kata kerja.

4		<i>Decision</i>	Simbol cabang dimana terdapat lebih dari satu pilihan.
5		<i>Swimlane</i>	Diagram alir yang menunjukkan objek melakukan aktivitas.

2.2.5 Pengertian *Class Diagram*

Menurut Jurnal yang berjudul “Penerapan UML *Class Diagram* dalam Perancangan Sistem Informasi Perpustakaan Berbasis *Web*” yang di karang oleh Sandy Wahyu Ramdany dan teman teman, mengatakan bahwa *Class diagram* merupakan hubungan antar kelas dan penjelasan detail tiap tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Jadi dapat dikatakan bahwa *Class Diagram* adalah visual dari struktur sistem program pada jenis-jenis yang di bentuk. *Class Diagram* merupakan alur jalannya sebuah database pada system yang akan dibangun atau dibuat. *Class diagram* juga disebut kumpulan dari beberapa *class* dan relasinya (Ramdany, 2024).

2.2.6 Pengertian CodeIgniter 4

Menurut Jurnal yang berjudul “Pengembangan Sistem Informasi Pelayanan Surat Menyurat (SIRA) Berbasis *Website* Dengan Menggunakan *Framework* Codeigniter” yang di karang oleh Ahmad Syaebani dan teman teman, mengatakan bahwa CodeIgniter adalah kerangka kerja pengembangan aplikasi PHP berdasarkan arsitektur yang terstruktur. CodeIgniter memiliki tujuan untuk memberikan alat bantu yang dibutuhkan seperti *helpers* and *libraries* untuk

mengimplementasi tugas yang biasa dilakukan. Dengan demikian, pengembangan proyek menjadi lebih mudah dan cepat. Dan pengembang tidak perlu menulis lagi dari awal. CodeIgniter adalah sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi PHP dinamis. Tujuan utama pengembangan CodeIgniter adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat daripada menulis semua kode dari awal. CodeIgniter menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan (Syaebani et al., 2021).

2.2.7. Pengertian User Acceptance Testing (UAT)

Menurut Jurnal yang berjudul “User Acceptance Test dan Penerapan Model ADDIE pada Media Pembelajaran Interaktif Pengenalan Profesi” yang di karang oleh Ari Nurul Alfian dan teman teman, mengatakan bahwa User Acceptance Testing adalah pengujian pada sistem setelah dilakukan pengembangan dengan cara melakukan pengujian oleh user yang dapat menghasilkan dokumen dan dapat dijadikan bukti dalam pengembangan aplikasi sesuai dengan hasil uji (Alfian et al., 2024).

Metode perhitungan yang digunakan untuk mengevaluasi hasil pengujian UAT dijelaskan dalam Persamaan dibawah.

$$P = \frac{S}{\text{Skor}} \times 100\% \dots\dots (\text{Chamida et al., 2021})$$

Keterangan:

P = Nilai presentase

S = Jumlah frekuensi dikalikan dengan skor jawaban

Skor = Skor tertinggi dikali dengan Jumlah ideal sampel

2.2.8. Black Box Testing

Metode Black Box Testing merupakan pengujian untuk menunjukkan kesalahan pada sistem aplikasi seperti kesalahan pada fungsi sistem

aplikasi, serta menu aplikasi yang hilang. Jadi Black Box testing merupakan metode uji fungsionalitas sistem aplikasi. Dalam melakukan pengujian menggunakan masukan data acak dengan tujuan untuk mendapatkan hasil yang pasti. Dikatakan pasti artinya bila salah, maka di tolak oleh sistem informasi atau data input tersebut tidak dapat disimpan dalam *database*, sedangkan bila data *input* benar maka dapat di terima / masuk di *database* sistem informasi (Uminingsih et al., 2022).

III. METODOLOGI PENELITIAN

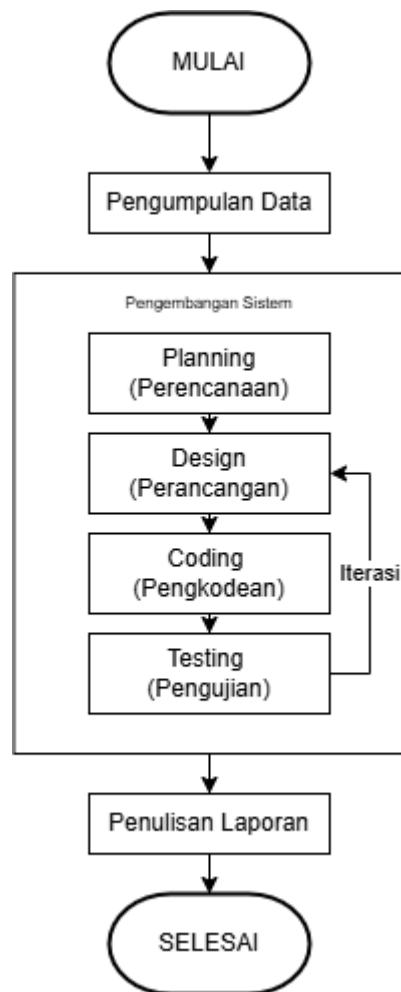
3.1. Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada semester ganjil tahun akademik 2024/2025 di Gedung Ilmu Komputer, Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung, serta di SMPN 29 Bandar Lampung yang berlokasi di Jl. Penghijauan No.50, Way Dadi, Kec. Sukarame, Kota Bandar Lampung.

3.2. Tahapan Penelitian

Proses penelitian ini terdiri dari tiga bagian utama, yaitu pengumpulan data, pengembangan sistem, dan penulisan laporan. Tahap pengumpulan data dilakukan untuk memperoleh informasi yang relevan sebagai dasar dalam merancang dan membangun sistem. Data dikumpulkan melalui observasi langsung, wawancara dengan pihak sekolah seperti guru dan staf administrasi, dan studi literatur. Informasi yang diperoleh kemudian dianalisis untuk merumuskan kebutuhan sistem yang akan dikembangkan. Setelah itu, proses dilanjutkan ke tahap pengembangan sistem yang menggunakan metode Extreme Programming (XP). Pada tahap ini, pengembangan sistem mencakup perancangan model proses bisnis, desain basis data, antarmuka pengguna, hingga implementasi kode program menggunakan teknologi *web*. Pengembangan dilakukan secara iteratif dan inkremental sesuai prinsip XP, dengan fokus pada kolaborasi intensif, pengujian berkelanjutan, dan respons cepat terhadap perubahan kebutuhan guna memastikan fungsionalitas dan kemudahan penggunaan sistem. Setelah sistem selesai dikembangkan dan diuji, tahap terakhir adalah penulisan laporan yang bertujuan untuk mendokumentasikan seluruh rangkaian kegiatan penelitian. Laporan ini disusun secara sistematis dan ilmiah,

mencakup latar belakang, metode penelitian, hasil pengembangan sistem, serta evaluasi kinerja sistem yang telah dibangun. Tahapan proses penelitian ini dapat dilihat pada Gambar 2.



Gambar 2. Tahapan Penelitian

3.2.1. Tahap Pengumpulan Data

Tahap pengumpulan data diperoleh melalui wawancara, studi pustaka, dan observasi. Pengumpulan data dilakukan untuk memahami proses alur kerja bisnis yang terjadi di SMPN 29 Bandar Lampung serta untuk memahami teori-teori yang diterapkan dalam penelitian ini. Tahapan pengumpulan data adalah sebagai berikut:

1. Wawancara

Metode wawancara digunakan untuk mengumpulkan informasi dari pihak-pihak terkait di SMPN 29 Bandar Lampung, seperti kepala sekolah, guru, dan staf tata usaha. Proses wawancara ini dilakukan untuk memahami bagaimana proses pengelolaan data akademik dan pembelajaran yang dilakukan di sekolah. Informasi yang diperoleh mencakup data tertulis maupun lisan, seperti pencatatan data siswa, pengelolaan nilai, jadwal pelajaran, serta kebutuhan dan kendala yang dihadapi dalam pengelolaan administrasi akademik. Hasil wawancara ini menjadi dasar dalam merancang dan mengembangkan sistem informasi akademik yang sesuai dengan kebutuhan sekolah.

2. Studi Pustaka

Metode studi pustaka dilakukan dengan mencari informasi dari berbagai sumber seperti buku, jurnal, artikel ilmiah, dan skripsi yang dijadikan referensi dalam penyusunan laporan penelitian. Sumber-sumber tersebut berkaitan dengan topik yang diteliti, yaitu pengembangan sistem informasi akademik. Studi pustaka ini bertujuan untuk memperoleh landasan teori yang relevan serta memperkaya pemahaman peneliti terkait konsep sistem informasi, manajemen data akademik, dan teknologi yang digunakan dalam pengembangan sistem.

3. Observasi

Metode observasi dilakukan dengan mengamati secara langsung aktivitas yang berlangsung di SMPN 29 Bandar Lampung. Observasi ini mencakup pengamatan terhadap proses pengelolaan data akademik dan pembelajaran, seperti pencatatan data siswa, pengisian nilai, penyusunan jadwal, serta interaksi antara guru, siswa, dan staf tata usaha dalam kegiatan administrasi akademik. Hasil observasi ini digunakan untuk memahami kebutuhan riil

sekolah dan menjadi dasar dalam merancang sistem informasi akademik yang sesuai.

3.2.2. Tahap Pengembangan Sistem

Tahap selanjutnya adalah pengembangan sistem dengan menggunakan metode Extreme Programming (XP). Tahapan yang dilakukan dalam metode Extreme Programming meliputi perencanaan, perancangan, pengkodean, dan pengujian.

1. *Planning* (Perencanaan)

A. User Stories, Acceptance Test Criteria dan Value

Dalam pengembangan sistem informasi akademik ini, kebutuhan pengguna dirumuskan dalam bentuk User Stories, yaitu deskripsi singkat tentang fitur yang diinginkan dari perspektif pengguna seperti Guru, Siswa, dan Admin. Tabel User Stories terdiri dari tiga kolom utama: User Stories yang menjelaskan kebutuhan pengguna, Acceptance Test Criteria yang menjadi standar untuk memastikan fitur sudah sesuai dan dapat diterima, serta Value yang menunjukkan prioritas pengerjaan berdasarkan tingkat kepentingannya, yang mana berada pada jarak 1 yang paling rendah hingga 10 yang paling tinggi. Dengan pengelolaan kebutuhan yang terstruktur ini, proses pengembangan dapat fokus pada fitur utama yang mendukung efektivitas dan kualitas sistem. User stories dapat dilihat pada tabel 3 dibawah.

Tabel 3. User Stories, Acceptance Test Criteria dan Value

No	User Stories	Acceptance Test Criteria	Value
----	--------------	-----------------------------	-------

1	Sebagai pengguna (Guru atau Admin), saya ingin dapat mengelola nilai siswa (menambah, mengubah, atau menghapus) di dalam sistem, agar saya bisa memperbarui nilai sesuai perkembangan akademik.	Guru/Admin dapat menambah, mengubah, atau menghapus nilai siswa, dan perubahan tersebut tercatat dengan benar di sistem.	9
2	Sebagai pengguna (Siswa, Guru, Admin), saya ingin dapat melihat nilai saya atau siswa di dalam sistem, agar saya bisa memantau perkembangan hasil belajar.	Sistem menampilkan nilai siswa dengan data yang akurat sesuai dengan yang terdaftar.	9
3	Sebagai pengguna (Admin), saya ingin dapat mengelola riwayat administrasi (menambah, mengubah, atau menghapus) di dalam sistem, agar saya bisa memastikan kelengkapan dan keakuratan data administratif.	Admin dapat menambah, mengubah, atau menghapus riwayat administrasi, dan perubahan tersebut tercatat dengan baik.	9
4	Sebagai pengguna (Siswa, Guru, Admin), saya ingin dapat melihat riwayat administrasi di dalam sistem, agar saya bisa mengetahui aktivitas administratif yang telah dilakukan.	Pengguna dapat melihat riwayat administrasi yang relevan sesuai peran mereka.	9
5	Sebagai pengguna (Admin), saya ingin dapat mengelola data siswa (menambah, mengubah, atau menghapus) di dalam sistem, agar saya bisa menjaga agar data siswa selalu akurat dan terbaru.	Admin dapat menambah, mengubah, atau menghapus data siswa, dan data yang ditampilkan selalu terbaru.	8
6	Sebagai pengguna (Siswa, Guru, Admin), saya ingin dapat melihat data siswa di	Pengguna dapat mengakses data siswa yang akurat dan relevan.	8

	dalam sistem, agar saya bisa mengetahui informasi terkait siswa.		
7	Sebagai pengguna (Admin), saya ingin dapat mengelola data guru (menambah, mengubah, atau menghapus) di dalam sistem, agar saya bisa menjaga agar data guru selalu terbaru.	Admin dapat menambah, mengubah, atau menghapus data guru, dan data guru yang ditampilkan selalu terbaru.	8
8	Sebagai pengguna (Siswa, Guru, Admin), saya ingin dapat melihat data guru di dalam sistem, agar saya bisa mengetahui informasi terkait guru sekolah.	Pengguna dapat melihat data guru yang relevan di dalam sistem.	8
9	Sebagai pengguna (Admin), saya ingin dapat mengelola data pegawai (menambah, mengubah, atau menghapus) di dalam sistem, agar saya bisa menjaga agar data pegawai selalu terbaru.	Admin dapat menambah, mengubah, atau menghapus data pegawai, dan data pegawai yang ditampilkan selalu terbaru.	8
10	Sebagai pengguna (Siswa, Guru, Admin), saya ingin dapat melihat data pegawai di dalam sistem, agar saya bisa mengetahui informasi terkait pegawai sekolah.	Pengguna dapat melihat data pegawai yang relevan di dalam sistem.	8
11	Sebagai pengguna (Admin), saya ingin dapat mengelola akun pengguna (membuat, mengaktifasi, memperbarui, atau menghapus akun), agar saya dapat mengatur akses pengguna terhadap sistem.	Admin dapat membuat, mengaktifkan, memperbarui, atau menghapus akun pengguna sesuai kebutuhan.	8
12	Sebagai pengguna (Admin), saya ingin dapat mengelola jadwal mata pelajaran	Admin dapat menambah, mengubah, atau	7

	(menambah, mengubah, atau menghapus) di dalam sistem, agar saya bisa memperbarui jadwal jika ada perubahan.	menghapus jadwal mata pelajaran, dan perubahan tersebut tercatat dengan baik.	
13	Sebagai pengguna (Siswa, Guru, Admin), saya ingin dapat melihat jadwal mata pelajaran di dalam sistem, agar saya bisa mengetahui mata pelajaran yang diajarkan.	Pengguna dapat melihat jadwal mata pelajaran yang relevan.	7
14	Sebagai pengguna (Siswa, Guru, Admin), saya ingin dapat melihat kemajuan belajar saya atau siswa di dalam sistem, agar saya bisa mengetahui sejauh mana perkembangan pembelajaran yang telah dicapai.	Pengguna dapat melihat kemajuan belajar yang relevan, termasuk nilai dan status perkembangan siswa.	7
15	Sebagai pengguna (Siswa, Guru, Admin), saya ingin dapat login menggunakan username dan password, agar saya bisa mengakses fitur dan informasi yang relevan sesuai dengan peran saya di sistem.	Pengguna dapat login dengan username dan password, dan dapat mengakses fitur sesuai dengan peran mereka.	6
16	Sebagai pengguna (Siswa, Guru, Admin), saya ingin dapat logout dari sistem, agar akun saya tetap aman setelah selesai menggunakan aplikasi.	Pengguna dapat logout dengan sukses dan tidak dapat mengakses fitur yang memerlukan login tanpa kredensial yang valid.	6

B. Iteration Plan

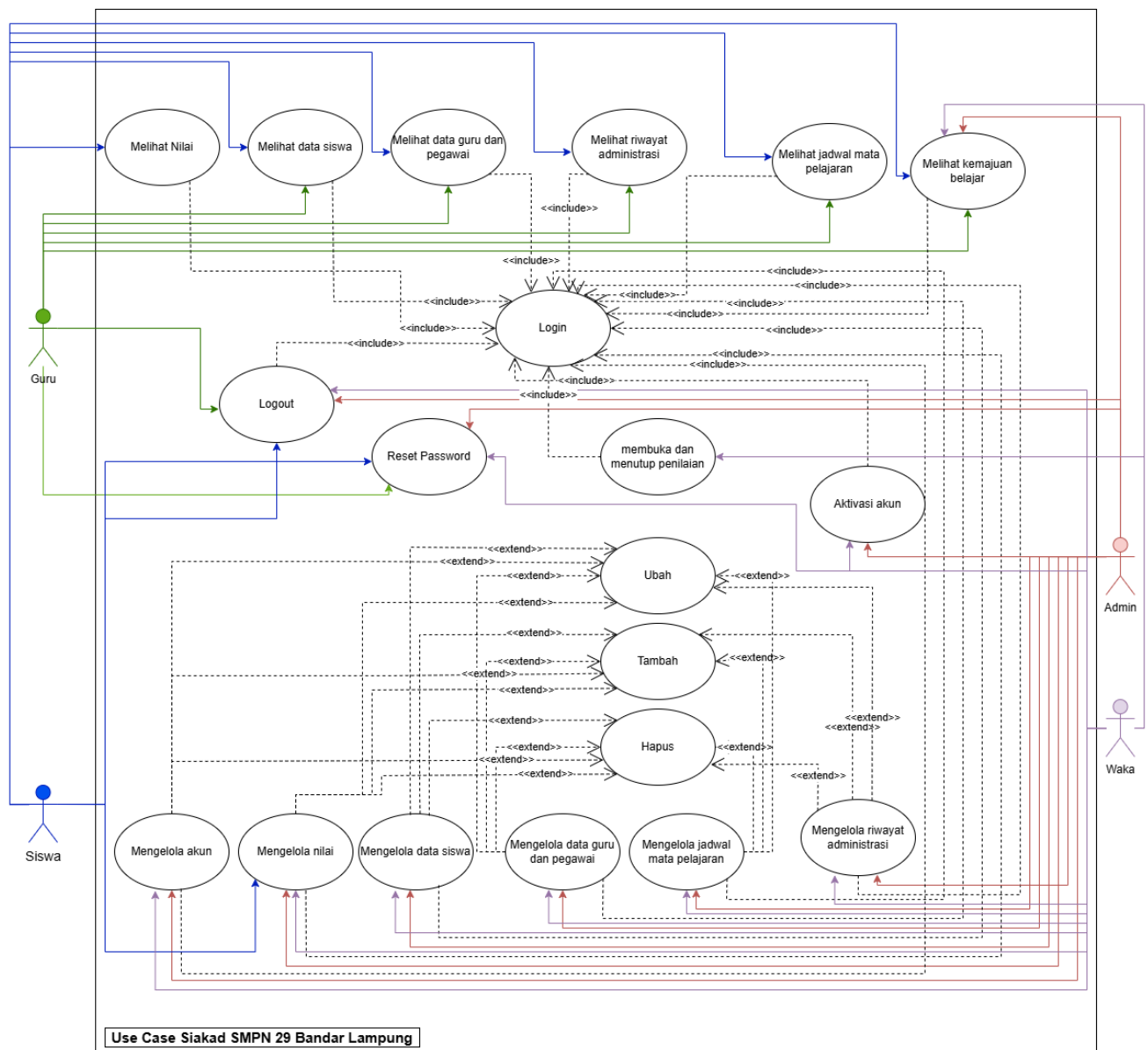
Pada tahap ini, Iteration Plan ditentukan berdasarkan User Stories yang telah dibuat sebelumnya dan diurutkan berdasarkan tingkat kepentingan mulai dari 1 paling rendah hingga 10 paling tinggi.

Tabel 4. Iteration Plan

Iterasi	Fungsional	Kriteria Penerimaan	Nilai
1	Dapat melihat detail nilai siswa	Berhasil menampilkan data nilai siswa	9
	Dapat menambah data nilai siswa	Berhasil menambah data nilai siswa	9
	Dapat mengubah data nilai siswa	Berhasil mengubah data nilai siswa	9
	Dapat menghapus data nilai siswa	Berhasil menghapus data nilai siswa	9
	Dapat melihat detail riwayat administrasi siswa	Berhasil menampilkan data riwayat administrasi siswa	9
	Dapat menambah data riwayat administrasi siswa	Berhasil menambah data riwayat administrasi siswa	9
	Dapat mengubah data riwayat administrasi siswa	Berhasil mengubah data riwayat administrasi siswa	9
	Dapat menghapus data riwayat administrasi siswa	Berhasil menghapus data riwayat administrasi siswa	9
	Dapat melihat data siswa	Berhasil menampilkan data siswa	8
	Dapat menambah data siswa	Berhasil menambah data siswa	8
	Dapat mengubah data siswa	Berhasil mengubah data siswa	8
	Dapat menghapus data siswa	Berhasil menghapus data siswa	8
	Dapat melihat data guru	Berhasil menampilkan data guru	8
	Dapat menambah data guru	Berhasil menambah data guru	8
	Dapat mengubah data guru	Berhasil mengubah data guru	8
	Dapat menghapus data guru	Berhasil menghapus data guru	8
	Dapat melihat data pegawai	Berhasil menampilkan pegawai	8

	Dapat menambah data pegawai	Berhasil menambah data pegawai	8
	Dapat mengubah data pegawai	Berhasil mengubah data pegawai	8
	Dapat menghapus data pegawai	Berhasil menghapus data pegawai	8
	Dapat melihat data akun	Berhasil menampilkan data akun	8
	Dapat menambah data akun	Berhasil menambah data akun	8
	Dapat mengubah data akun	Berhasil mengubah data akun	8
	Dapat menghapus data akun	Berhasil menghapus data akun	8
	Dapat mengaktifasi akun	Berhasil mengaktifasi akun	8
	Dapat melihat detail jadwal mata pelajaran	Berhasil menampilkan data jadwal mata pelajaran	7
	Dapat menambah data jadwal mata pelajaran	Berhasil menambah data jadwal mata pelajaran	7
	Dapat mengubah data jadwal mata pelajaran	Berhasil mengubah data jadwal mata pelajaran	7
	Dapat menghapus data jadwal mata pelajaran	Berhasil menghapus data jadwal mata pelajaran	7
	Dapat melihat detail kemajuan belajar siswa	Berhasil menampilkan data kemajuan belajar siswa	7
	Dapat login menggunakan username dan password	Berhasil masuk ke dalam sistem dan langsung menampilkan halaman informasi publik	6
	Dapat logout dari sistem	Berhasil keluar dari sistem dan langsung menampilkan halaman informasi publik	6

C. Use Case Diagram



Gambar 3. Use Case Diagram

Gambar 3 diatas adalah *Use case diagram* yang menggambarkan interaksi antara aktor utama, yaitu Guru, Siswa, Admin, dan Waka, dengan berbagai fungsi yang tersedia dalam Sistem Informasi Akademik (SIADKAD) SMPN 29 Bandar Lampung. Setiap pengguna diwajibkan melakukan proses login sebagai langkah awal untuk mengakses sistem, yang mencakup verifikasi

dan keamanan akses. Setelah berhasil *login*, Guru dan Siswa dapat mengakses berbagai fitur sesuai peran dan kewenangannya. Guru dapat melihat nilai siswa, data siswa, data guru dan pegawai, riwayat administrasi, serta jadwal mata pelajaran, yang semuanya saling terhubung dan bergantung pada proses autentikasi login. Admin memiliki peran yang lebih luas, tidak hanya dapat melakukan login dan logout, tetapi juga mengelola data secara komprehensif, termasuk pengelolaan akun, data siswa, data guru dan pegawai, nilai akademik, riwayat administrasi, dan jadwal mata pelajaran. Fungsi pengelolaan data meliputi operasi tambah, hapus, dan ubah data, yang dimodelkan sebagai use case yang diperluas (extend) dari use case pengelolaan utama. Admin juga bertugas melakukan aktivasi akun pengguna sebagai bagian dari pengelolaan akses sistem.

Waka, sebagai aktor dengan peran setara dengan Admin, memiliki akses yang sama dengan Admin, namun dengan tambahan kewenangan khusus untuk mengatur status penilaian, yaitu membuka atau menutup penilaian sesuai dengan kebutuhan sistem. Selain itu, use case reset password memungkinkan pengguna untuk melakukan pemulihan akun mereka jika lupa kata sandi, dan use case mengatur status penilaian memungkinkan Admin dan Waka untuk membuka atau menutup penilaian dalam sistem.

Diagram ini juga menunjukkan bahwa fungsi-fungsi seperti melihat kemajuan belajar siswa, mengelola nilai, riwayat administrasi, dan jadwal pelajaran saling terkait dan bergantung pada proses login. Selain itu, logout merupakan proses yang dapat dilakukan oleh semua aktor setelah menyelesaikan aktivitasnya di dalam sistem. Secara keseluruhan, use case diagram ini menggambarkan alur interaksi dan cakupan fungsi utama dalam

SIAKAD SMPN 29 Bandar Lampung, yang dirancang untuk mendukung pengelolaan data akademik dan administrasi secara efisien serta memberikan kemudahan akses bagi seluruh pengguna sesuai peran masing-masing. Penjabaran *Use Case* dapat dilihat pada Sub bab *Use Case Description* dan *Activity Diagram* pada tahapan *Design*

D. Ringkasan Eksekutif

Pada proyek ini, dirancang sebuah sistem informasi berbasis *website* untuk mendukung pengelolaan data akademik dan proses pembelajaran di SMPN 29 Bandar Lampung. Sistem ini terdiri dari empat modul utama, yaitu Modul Pembelajaran, Modul Evaluasi, Modul Data Akademik, dan Modul Informasi Publik. Sistem ini dirancang untuk mempermudah pengelolaan administrasi akademik, proses belajar-mengajar, serta penyebaran informasi kepada pihak-pihak terkait secara terintegrasi dan efisien.

Sistem ini dibangun menggunakan *Framework* CodeIgniter 4, yang menawarkan performa optimal dan fleksibilitas tinggi untuk pengembangan aplikasi berbasis *web*. Metodologi *Extreme Programming* (XP) diterapkan dalam pengembangan untuk memastikan fleksibilitas terhadap perubahan kebutuhan, pengujian berkelanjutan, serta kolaborasi erat antara tim pengembang dan pengguna. Dengan adanya sistem ini, SMPN 29 Bandar Lampung diharapkan dapat meningkatkan efisiensi pengelolaan informasi akademik, mendukung proses pembelajaran dan evaluasi yang lebih efektif, serta memberikan layanan informasi publik yang transparan dan mudah diakses.

E. Nilai Bisnis

Adapun nilai bisnis yang dapat diperoleh dengan adanya Sistem ini diuraikan menjadi 2 poin, yaitu Terukur dan Tidak Terukur.

- **Terukur:**

- Peningkatan Efisiensi: Sistem ini akan mengurangi waktu dan tenaga yang diperlukan dalam pengelolaan data akademik, pelaksanaan pembelajaran, evaluasi, dan penyampaian informasi, sehingga meningkatkan produktivitas guru dan staf administrasi.
- Keakuratan Data: Dengan sistem yang terintegrasi, risiko kesalahan dalam pengelolaan data akademik dan evaluasi dapat diminimalkan, sehingga memberikan keandalan yang lebih tinggi dalam pengambilan keputusan.
- Penghematan Biaya Operasional: Dengan digitalisasi proses, kebutuhan penggunaan kertas untuk administrasi dan laporan akademik dapat dikurangi, yang secara langsung menurunkan biaya operasional sekolah.

- **Tidak terukur:**

- Peningkatan Kepuasan Pengguna: Sistem yang efisien dan mudah digunakan akan meningkatkan pengalaman siswa, guru, dan orang tua dalam mengakses informasi akademik, materi pembelajaran, dan hasil evaluasi.
- Reputasi Sekolah: Penerapan teknologi modern akan meningkatkan citra SMPN 29 Bandar Lampung sebagai sekolah yang inovatif dan adaptif terhadap perkembangan teknologi di bidang pendidikan.
- Transparansi dan Keterbukaan Informasi: Dengan penyampaian informasi publik yang terintegrasi, orang tua dan masyarakat dapat lebih mudah memantau aktivitas dan

perkembangan siswa, yang berkontribusi pada hubungan yang lebih baik antara sekolah dan pihak-pihak terkait.

F. Sponsor Proyek

Sponsor proyek pada pengembangan sistem informasi ini adalah SMPN 29 Bandar Lampung, yang berlokasi di Jl. Soekarno Hatta No. 50, Bandar Lampung. Sebagai lembaga pendidikan tingkat menengah, SMPN 29 Bandar Lampung memiliki visi untuk meningkatkan mutu layanan pendidikan melalui penerapan teknologi informasi yang terintegrasi. Pihak sekolah, yang dipimpin oleh Kepala Sekolah, Bapak Samsuri, berperan aktif dalam memberikan arahan strategis, kebutuhan spesifik, serta dukungan penuh terhadap pengembangan sistem ini. Dengan sistem yang dirancang untuk mendukung pengelolaan data akademik, proses pembelajaran, evaluasi, dan informasi publik, SMPN 29 Bandar Lampung bertujuan untuk menjadi pelopor dalam transformasi digital di dunia pendidikan tingkat menengah.

Melalui proyek ini, SMPN 29 Bandar Lampung tidak hanya bertindak sebagai sponsor utama, tetapi juga sebagai mitra strategis dalam memastikan sistem yang dibangun sesuai dengan kebutuhan pengguna dan dapat memberikan dampak positif terhadap siswa, guru, serta masyarakat sekitar.

G. Kebutuhan Bisnis

Berdasarkan kebutuhan untuk meningkatkan efisiensi dan keandalan dalam pengelolaan informasi akademik dan pembelajaran di SMPN 29 Bandar Lampung, diperlukan sebuah sistem yang dapat mempermudah dan mengamankan proses pengelolaan data akademik, pembelajaran, evaluasi, serta penyampaian informasi publik. Sistem ini harus mampu

mendukung berbagai aktivitas akademik, seperti pengelolaan data siswa dan guru, pengaturan jadwal pembelajaran, evaluasi hasil belajar, serta penyebaran informasi yang efektif kepada siswa, orang tua, dan masyarakat. Dengan konektivitas yang lebih baik melalui *platform* berbasis *web*, sistem ini diharapkan dapat memberikan solusi digital yang handal untuk mendukung transformasi digital di lingkungan pendidikan SMPN 29 Bandar Lampung.

H. Persyaratan Bisnis

Kebutuhan bisnis pada pengembangan sistem informasi ini terdiri dari kebutuhan non fungsional dan kebutuhan fungsional. Kebutuhan non fungsional yaitu:

- 1) Sistem harus mampu berjalan secara *online* dengan baik melalui *website*
- 2) Sistem harus memiliki tampilan yang ramah pengguna
- 3) Sistem harus kompatibel dengan *browser* populer seperti Google Chrome, Mozilla Firefox, dan Microsoft Edge
- 4) Sistem harus dapat diakses dari berbagai perangkat, termasuk komputer, tablet, dan ponsel, melalui *browser* modern
- 5) Sistem harus dapat memuat halaman *website* kurang dari 10 detik
- 6) Sistem harus memiliki mekanisme *Login* seperti menggunakan *userName* dan *password*

Sedangkan kebutuhan fungsional adalah sebagai berikut:

- 1) User dapat mengakses nilai, jadwal mata pelajaran, data siswa, data guru dan pegawai, prestasi, riwayat administrasi, galeri kegiatan, ekstrakurikuler, sejarah singkat, hymne, visi misi, struktur organisasi, *ranking*, *e-raport*, dan kemajuan belajar

- 2) Admin dapat mengelola data siswa, data guru dan pegawai, nilai, jadwal mata pelajaran, riwayat administrasi, galeri kegiatan, prestasi, sejarah singkat, dan ekstrakurikuler
- 3) Guru dapat mengelola nilai siswa

I. Masalah Khusus / Kendala

Kendala yang dihadapi dalam pengembangan sistem informasi ini adalah:

- **Kompatibilitas:** Sistem harus dapat dijalankan di berbagai perangkat seperti komputer, smartphone dan tablet.
- **Keamanan:** Memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses fitur-fitur tertentu sesuai dengan peran mereka.
- **Estetika:** Sistem harus memiliki tampilan yang nyaman dilihat dan ramah pengguna untuk memastikan sistem dapat dimengerti semua kalangan

J. Persyaratan Nonfungsional

1) Persyaratan Operasional:

- Sistem harus dapat diakses secara *online* baik melalui *website*.
- Sistem harus dapat dijalankan di perangkat *smartphone*, tablet, maupun komputer.

2) Persyaratan Kinerja:

- Sistem harus mampu memproses penggunaan fitur, seperti akses data akademik, dan unggahan nilai pembelajaran, dalam waktu tidak lebih dari 30 detik per permintaan.

3) Persyaratan Keandalan:

- Sistem harus memiliki *uptime* 99.9% untuk memastikan ketersediaan layanan yang konsisten bagi pengguna.

4) **Persyaratan Keselamatan:**

- Enkripsi data harus diterapkan untuk semua data sensitif seperti *password* akun pengguna

5) **Persyaratan Keamanan:**

- memastikan bahwa setiap fitur hanya dapat diakses oleh pengguna yang memiliki peran dan otorisasi sesuai dengan hak akses yang telah ditentukan.

6) **Persyaratan Kegunaan:**

- Antarmuka pengguna harus intuitif dan mudah digunakan, dengan desain yang responsif untuk berbagai perangkat.

K. Persyaratan Fungsional

Sistem informasi yang dibangun memiliki kebutuhan fungsional sebagai berikut:

- **Penilaian:** Guru dapat menambah, menghapus, dan mengubah nilai siswa.
- **Manajemen Data Akademik dan Informasi Publik:** Admin dapat menambah, mengubah, dan menghapus data siswa, data guru dan pegawai, jadwal mata pelajaran, riwayat administrasi, galeri kegiatan, ekstrakurikuler, sejarah singkat, dan prestasi
- **Pembelajaran:** Pengguna dapat mengakses nilai, kemajuan belajar, data siswa, data guru dan pegawai, jadwal mata pelajaran, riwayat administrasi, informasi publik, *ranking*, dan *e-raport*

L. Hasil Kegiatan Wawancara

Berdasarkan hasil wawancara, ditemukan bahwa proses pengelolaan data akademik dan pembelajaran di SMPN 29 Bandar Lampung saat ini menghadapi beberapa kendala utama,

seperti kurangnya transparansi dalam penyampaian nilai kepada siswa dan orang tua, keterbatasan dalam penyebaran informasi akademik kepada orang tua secara efektif, serta keamanan penyimpanan data akademik yang belum optimal. Selain itu, proses pengelolaan data dan informasi masih membutuhkan waktu yang relatif lama, sehingga efisiensi kerja belum tercapai secara maksimal.

Pengguna sistem, termasuk guru, siswa, dan staf administrasi, mengharapkan adanya sistem yang dapat meningkatkan transparansi nilai, memungkinkan penyampaian informasi yang mudah diakses oleh orang tua, serta menjamin keamanan data akademik dari ancaman kehilangan atau kebocoran. Sistem juga diharapkan dapat mendukung efisiensi kerja melalui proses digitalisasi, seperti pengelolaan data siswa, pengolahan nilai, dan distribusi informasi secara terintegrasi.

M. Hasil Kegiatan Observasi

Berdasarkan hasil observasi yang dilakukan terhadap proses pengelolaan data akademik dan pembelajaran di SMPN 29 Bandar Lampung, ditemukan beberapa masalah yang signifikan baik dari sisi siswa, guru, maupun staf administrasi, yaitu:

1) Proses Pengelolaan Data dan Nilai Akademik:

- **Siswa/Orang Tua:** Kesulitan dalam mendapatkan informasi nilai akademik secara transparan dan cepat, serta terbatasnya akses untuk memantau perkembangan akademik siswa.
- **Guru/Admin:** Tantangan dalam mengelola dan memproses data nilai secara manual, yang memakan waktu dan rentan terhadap kesalahan input atau perhitungan.

2) Efisiensi Proses:

- **Guru/Admin:** Lamanya waktu yang dibutuhkan untuk mengolah data siswa, nilai, dan materi pembelajaran, yang mengurangi produktivitas dan efisiensi kerja.

3) Informasi Publik yang Mudah Diakses:

- **Siswa/Orang Tua:** Kesulitan dalam mencari informasi publik seperti daftar prestasi, atau berita sekolah, yang seharusnya mudah ditemukan di satu *platform* terintegrasi.
- **Guru/Admin:** Tantangan dalam menyediakan informasi publik secara terpusat dan efisien, yang menyebabkan informasi tidak selalu tersampaikan kepada pihak yang membutuhkan.

2. Design (Perancangan)

A. CRC Cards

USERS ✕ • Menyimpan data login pengguna sistem (email, username, password) • AUTH_GROUPS_USERS	AUTH_GROUPS ✕ • Menyimpan daftar grup/rol pengguna sistem • AUTH_GROUPS_USERS	AUTH_GROUPS_USERS ✕ • Menyimpan relasi many-to-many antara grup dan user • USERS
JADWAL ✕ • Menyimpan informasi jadwal pelajaran berdasarkan hari dan kelas	PEGAWAI ✕ • Menyimpan data pegawai sekolah termasuk jabatan dan status	GURU ✕ • Menyimpan data guru dan informasi kontak
SEJARAH ✕ • Menyimpan catatan sejarah kepemimpinan	PRESTASI ✕ • Menyimpan data prestasi yang diraih oleh siswa atau sekolah	EKSTRAKURIKULER ✕ • Menyimpan data kegiatan ekstrakurikuler
GALERIKEGIATAN ✕ • Menyimpan dokumentasi kegiatan sekolah	SISWA ✕ • Menyimpan data pribadi siswa • Menyimpan info kelas & semester saat ini • RIWAYAT_ADMINISTRASI • NILAI • RIWAYAT_NILAI • NILAI_AKHIR • RIWAYAT_NILAI_AKHIR	RIWAYAT_ADMINISTRASI ✕ • Menyimpan catatan pembayaran siswa • Mencatat tanggal dan nominal pembayaran • SISWA
NILAI ✕ • Menyimpan nilai siswa dari tiap penilaian (PH1-UAS) • Menghitung & menyimpan rata-rata nilai • SISWA	RIWAYAT_NILAI ✕ • Menyimpan nilai historis siswa dari semester sebelumnya • SISWA • NILAI	NILAI_AKHIR ✕ • Menyimpan nilai akhir siswa per semester & kelas • SISWA • NILAI
RIWAYAT_NILAI_AKHIR ✕ • Menyimpan data nilai akhir historis dari semester sebelumnya • SISWA • NILAI		

Gambar 4. CRC Cards

Gambar 4 Diatas adalah *CRC Cards* (Class-Responsibility-Collaborator Cards) yang menggambarkan tanggung jawab utama dan kolaborasi antar kelas dalam Sistem Informasi Akademik SMPN 29 Bandar Lampung. Setiap kartu merepresentasikan sebuah kelas yang berisi informasi mengenai fungsi utama (responsibility) yang diemban serta kelas lain yang menjadi kolaborator dalam menjalankan fungsi tersebut.

Kelas *Users* menyimpan data login pengguna sistem seperti email, username, dan password, serta berkolaborasi dengan kelas *AuthGroupsUsers* untuk mengelola relasi pengguna dengan grup atau peran tertentu. Kelas *AuthGroups* dan *AuthGroupsUsers* bertugas mengatur daftar grup atau peran pengguna dan relasinya dengan pengguna itu sendiri. Kelas *Jadwal* menyimpan informasi jadwal pelajaran berdasarkan hari dan kelas, berperan penting dalam pengelolaan aktivitas belajar mengajar.

Kelas *Pegawai* dan *Guru* masing-masing menyimpan data pegawai dan guru termasuk identitas serta kontak mereka. Kelas *Sejarah* mencatat catatan sejarah kepemimpinan sekolah, sementara kelas *Prestasi* menyimpan data prestasi yang diraih oleh siswa atau sekolah. Kelas *Ekstrakurikuler* bertugas menyimpan data kegiatan ekstrakurikuler yang diadakan oleh sekolah, sedangkan kelas *GaleriKegiatan* mendokumentasikan berbagai kegiatan sekolah dalam bentuk foto dan deskripsi.

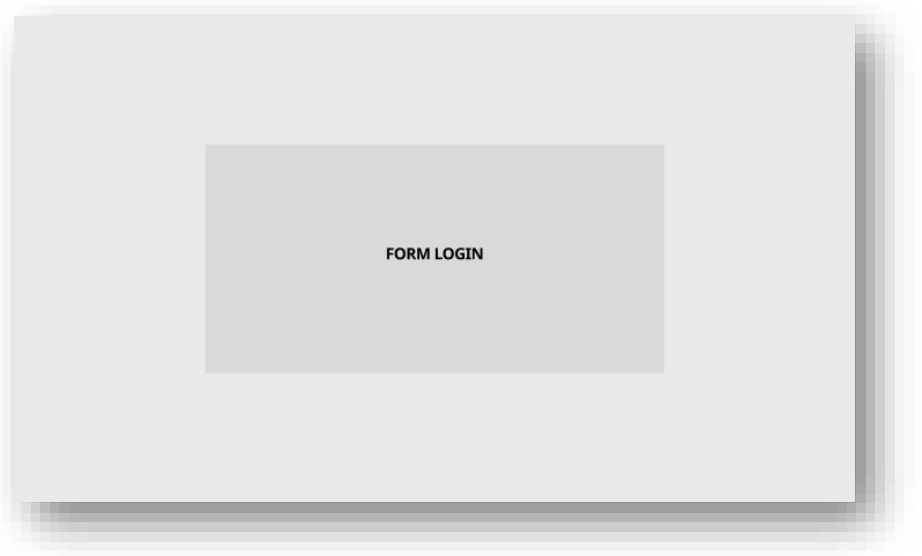
Kelas *Siswa* merupakan pusat data yang menyimpan data pribadi siswa serta informasi kelas dan semester saat ini, sekaligus berkolaborasi dengan beberapa kelas pendukung seperti *RiwayatAdministrasi* yang menyimpan catatan pembayaran siswa, *Nilai* yang menyimpan nilai per penilaian siswa, *RiwayatNilai* yang menyimpan nilai historis siswa dari semester

sebelumnya, serta NilaiAkhir dan RiwayatNilaiAkhir yang mencatat nilai akhir siswa per semester dan kelas secara historis.

Setiap kelas pada CRC *Cards* ini memiliki tanggung jawab jelas yang mendukung fungsi sistem secara keseluruhan, serta kolaborasi dengan kelas lain untuk memastikan integrasi data yang baik dan pengelolaan informasi akademik yang efektif. Struktur ini membantu memastikan bahwa pengembangan sistem berjalan dengan fokus pada modularitas dan keterhubungan antar fungsi utama dalam SIAKAD SMPN 29 Bandar Lampung.

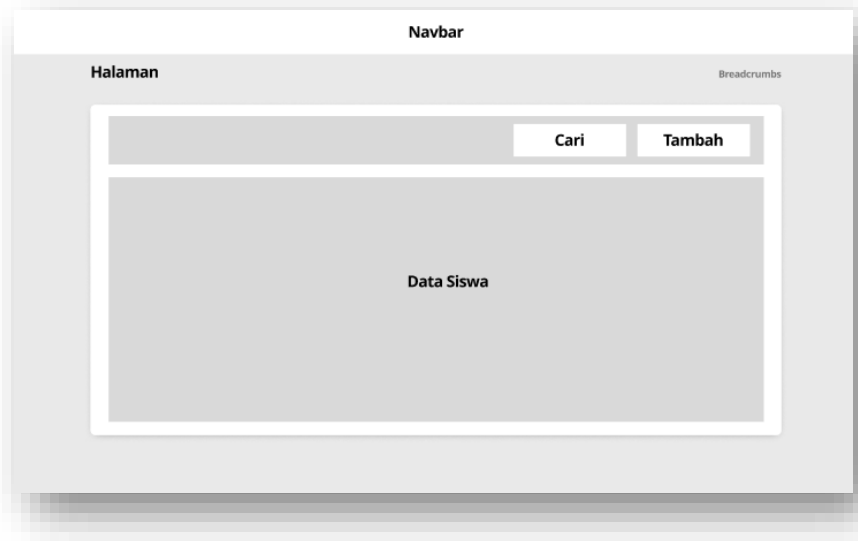
B. *Prototype*

1) Halaman *Login*



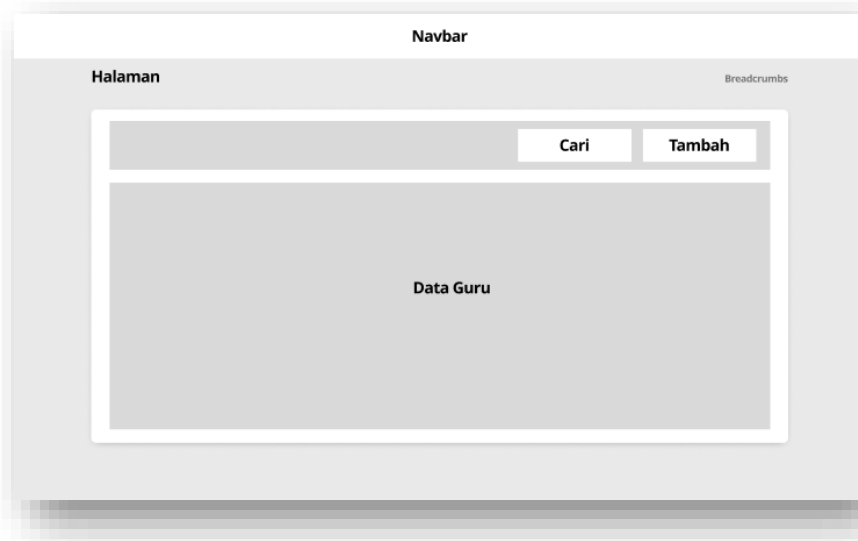
Gambar 5. Halaman Login

2) Halaman Data Siswa



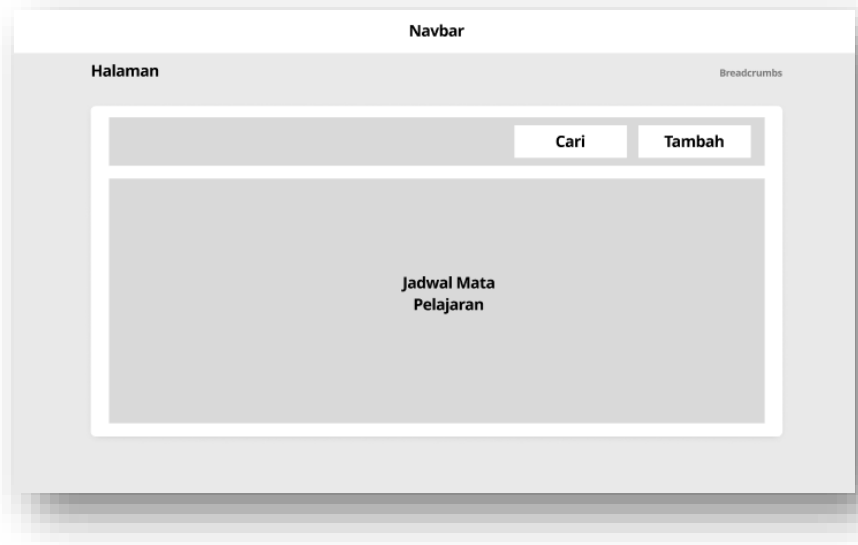
Gambar 6. Halaman Data Siswa

3) Halaman Data Guru dan Pegawai



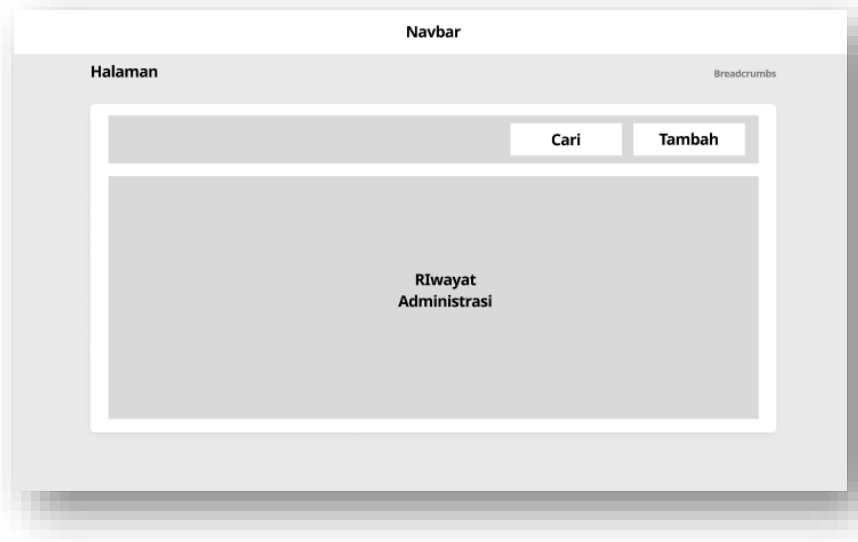
Gambar 7. Halaman Data Guru dan Pegawai

4) Halaman Jadwal Mata Pelajaran



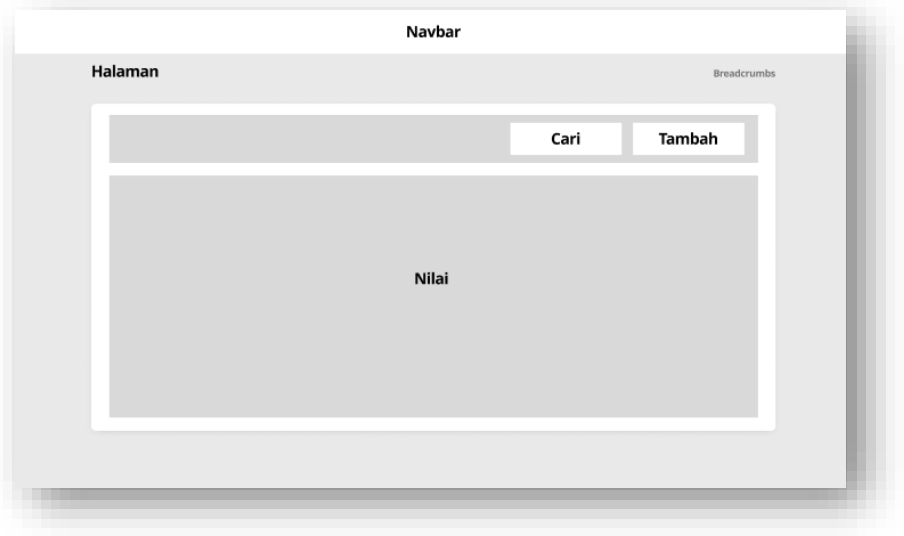
Gambar 8. Halaman Jadwal Mata Pelajaran

5) Halaman Riwayat Administrasi



Gambar 9. Halaman Riwayat Adminstrasi

6) Halaman Nilai



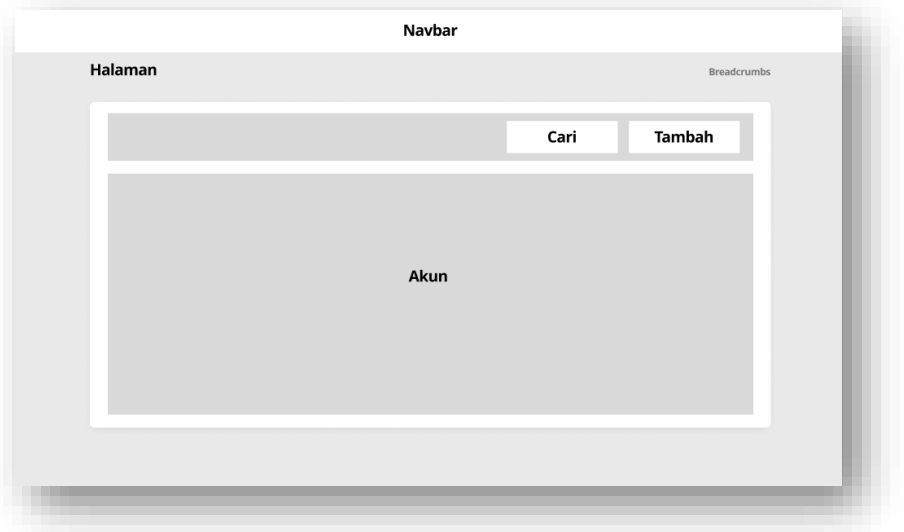
Gambar 10. Halaman Nilai

7) Halaman Kemajuan Belajar



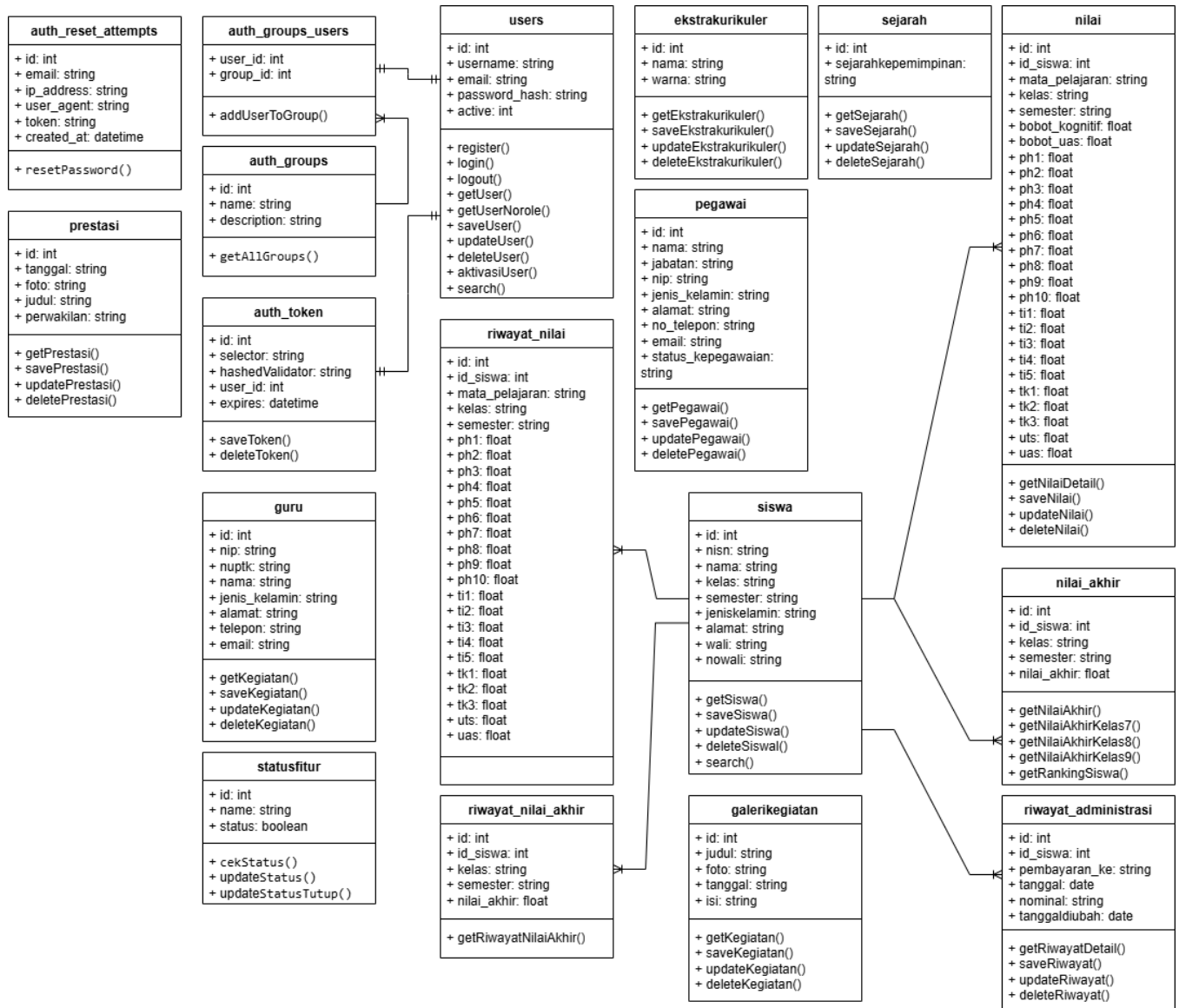
Gambar 11. Halaman Kemajuan Belajar

8) Halaman Manajemen Akun



Gambar 12. Halaman Manajemen Akun

C. Class Diagram



Gambar 13. Class Diagram

Gambar 13 di atas adalah *Class Diagram* menggambarkan struktur dan hubungan antar entitas utama dalam Sistem Informasi Akademik (SIADAK) SMPN 29 Bandar Lampung. Kelas *users* berfungsi sebagai pengelola akun pengguna yang mencakup siswa, guru, dan pegawai, dengan fitur registrasi,

login, logout, serta manajemen data pengguna termasuk aktivasi akun dan pencarian. Kelas siswa menjadi pusat data utama yang menyimpan informasi penting siswa seperti NISN, kelas, semester, serta kontak wali, dan berelasi dengan kelas nilai, nilai_akhir, riwayat_nilai, dan riwayat_administrasi untuk mengelola data nilai akademik serta administrasi keuangan siswa secara terintegrasi. Kelas nilai mengelola data nilai detail berdasarkan mata pelajaran dan semester, sedangkan kelas nilai_akhir dan riwayat_nilai_akhir mencatat nilai akhir dan peringkat siswa pada tiap periode. Kelas riwayat_nilai menyimpan nilai historis siswa dari semester sebelumnya secara rinci, sehingga mendukung pelacakan perkembangan akademik siswa dari waktu ke waktu. Kelas riwayat_administrasi merekam catatan pembayaran dan administrasi siswa yang berkaitan dengan aktivitas finansial di sekolah. Sementara itu, kelas pegawai dan guru menyimpan data tenaga kependidikan dan pendidik yang lengkap dengan atribut NIP, NUPTK, nama, jenis kelamin, alamat, telepon, dan email, serta didukung metode pengelolaan data secara lengkap.

Kelas pendukung seperti ekstrakurikuler, sejarah, prestasi, dan galerikegiatan menambah fungsi dokumentasi dan pengelolaan aktivitas sekolah. Kelas auth_groups menyimpan daftar grup atau role pengguna dalam sistem, sedangkan kelas auth_groups_users berfungsi sebagai penghubung many-to-many antara users dan auth_groups, memungkinkan pengelolaan peran atau hak akses yang fleksibel pada sistem. Relasi antar kelas mencerminkan kebutuhan integrasi data dalam pengelolaan akademik SMPN 29 Bandar Lampung, memastikan sistem dapat berjalan dengan efisien dan menyeluruh. Dengan metode CRUD yang lengkap, sistem ini dapat mengelola seluruh data akademik dan

administratif secara efektif, mendukung layanan pendidikan yang lebih transparan dan terstruktur di lingkungan sekolah.

Kelas `auth_token` berfungsi untuk menyimpan dan mengelola token autentikasi yang digunakan untuk validasi sesi pengguna. Kelas ini memiliki metode untuk menyimpan token (`saveToken`) dan menghapus token (`deleteToken`). Kelas `auth_reset_attempts` menyimpan data tentang upaya reset password yang dilakukan oleh pengguna. Kelas ini memiliki atribut seperti email, IP address, user agent, dan token yang terkait dengan upaya reset password, serta metode untuk mereset password (`resetPassword`).

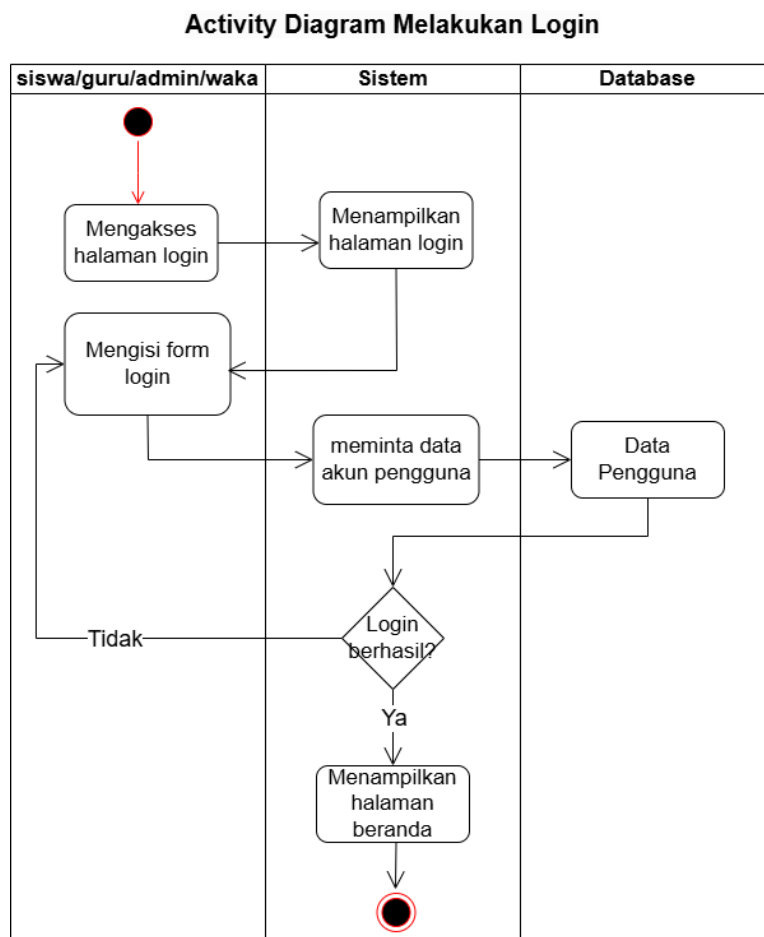
D. *Activity Diagram*

Selanjutnya dibuat *Activity Diagram* berdasarkan *Use Case* yang telah dibuat sebelumnya. Adapun *Activity Diagram* adalah sebagai berikut

1) *Activity Diagram* Melakukan *Login*

Activity Diagram pada Gambar 14 menjelaskan alur proses *Login* bagi pengguna yang terdiri dari siswa, guru, atau admin. Proses dimulai ketika pengguna mengakses halaman *Login*, kemudian sistem menampilkan halaman *Login* yang memungkinkan pengguna untuk mengisi *form Login* dengan kredensial mereka. Setelah *form Login* diisi, sistem akan meminta data akun pengguna dari *database* untuk melakukan proses verifikasi. Jika data yang dimasukkan sesuai dengan yang tersimpan di *database*, maka *Login* berhasil dan sistem akan menampilkan halaman beranda. Namun, jika *Login* gagal, pengguna harus kembali mengisi *form Login* untuk mencoba kembali. Diagram ini menggambarkan interaksi

antara pengguna, sistem, dan *database* dalam proses autentikasi *Login*.

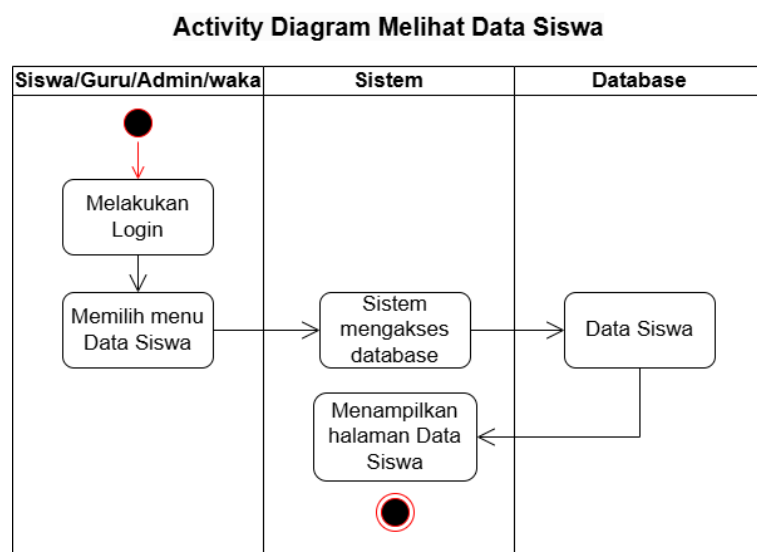


Gambar 14. *Activity Diagram* Melakukan Login

2) *Activity Diagram* Melihat Data Siswa

Activity Diagram pada Gambar 15 menjelaskan alur proses pengguna dalam melihat data siswa. Proses dimulai ketika pengguna, yang dapat berupa siswa, guru, atau admin, terlebih dahulu melakukan Login ke sistem. Setelah berhasil Login, pengguna memilih menu data siswa. Selanjutnya, sistem akan mengakses database untuk mengambil data siswa yang tersimpan. Setelah data berhasil diperoleh, sistem akan

menampilkan halaman yang berisi informasi data siswa kepada pengguna. Diagram ini menunjukkan interaksi antara pengguna, sistem, dan database dalam menampilkan data siswa secara dinamis, dengan tahap Login sebagai langkah awal sebelum mengakses informasi tersebut.

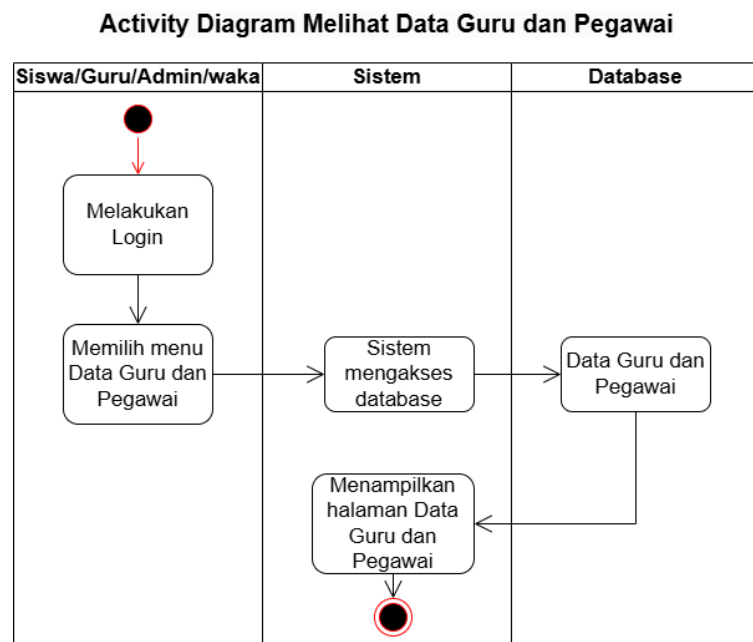


Gambar 15. *Activity Diagram* Melihat Data Siswa

3) *Activity Diagram* Melihat Data Guru dan Pegawai

Activity Diagram pada Gambar 16 menjelaskan alur proses pengguna dalam melihat data guru dan pegawai. Proses dimulai ketika pengguna, yang dapat berupa siswa, guru, atau admin, terlebih dahulu melakukan Login ke sistem. Setelah berhasil Login, pengguna memilih menu data guru dan pegawai. Selanjutnya, sistem akan mengakses database untuk mengambil data guru dan pegawai yang tersimpan. Setelah data berhasil diperoleh, sistem akan menampilkan halaman yang berisi informasi mengenai data guru dan pegawai kepada pengguna. Diagram ini menunjukkan interaksi antara pengguna, sistem, dan database dalam menampilkan informasi

data guru dan pegawai secara dinamis, dengan tahap Login sebagai langkah awal sebelum mengakses data tersebut.

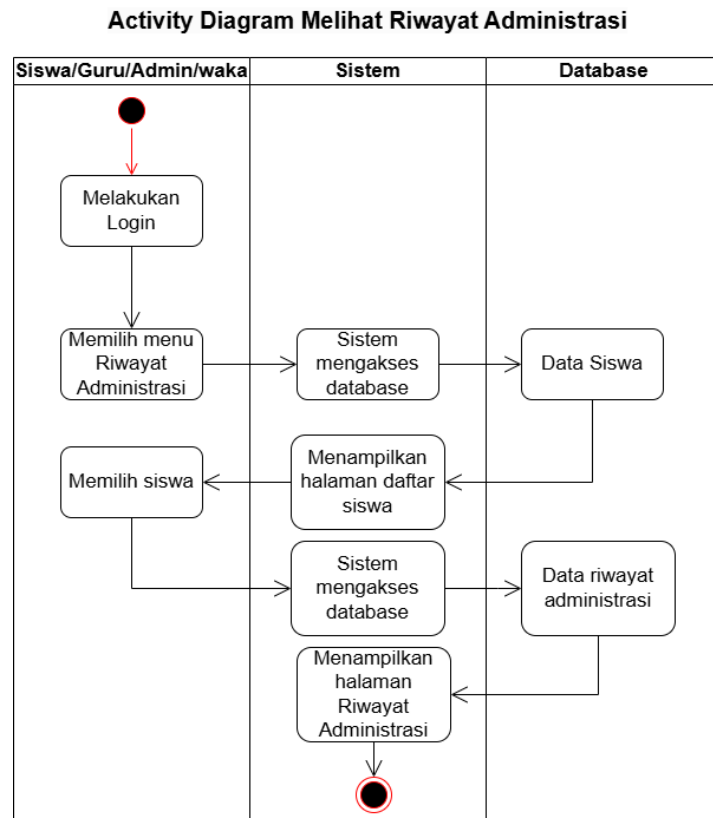


Gambar 16. *Activity Diagram* Melihat Data Guru dan Pegawai

4) *Activity Diagram* Melihat Riwayat Administrasi

Activity Diagram pada Gambar 17 menjelaskan alur proses pengguna dalam melihat riwayat administrasi. Proses dimulai ketika pengguna, yang dapat berupa siswa, guru, atau admin, terlebih dahulu melakukan Login ke sistem. Setelah berhasil Login, pengguna memilih menu riwayat administrasi, kemudian sistem mengakses database untuk mengambil data siswa dan menampilkan halaman daftar siswa. Pengguna kemudian memilih salah satu siswa yang ingin dilihat riwayat administrasinya. Setelah itu, sistem kembali mengakses database untuk mengambil data riwayat administrasi siswa yang dipilih. Setelah data berhasil diperoleh, sistem akan menampilkan halaman yang berisi riwayat administrasi siswa

tersebut. Diagram ini menunjukkan interaksi antara pengguna, sistem, dan database dalam menampilkan informasi riwayat administrasi secara bertahap dan dinamis.

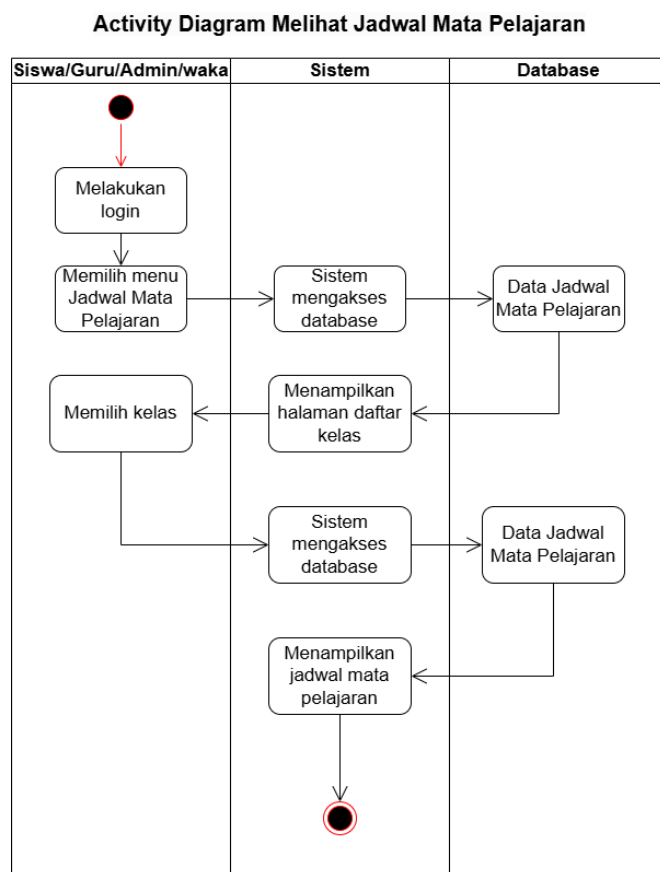


Gambar 17. *Activity Diagram* Melihat Riwayat Administrasi

5) *Activity Diagram* Melihat Jadwal Mata Pelajaran

Activity Diagram pada Gambar 18 menjelaskan alur proses pengguna dalam melihat jadwal mata pelajaran. Proses dimulai ketika pengguna, yang dapat berupa siswa, guru, atau admin, terlebih dahulu melakukan Login ke sistem. Setelah berhasil Login, pengguna memilih menu jadwal mata pelajaran. Selanjutnya, sistem akan mengakses database untuk mengambil data jadwal mata pelajaran yang tersimpan. Setelah data berhasil diperoleh, sistem akan menampilkan

halaman yang berisi informasi jadwal mata pelajaran kepada pengguna. Diagram ini menunjukkan interaksi antara pengguna, sistem, dan database dalam menampilkan jadwal mata pelajaran secara dinamis setelah proses autentikasi dilakukan.

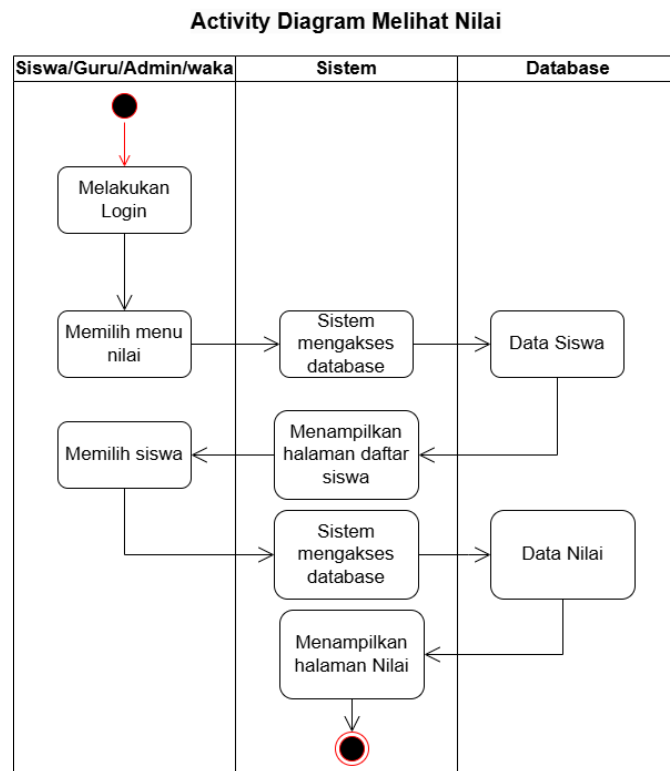


Gambar 18. *Activity Diagram* Melihat Jadwal Mata Pelajaran

6) *Activity Diagram* Melihat Nilai

Activity Diagram pada Gambar 19 menjelaskan alur proses pengguna dalam melihat nilai. Proses dimulai ketika pengguna, yang dapat berupa siswa, guru, atau admin, terlebih dahulu melakukan Login ke sistem. Setelah berhasil Login, pengguna memilih menu nilai. Selanjutnya, sistem akan

mengakses database untuk mengambil data siswa dan menampilkan halaman daftar siswa. Pengguna kemudian memilih salah satu siswa yang ingin dilihat nilainya. Setelah itu, sistem kembali mengakses database untuk mengambil data nilai siswa yang dipilih. Setelah data berhasil diperoleh, sistem akan menampilkan halaman yang berisi informasi nilai siswa tersebut. Diagram ini menunjukkan interaksi antara pengguna, sistem, dan database dalam menampilkan informasi nilai secara bertahap dan dinamis setelah proses autentikasi dilakukan.

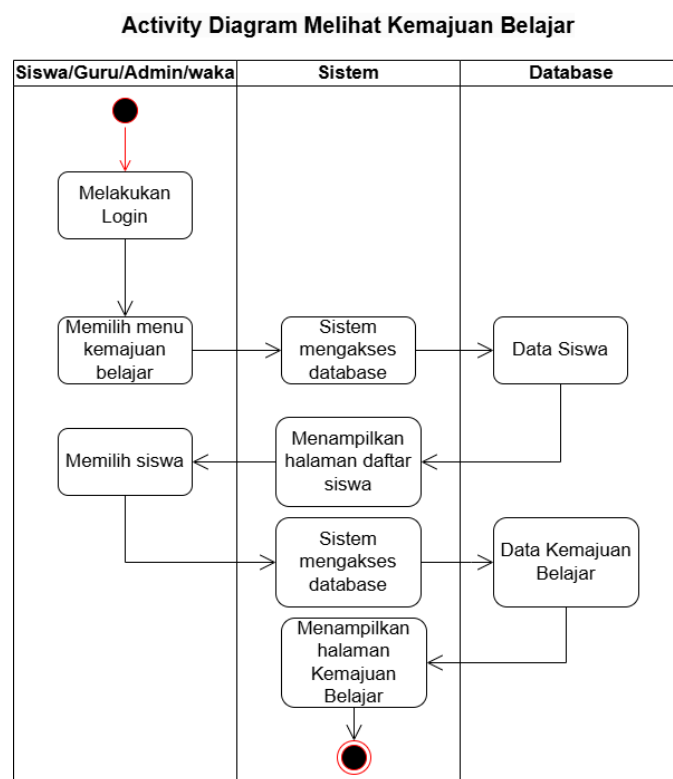


Gambar 19. *Activity Diagram Melihat Nilai*

7) *Activity Diagram Melihat Kemajuan Belajar*

Activity Diagram pada Gambar 20 menjelaskan alur proses pengguna dalam melihat kemajuan belajar. Proses dimulai

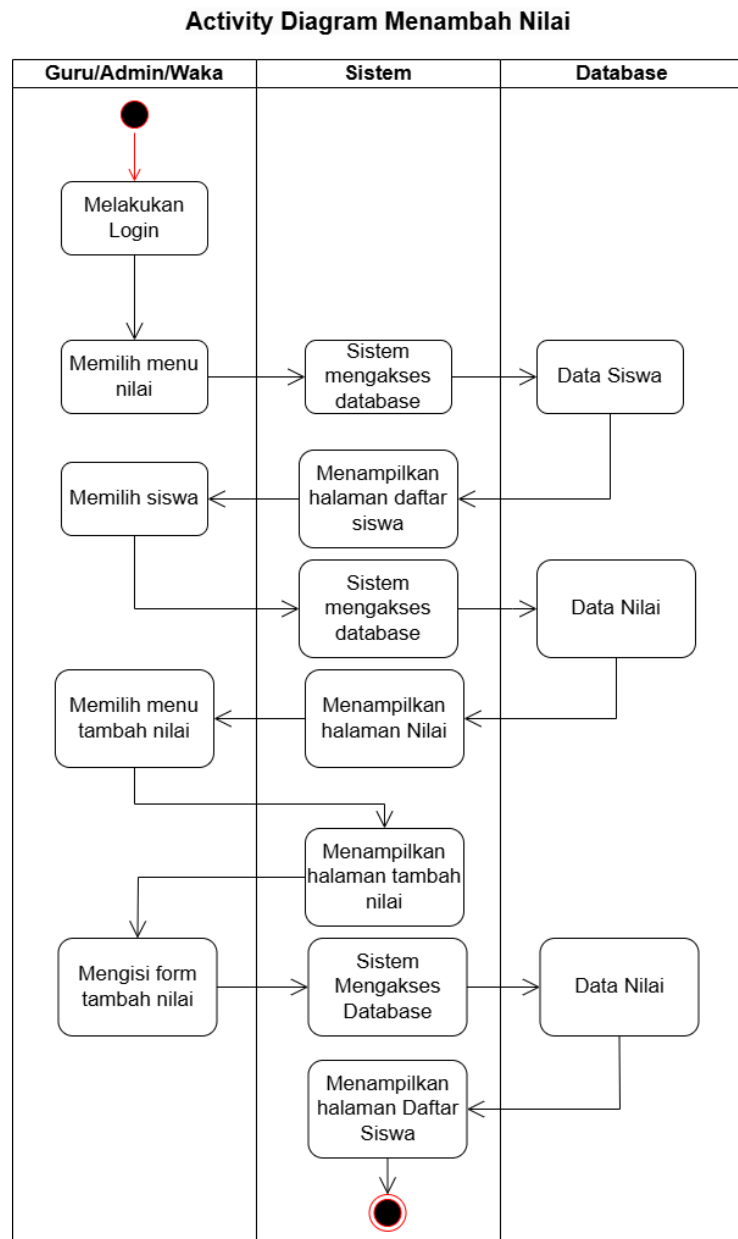
ketika pengguna, yang dapat berupa siswa, guru, atau admin, terlebih dahulu melakukan Login ke sistem. Setelah berhasil Login, pengguna memilih menu kemajuan belajar. Selanjutnya, sistem akan mengakses database untuk mengambil data siswa dan menampilkan halaman daftar siswa. Pengguna kemudian memilih salah satu siswa yang ingin dilihat kemajuan belajarnya. Setelah itu, sistem kembali mengakses database untuk mengambil data kemajuan belajar siswa yang dipilih. Setelah data berhasil diperoleh, sistem akan menampilkan halaman yang berisi informasi kemajuan belajar siswa tersebut. Diagram ini menunjukkan interaksi antara pengguna, sistem, dan database dalam menampilkan informasi kemajuan belajar secara bertahap dan dinamis setelah proses autentikasi dilakukan.



Gambar 20. *Activity Diagram* Melihat Kemajuan Belajar

8) *Activity Diagram* Menambah Nilai

Activity Diagram pada Gambar 21 menjelaskan alur proses guru atau admin dalam menambahkan nilai siswa. Proses dimulai ketika pengguna melakukan Login ke sistem. Setelah berhasil Login, pengguna memilih menu nilai, kemudian sistem mengakses database untuk mengambil data siswa dan menampilkan halaman daftar siswa. Selanjutnya, pengguna memilih salah satu siswa yang ingin ditambahkan nilainya. Setelah itu, sistem kembali mengakses database untuk mengambil data nilai siswa tersebut dan menampilkan halaman nilai. Pengguna kemudian memilih menu tambah nilai, yang menyebabkan sistem menampilkan halaman tambah nilai. Guru atau admin mengisi form tambah nilai sesuai dengan data yang ingin dimasukkan. Setelah form diisi, sistem akan mengakses database untuk menyimpan data nilai baru. Setelah data berhasil disimpan, sistem menampilkan halaman nilai yang telah diperbarui. Diagram ini menunjukkan interaksi antara pengguna, sistem, dan database dalam menambahkan nilai siswa secara bertahap dan dinamis setelah proses autentikasi dilakukan.

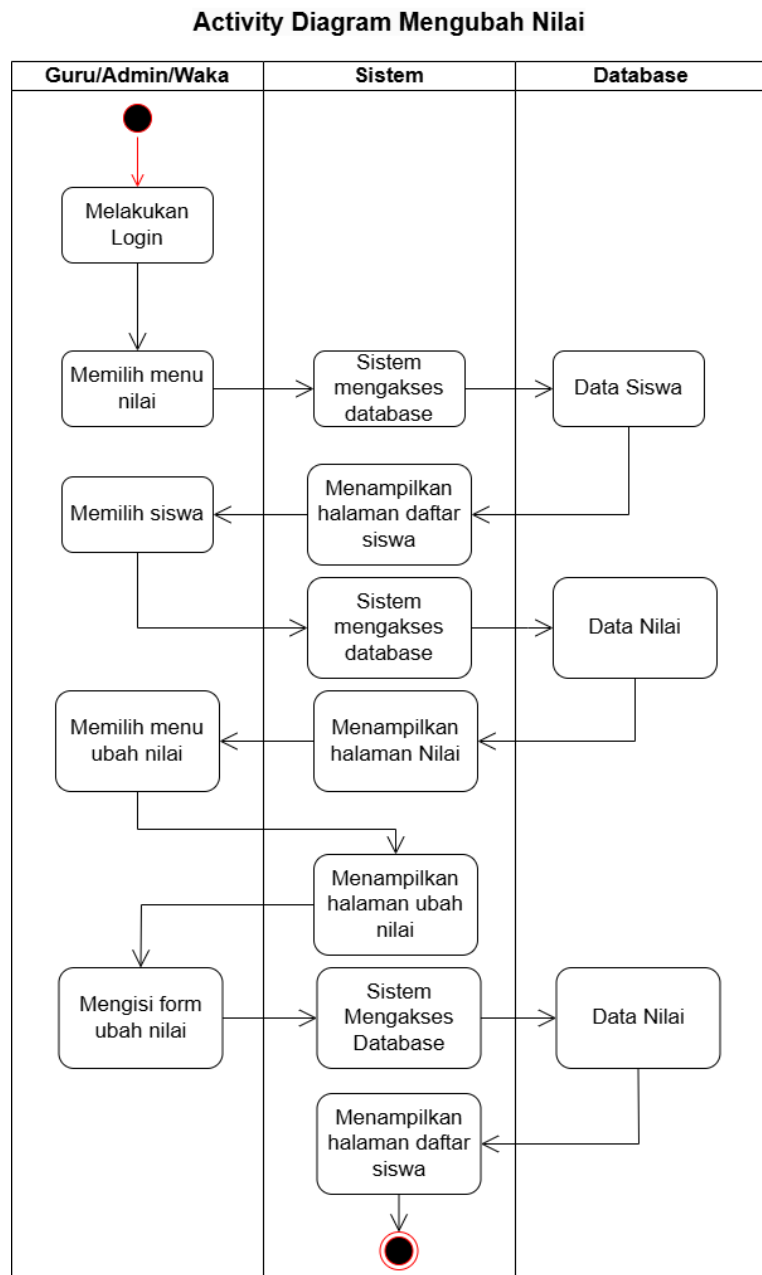


Gambar 21. *Activity Diagram Menambah Nilai*

9) *Activity Diagram Mengubah Nilai*

Activity Diagram pada Gambar 22 menjelaskan alur proses guru atau admin dalam mengubah nilai siswa. Proses dimulai ketika pengguna melakukan Login ke sistem. Setelah berhasil Login, pengguna memilih menu nilai, kemudian sistem

mengakses database untuk mengambil data siswa dan menampilkan halaman daftar siswa. Selanjutnya, pengguna memilih salah satu siswa yang ingin diubah nilainya. Setelah itu, sistem kembali mengakses database untuk mengambil data nilai siswa tersebut dan menampilkan halaman nilai. Pengguna kemudian memilih menu ubah nilai, sehingga sistem menampilkan halaman ubah nilai. Guru atau admin mengisi form ubah nilai dengan data yang diperbarui. Setelah form diisi, sistem akan mengakses database untuk memperbarui data nilai yang tersimpan. Setelah data berhasil diperbarui, sistem menampilkan halaman nilai yang telah diperbarui. Diagram ini menunjukkan interaksi antara pengguna, sistem, dan database dalam proses pengubahan nilai siswa secara bertahap dan dinamis setelah proses autentikasi dilakukan.

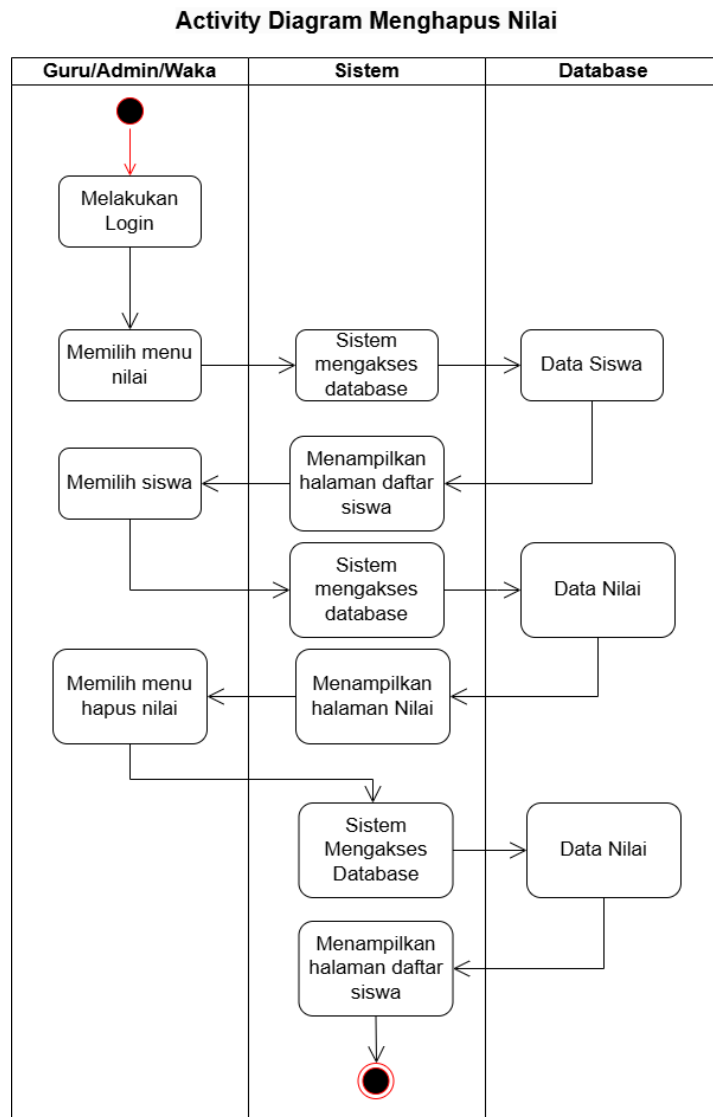


Gambar 22. Activity Diagram Mengubah Nilai

10) Activity Diagram Menghapus Nilai

Activity Diagram pada Gambar 23 menjelaskan alur proses guru atau admin dalam menghapus nilai siswa. Proses dimulai ketika pengguna melakukan Login ke sistem. Setelah berhasil Login, pengguna memilih menu nilai, kemudian sistem

mengakses database untuk mengambil data siswa dan menampilkan halaman daftar siswa. Selanjutnya, pengguna memilih salah satu siswa yang ingin dihapus nilainya. Setelah itu, sistem kembali mengakses database untuk mengambil data nilai siswa tersebut dan menampilkan halaman nilai. Pengguna kemudian memilih menu hapus nilai, sehingga sistem akan mengakses database untuk menghapus data nilai siswa yang dipilih. Setelah data berhasil dihapus, sistem menampilkan halaman nilai yang telah diperbarui tanpa data yang dihapus. Diagram ini menunjukkan interaksi antara pengguna, sistem, dan database dalam proses penghapusan nilai siswa secara bertahap dan dinamis setelah proses autentikasi dilakukan.

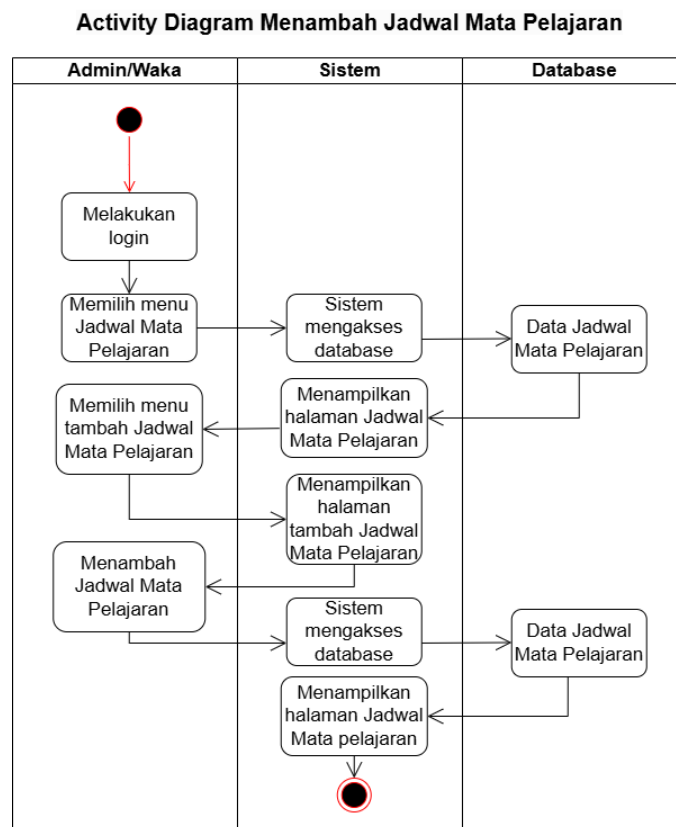


Gambar 23. *Activity Diagram* Menghapus Nilai

11) *Activity Diagram* Menambah Jadwal Mata Pelajaran

Activity Diagram pada Gambar 24 menjelaskan alur proses admin dalam menambahkan jadwal mata pelajaran. Proses dimulai ketika admin melakukan Login ke sistem. Setelah berhasil Login, admin memilih menu jadwal mata pelajaran. Selanjutnya, sistem akan mengakses database untuk mengambil data jadwal mata pelajaran yang sudah ada dan

menampilkan halaman jadwal mata pelajaran. Setelah itu, admin memilih menu tambah jadwal mata pelajaran, yang menyebabkan sistem menampilkan halaman tambah jadwal mata pelajaran. Admin kemudian mengisi informasi jadwal mata pelajaran yang baru dan menginputnya ke dalam sistem. Setelah data dimasukkan, sistem akan mengakses database untuk menyimpan jadwal baru. Setelah data berhasil disimpan, sistem menampilkan halaman jadwal mata pelajaran yang telah diperbarui dengan jadwal yang baru. Diagram ini menunjukkan interaksi antara admin, sistem, dan database dalam menambahkan jadwal mata pelajaran secara bertahap dan dinamis setelah proses autentikasi dilakukan.

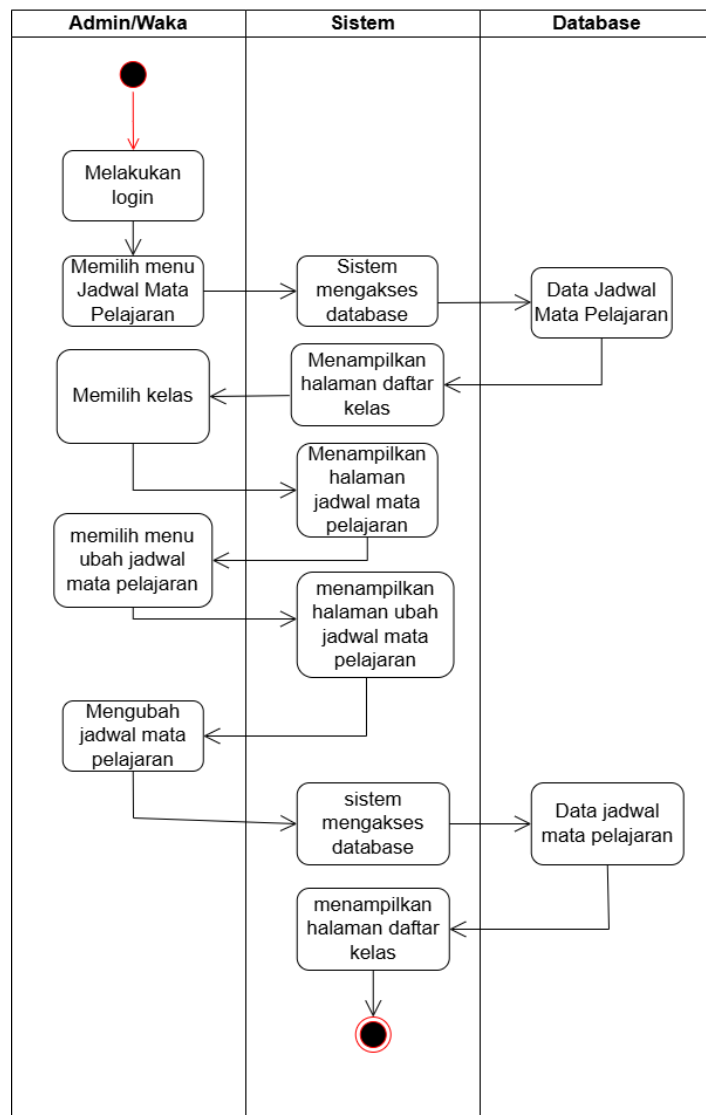


Gambar 24. *Activity Diagram Menambah Jadwal Mata Pelajaran*

12) *Activity Diagram* Mengubah Jadwal mata Pelajaran

Activity Diagram pada Gambar 25 menjelaskan alur proses admin dalam mengubah jadwal mata pelajaran. Proses dimulai ketika admin melakukan Login ke sistem. Setelah berhasil Login, admin memilih menu jadwal mata pelajaran. Selanjutnya, sistem akan mengakses database untuk mengambil data jadwal mata pelajaran yang sudah ada dan menampilkan halaman jadwal mata pelajaran. Setelah itu, admin memilih menu ubah jadwal mata pelajaran, yang menyebabkan sistem menampilkan halaman ubah jadwal mata pelajaran. Admin kemudian melakukan perubahan pada jadwal mata pelajaran sesuai kebutuhan dan menginput data yang diperbarui ke dalam sistem. Setelah perubahan dilakukan, sistem akan mengakses database untuk memperbarui data jadwal yang tersimpan. Setelah data berhasil diperbarui, sistem menampilkan halaman jadwal mata pelajaran yang telah diperbarui dengan informasi terbaru. Diagram ini menunjukkan interaksi antara admin, sistem, dan database dalam memperbarui jadwal mata pelajaran secara bertahap dan dinamis setelah proses autentikasi dilakukan.

Activity Diagram Mengubah Jadwal Mata Pelajaran

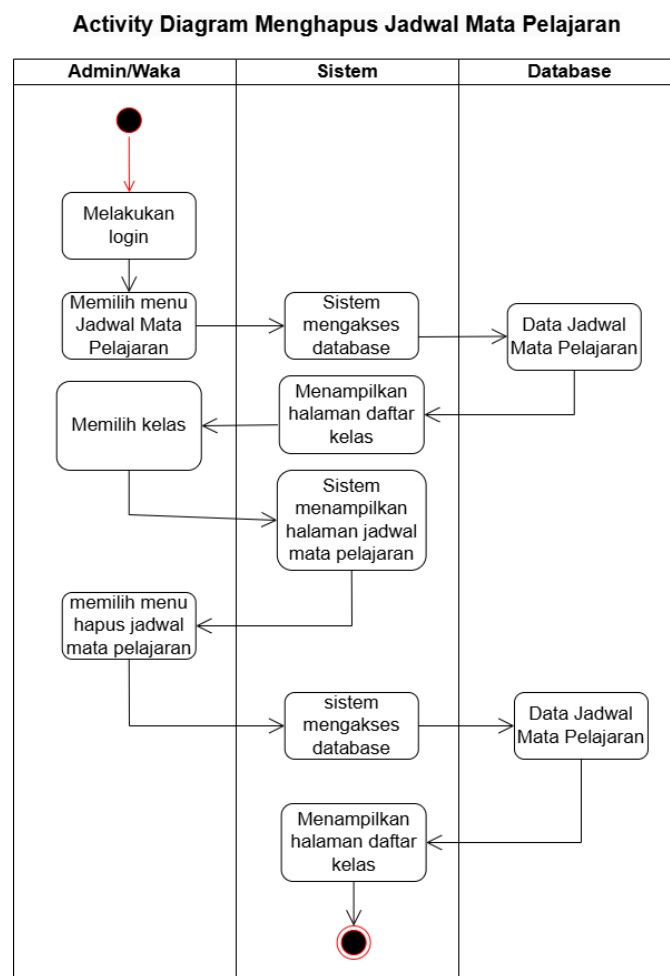


Gambar 25. Activity Diagram Mengubah Jadwal mata Pelajaran

13) Activity Diagram Menghapus Jadwal Mata Pelajaran

Activity Diagram pada Gambar 26 menjelaskan alur proses admin dalam menghapus jadwal mata pelajaran. Proses dimulai ketika admin melakukan Login ke sistem. Setelah berhasil Login, admin memilih menu jadwal mata pelajaran. Selanjutnya, sistem akan mengakses database untuk

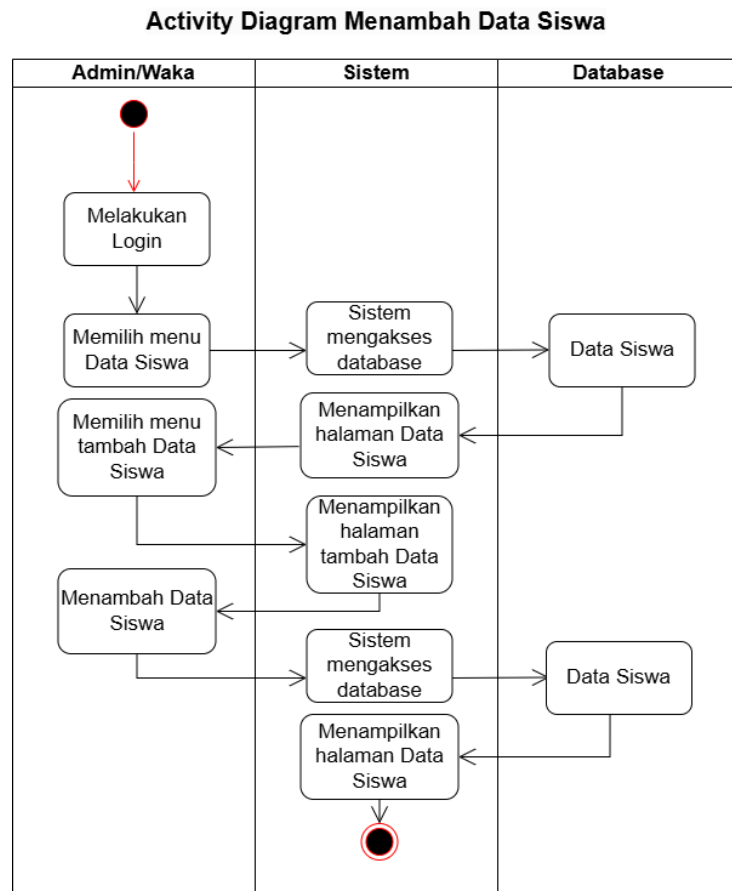
mengambil data jadwal mata pelajaran yang sudah ada dan menampilkan halaman jadwal mata pelajaran. Setelah itu, admin memilih menu hapus jadwal mata pelajaran, yang menyebabkan sistem mengakses database untuk menghapus data jadwal yang dipilih. Setelah data berhasil dihapus, sistem menampilkan halaman jadwal mata pelajaran yang telah diperbarui tanpa data yang telah dihapus. Diagram ini menunjukkan interaksi antara admin, sistem, dan database dalam menghapus jadwal mata pelajaran secara bertahap dan dinamis setelah proses autentikasi dilakukan.



Gambar 26. *Activity Diagram* Menghapus Jadwal Mata Pelajaran

14) *Activity Diagram* Menambah Data Siswa

Activity Diagram pada Gambar 27 menjelaskan alur proses admin dalam menambahkan data siswa. Proses dimulai ketika admin melakukan Login ke sistem. Setelah berhasil Login, admin memilih menu data siswa. Selanjutnya, sistem akan mengakses database untuk mengambil data siswa yang sudah ada dan menampilkan halaman data siswa. Setelah itu, admin memilih menu tambah data siswa, yang menyebabkan sistem menampilkan halaman tambah data siswa. Admin kemudian mengisi informasi siswa baru dan menginputnya ke dalam sistem. Setelah data dimasukkan, sistem akan mengakses database untuk menyimpan data siswa yang baru. Setelah data berhasil disimpan, sistem menampilkan halaman data siswa yang telah diperbarui dengan informasi siswa yang baru ditambahkan. Diagram ini menunjukkan interaksi antara admin, sistem, dan database dalam menambahkan data siswa secara bertahap dan dinamis setelah proses autentikasi dilakukan.

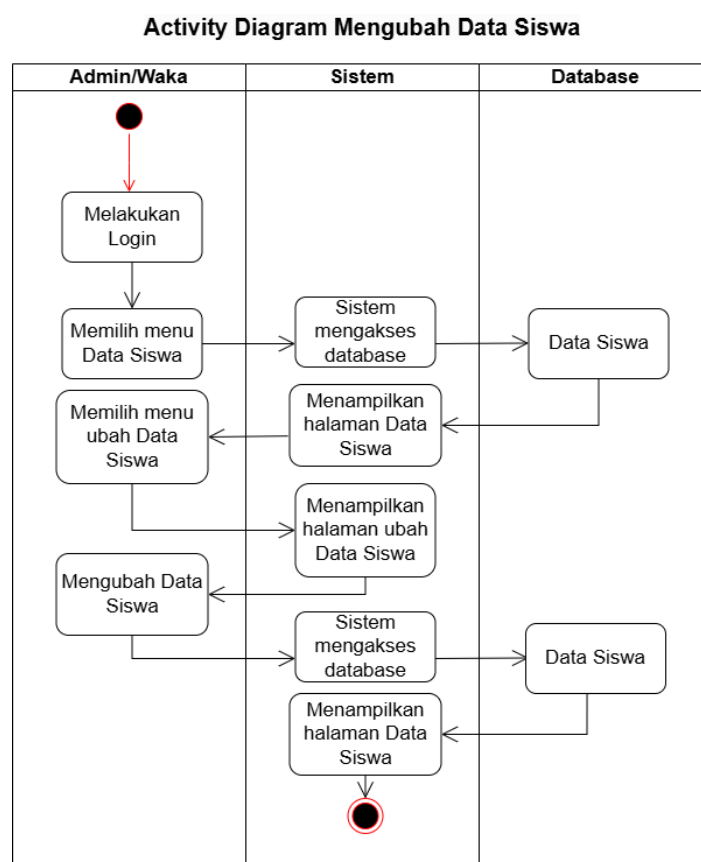


Gambar 27. Activity Diagram Menambah Data Siswa

15) Activity Diagram Mengubah Data Siswa

Activity Diagram pada Gambar 28 menjelaskan alur proses admin dalam mengubah data siswa. Proses dimulai ketika admin melakukan Login ke sistem. Setelah berhasil Login, admin memilih menu data siswa. Selanjutnya, sistem akan mengakses database untuk mengambil data siswa yang sudah ada dan menampilkan halaman data siswa. Setelah itu, admin memilih menu ubah data siswa, yang menyebabkan sistem menampilkan halaman ubah data siswa. Admin kemudian melakukan perubahan pada informasi siswa sesuai kebutuhan dan menginput data yang diperbarui ke dalam sistem. Setelah perubahan dilakukan, sistem akan mengakses database untuk

memperbarui data siswa yang tersimpan. Setelah data berhasil diperbarui, sistem menampilkan halaman data siswa yang telah diperbarui dengan informasi terbaru. Diagram ini menunjukkan interaksi antara admin, sistem, dan database dalam memperbarui data siswa secara bertahap dan dinamis setelah proses autentikasi dilakukan.

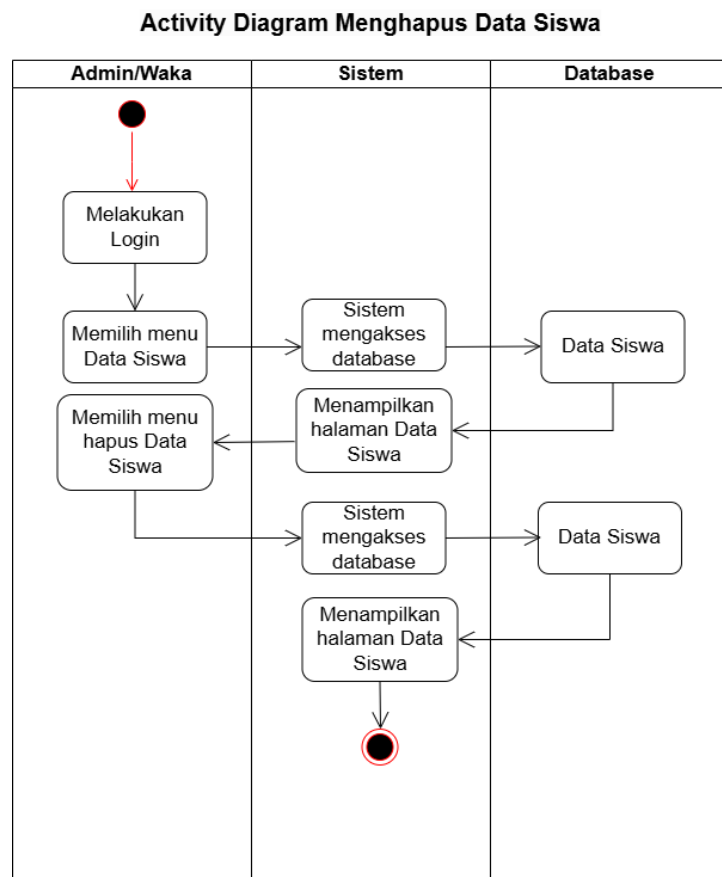


Gambar 28. Activity Diagram Mengubah Data Siswa

16) Activity Diagram Menghapus Data Siswa

Activity Diagram pada Gambar 29 menjelaskan alur proses admin dalam menghapus data siswa. Proses dimulai ketika admin melakukan Login ke sistem. Setelah berhasil Login, admin memilih menu data siswa. Selanjutnya, sistem akan

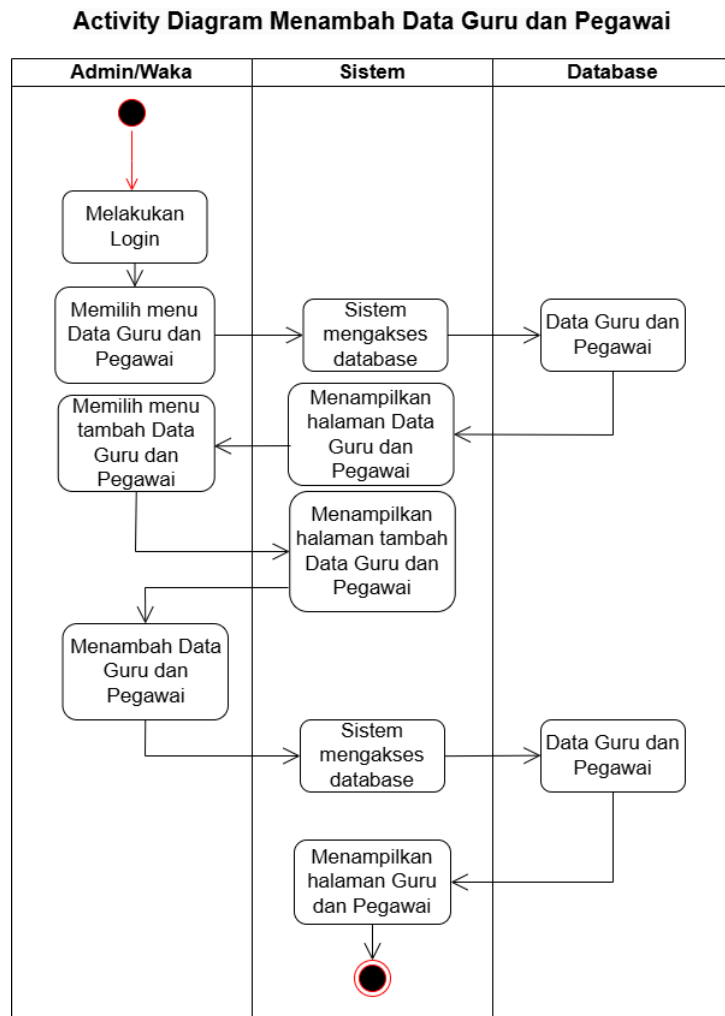
mengakses database untuk mengambil data siswa yang sudah ada dan menampilkan halaman data siswa. Setelah itu, admin memilih menu hapus data siswa, yang menyebabkan sistem mengakses database untuk menghapus data siswa yang dipilih. Setelah data berhasil dihapus, sistem menampilkan halaman data siswa yang telah diperbarui tanpa data yang telah dihapus. Diagram ini menunjukkan interaksi antara admin, sistem, dan database dalam menghapus data siswa secara bertahap dan dinamis setelah proses autentikasi dilakukan.



Gambar 29. Activity Diagram Menghapus Data Siswa

17) Activity Diagram Menambah Data Guru dan Pegawai

Activity Diagram pada Gambar 30 menjelaskan alur proses admin dalam menambahkan data guru dan pegawai. Proses dimulai ketika admin melakukan Login ke sistem, kemudian memilih menu data guru dan pegawai. Sistem akan mengakses database untuk mengambil data yang sudah ada dan menampilkan halaman data guru dan pegawai. Setelah itu, admin memilih menu tambah data guru dan pegawai, sehingga sistem menampilkan halaman tambah data. Admin kemudian mengisi informasi guru atau pegawai baru dan menginputnya ke dalam sistem. Setelah data dimasukkan, sistem akan mengakses database untuk menyimpan data yang baru. Setelah data berhasil disimpan, sistem menampilkan halaman data guru dan pegawai yang telah diperbarui dengan informasi terbaru. Diagram ini menggambarkan interaksi antara admin, sistem, dan database dalam menambahkan data guru dan pegawai secara bertahap dan dinamis setelah proses autentikasi dilakukan.

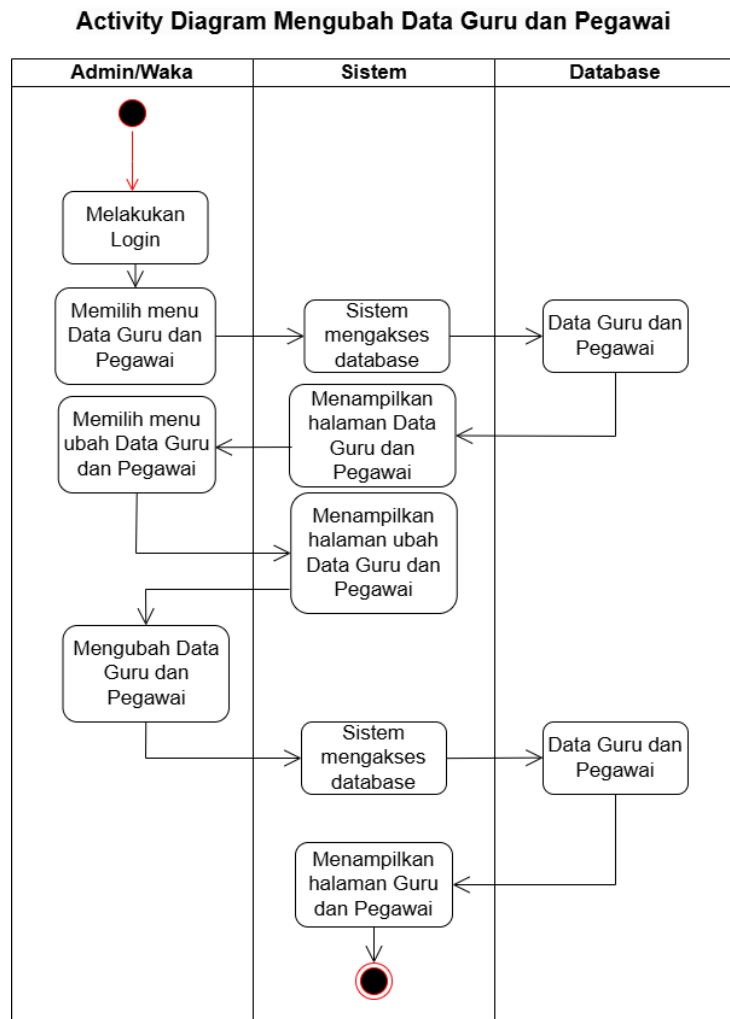


Gambar 30. *Activity Diagram* Menambah Data Guru dan Pegawai

18) *Activity Diagram* Mengubah Data Guru dan Pegawai

Activity Diagram pada Gambar 31 menjelaskan alur proses admin dalam mengubah data guru dan pegawai. Proses diawali dengan admin melakukan Login ke sistem, kemudian memilih menu data guru dan pegawai. Sistem mengakses database untuk mengambil data yang sudah ada dan menampilkan halaman data guru dan pegawai. Setelah itu, admin memilih menu ubah data guru dan pegawai, sehingga sistem menampilkan halaman ubah data. Admin kemudian

melakukan perubahan pada informasi guru atau pegawai sesuai kebutuhan dan menginput data yang diperbarui ke dalam sistem. Setelah perubahan dilakukan, sistem mengakses database untuk memperbarui data yang tersimpan. Setelah data berhasil diperbarui, sistem menampilkan halaman data guru dan pegawai yang telah diperbarui dengan informasi terbaru. Diagram ini menggambarkan interaksi antara admin, sistem, dan database dalam memperbarui data guru dan pegawai secara bertahap dan dinamis setelah proses autentikasi dilakukan.



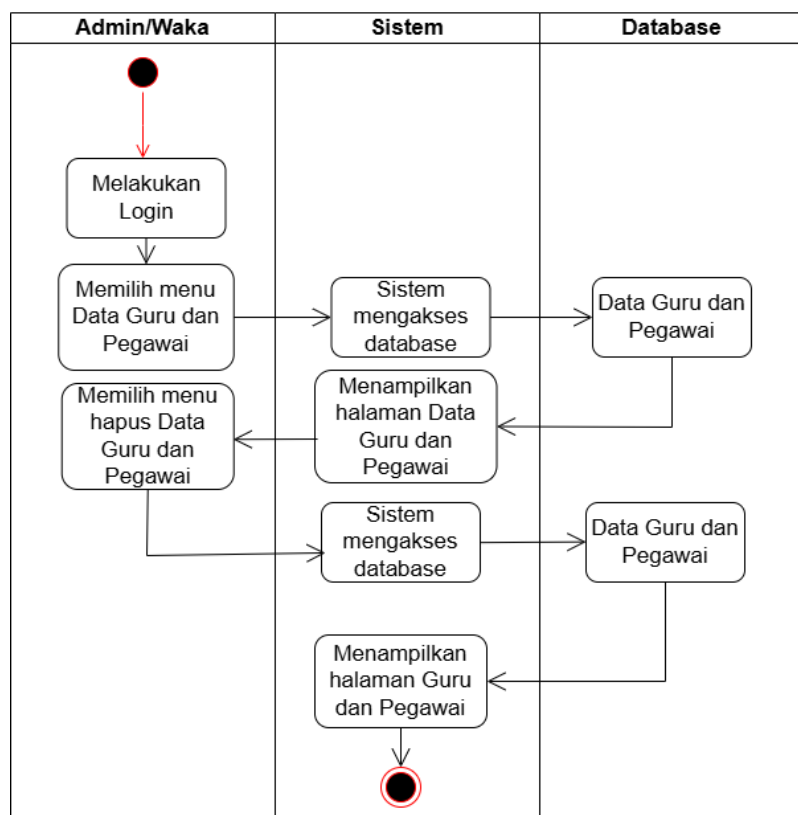
Gambar 31. *Activity Diagram* Mengubah Data Guru dan Pegawai

19) *Activity Diagram* Menghapus Data Guru dan Pegawai

Activity Diagram pada Gambar 32 menjelaskan alur proses admin dalam menghapus data guru dan pegawai. Proses dimulai ketika admin melakukan Login ke sistem, kemudian memilih menu data guru dan pegawai. Sistem mengakses database untuk mengambil data yang sudah ada dan menampilkan halaman data guru dan pegawai. Setelah itu, admin memilih menu hapus data guru dan pegawai, sehingga sistem mengakses database untuk menghapus data yang

dipilih. Setelah data berhasil dihapus, sistem menampilkan halaman data guru dan pegawai yang telah diperbarui tanpa data yang dihapus. Diagram ini menggambarkan interaksi antara admin, sistem, dan database dalam menghapus data guru dan pegawai secara bertahap dan dinamis setelah proses autentikasi dilakukan.

Activity Diagram Menghapus Data Guru dan Pegawai



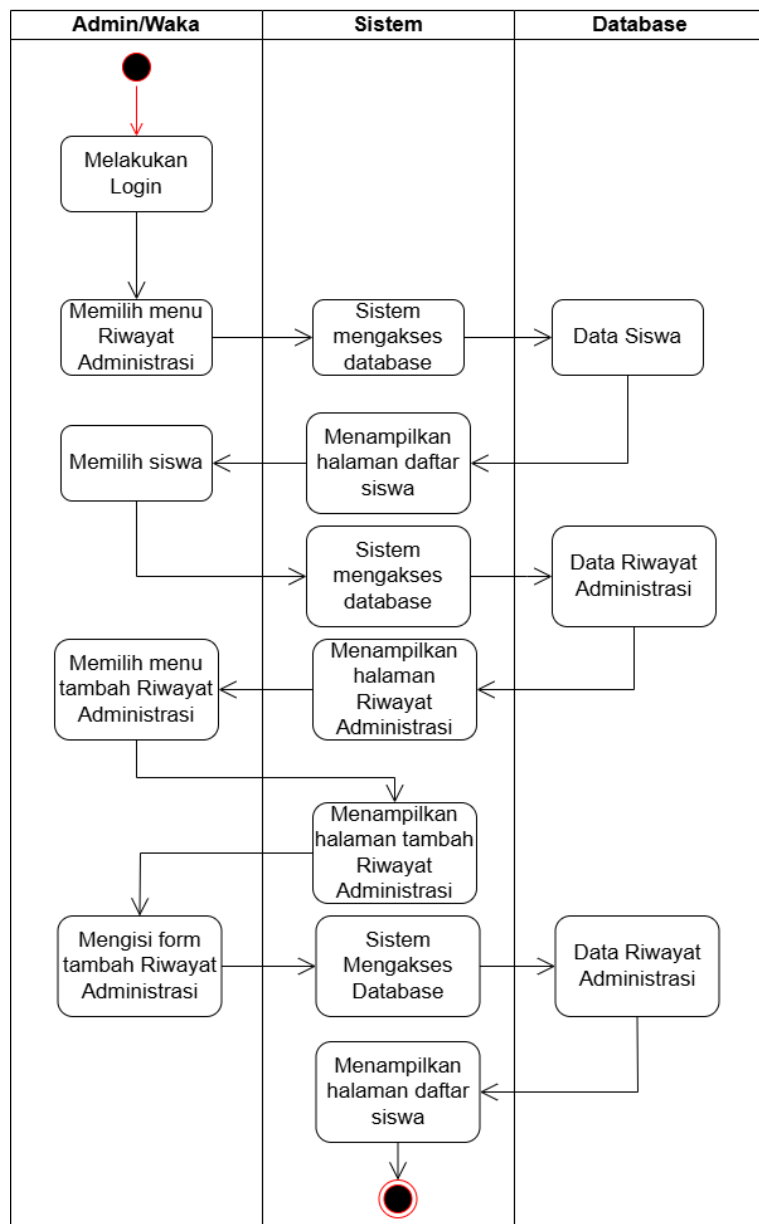
Gambar 32. Activity Diagram Menghapus Data Guru dan Pegawai

20) Activity Diagram Menambah Riwayat Administrasi

Activity Diagram pada Gambar 33 menjelaskan alur proses admin dalam menambahkan riwayat administrasi siswa. Proses dimulai ketika admin melakukan Login ke sistem, kemudian memilih menu riwayat administrasi. Sistem

mengakses database untuk mengambil data siswa yang sudah ada dan menampilkan halaman daftar siswa. Setelah itu, admin memilih salah satu siswa untuk ditambahkan riwayat administrasinya. Sistem kemudian mengakses database untuk mengambil data riwayat administrasi yang sudah ada dan menampilkan halaman riwayat administrasi. Setelah itu, admin memilih menu tambah riwayat administrasi, yang menyebabkan sistem menampilkan halaman tambah riwayat administrasi. Admin mengisi form tambah riwayat administrasi dengan informasi yang diperlukan, lalu sistem mengakses database untuk menyimpan data baru tersebut. Setelah data berhasil disimpan, sistem menampilkan halaman riwayat administrasi yang telah diperbarui. Diagram ini menggambarkan interaksi antara admin, sistem, dan database dalam menambahkan riwayat administrasi siswa secara bertahap dan dinamis setelah proses autentikasi dilakukan.

Activity Diagram Menambah Riwayat Administrasi



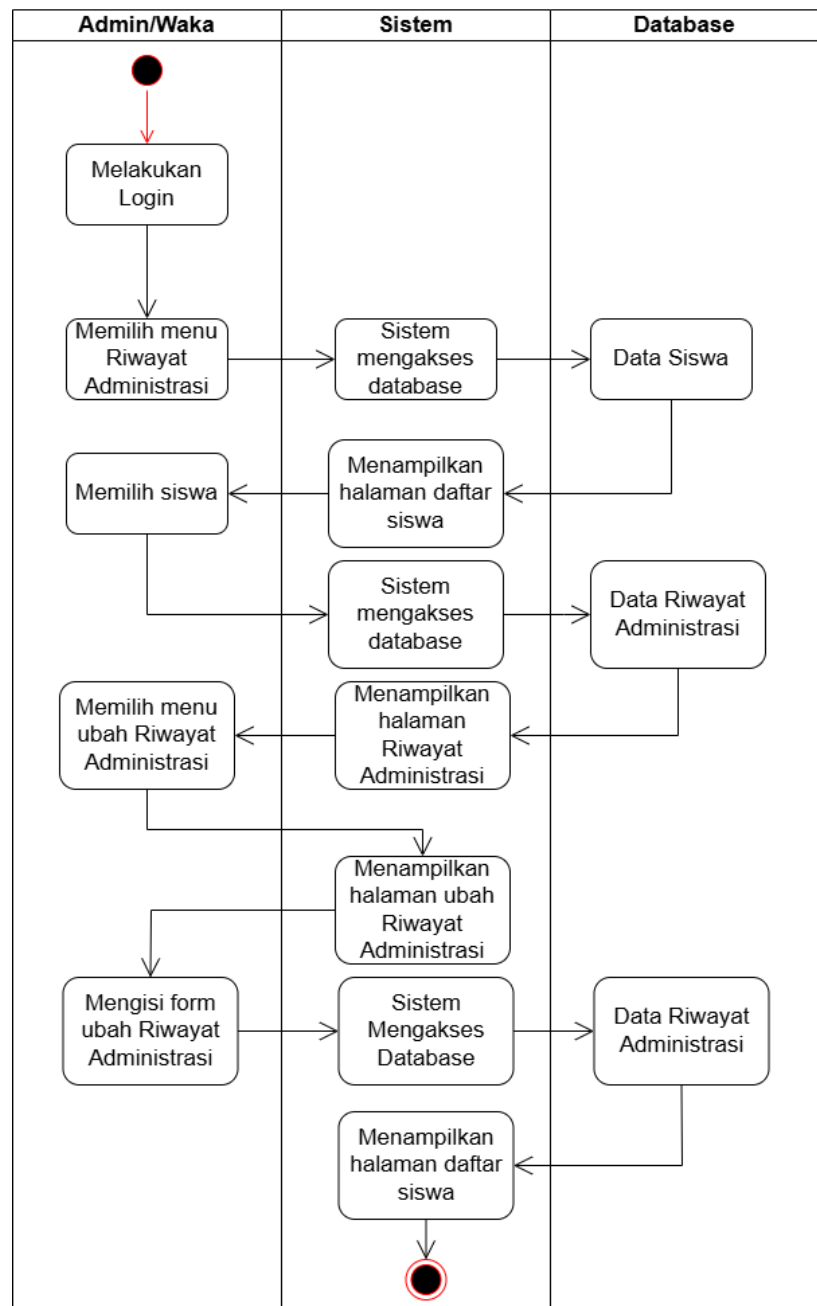
Gambar 33. Activity Diagram Menambah Riwayat Administrasi

21) Activity Diagram Mengubah Riwayat Administrasi

Activity Diagram pada Gambar 34 menjelaskan alur proses admin dalam mengubah riwayat administrasi siswa. Proses dimulai ketika admin melakukan Login ke sistem, kemudian

memilih menu riwayat administrasi. Sistem mengakses database untuk mengambil data siswa yang sudah ada dan menampilkan halaman daftar siswa. Setelah itu, admin memilih salah satu siswa untuk diubah riwayat administrasinya. Sistem kemudian mengakses database untuk mengambil data riwayat administrasi yang sudah ada dan menampilkan halaman riwayat administrasi. Setelah itu, admin memilih menu ubah riwayat administrasi, yang menyebabkan sistem menampilkan halaman ubah riwayat administrasi. Admin kemudian mengisi form ubah riwayat administrasi dengan informasi yang diperbarui, lalu sistem mengakses database untuk menyimpan perubahan tersebut. Setelah data berhasil diperbarui, sistem menampilkan halaman riwayat administrasi yang telah diperbarui dengan informasi terbaru. Diagram ini menggambarkan interaksi antara admin, sistem, dan database dalam mengubah riwayat administrasi siswa secara bertahap dan dinamis setelah proses autentikasi dilakukan.

Activity Diagram Mengubah Riwayat Administrasi

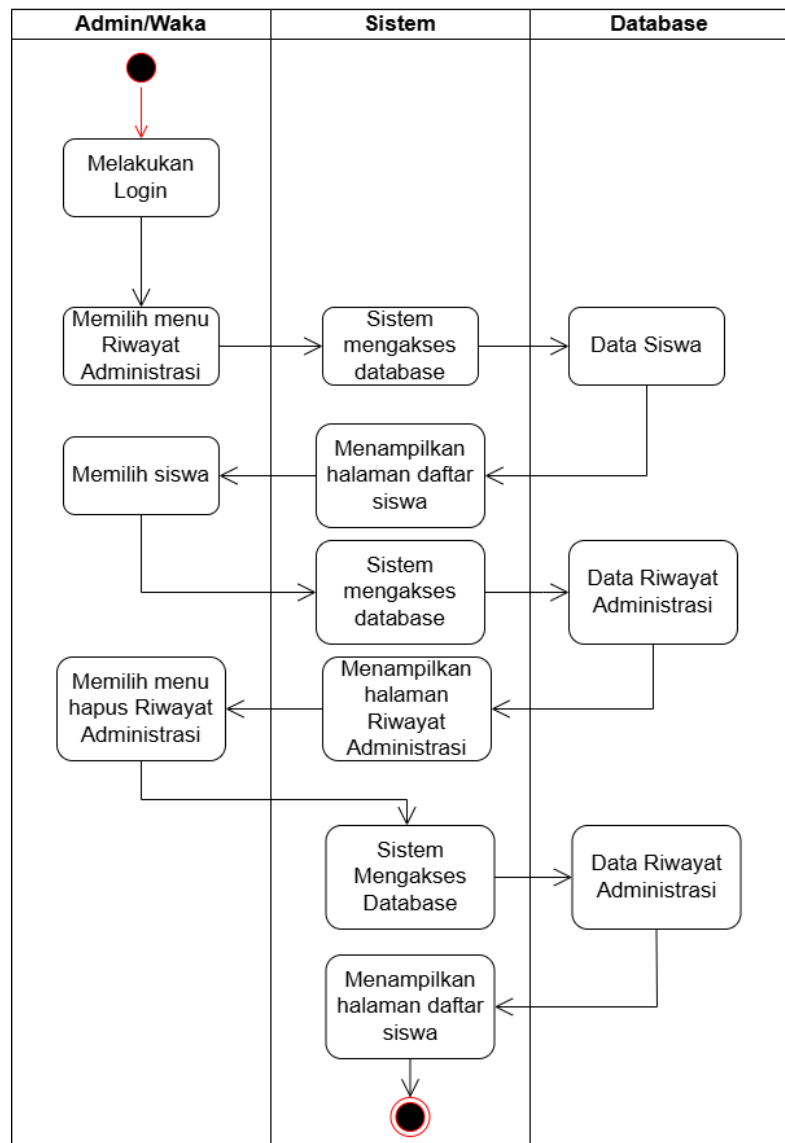


Gambar 34. Activity Diagram Mengubah Riwayat Administrasi

22) Activity Diagram Menghapus Riwayat Administrasi

Activity Diagram pada Gambar 35 menjelaskan alur proses admin dalam menghapus riwayat administrasi siswa. Proses dimulai ketika admin melakukan Login ke sistem, kemudian memilih menu riwayat administrasi. Sistem mengakses database untuk mengambil data siswa yang sudah ada dan menampilkan halaman daftar siswa. Setelah itu, admin memilih salah satu siswa untuk dihapus riwayat administrasinya. Sistem kemudian mengakses database untuk mengambil data riwayat administrasi yang sudah ada dan menampilkan halaman riwayat administrasi. Setelah itu, admin memilih menu hapus riwayat administrasi, sehingga sistem mengakses database untuk menghapus data yang dipilih. Setelah data berhasil dihapus, sistem menampilkan halaman riwayat administrasi yang telah diperbarui tanpa data yang dihapus. Diagram ini menggambarkan interaksi antara admin, sistem, dan database dalam menghapus riwayat administrasi siswa secara bertahap dan dinamis setelah proses autentikasi dilakukan.

Activity Diagram Menghapus Riwayat Administrasi

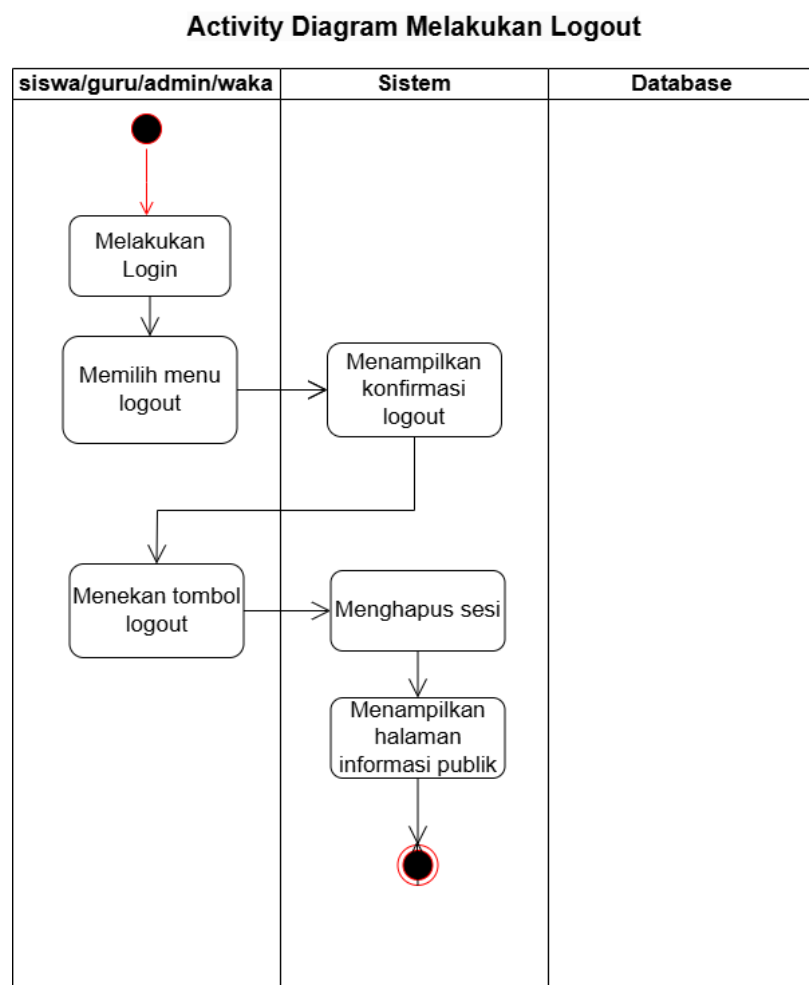


Gambar 35. Activity Diagram Menghapus Riwayat Administrasi

23) Activity Diagram Logout

Activity Diagram pada Gambar 36 menjelaskan alur proses Logout bagi pengguna yang terdiri dari siswa, guru, atau admin. Proses dimulai ketika pengguna memilih opsi Logout pada sistem. Setelah itu, sistem akan memproses permintaan

Logout dengan mengakhiri sesi aktif pengguna dan membersihkan data sesi dari memori. Setelah sesi berhasil diakhiri, sistem akan mengarahkan pengguna kembali ke halaman informasi publik. Diagram ini menggambarkan interaksi antara pengguna dan sistem dalam memastikan proses keluar dari sistem dilakukan dengan aman dan tepat.



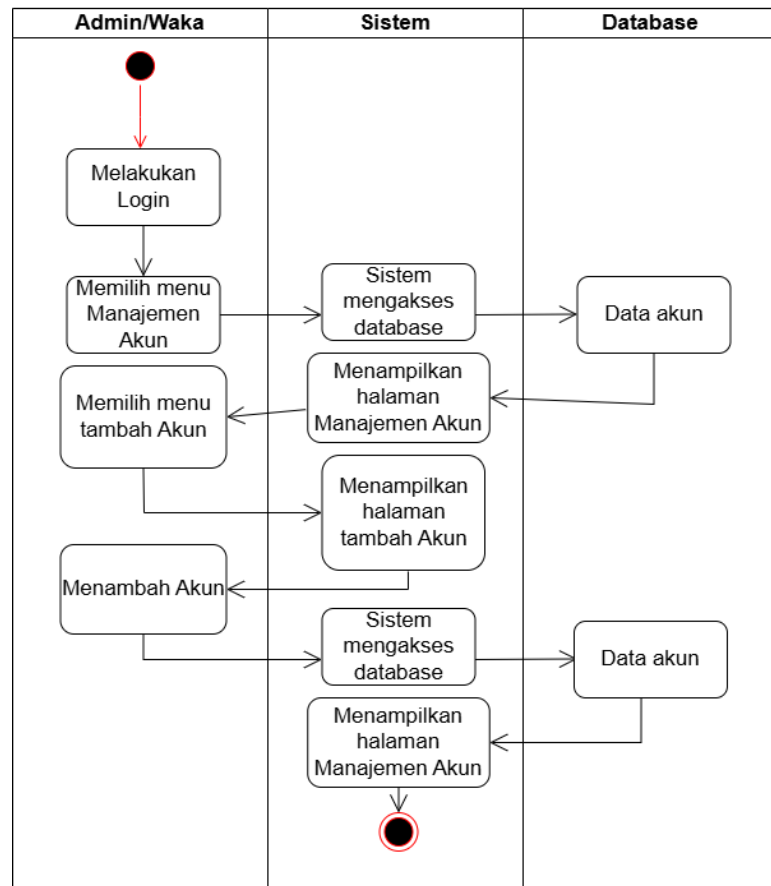
Gambar 36. *Activity Diagram Melakukan Logout*

24) *Activity Diagram Menambah Akun*

Activity Diagram pada Gambar 37 menjelaskan alur proses admin dalam menambahkan akun pengguna. Proses dimulai

ketika admin melakukan *Login* ke dalam sistem. Setelah berhasil *Login*, admin memilih menu manajemen akun. Selanjutnya, sistem akan mengakses database untuk mengambil data akun yang sudah ada dan menampilkan halaman daftar akun. Setelah itu, admin memilih menu tambah akun, yang menyebabkan sistem menampilkan halaman tambah akun baru. Admin kemudian mengisi informasi akun pengguna yang akan ditambahkan dan menginputnya ke dalam sistem. Setelah data akun dimasukkan, sistem akan mengakses database untuk menyimpan data akun baru tersebut. Setelah data berhasil disimpan, sistem menampilkan halaman daftar akun yang telah diperbarui dengan informasi akun yang baru ditambahkan. Diagram ini menunjukkan interaksi antara admin, sistem, dan database dalam proses penambahan akun pengguna secara bertahap setelah proses autentikasi dilakukan.

Activity Diagram Menambah Akun



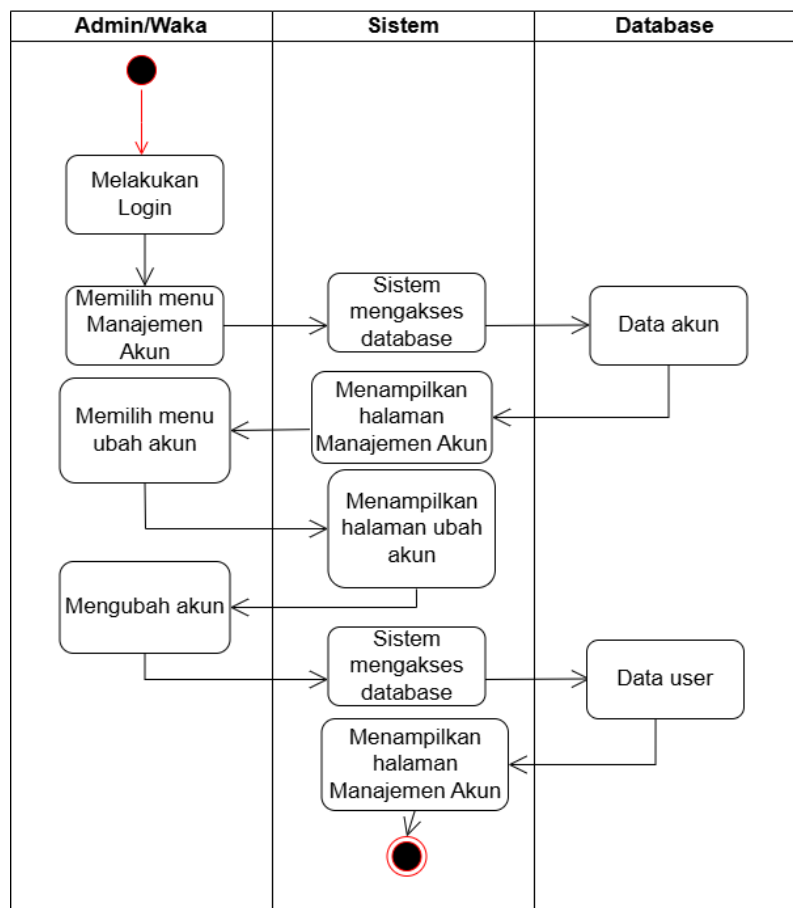
Gambar 37. Activity Diagram Menambah Akun

25) Activity Diagram Mengubah Akun

Activity Diagram pada Gambar 38 menjelaskan alur proses admin dalam mengubah akun pengguna. Proses dimulai ketika admin melakukan Login ke sistem. Setelah berhasil Login, admin memilih menu manajemen akun. Selanjutnya, sistem akan mengakses database untuk mengambil data akun yang sudah ada dan menampilkan halaman daftar akun. Setelah itu, admin memilih menu ubah akun, yang menyebabkan sistem menampilkan halaman ubah akun. Admin kemudian melakukan perubahan pada informasi akun sesuai kebutuhan dan menginput data yang telah diperbarui ke dalam sistem.

Setelah perubahan dilakukan, sistem akan mengakses database untuk memperbarui data akun yang tersimpan. Setelah data berhasil diperbarui, sistem menampilkan halaman daftar akun yang telah diperbarui dengan informasi terbaru. Diagram ini menunjukkan interaksi antara admin, sistem, dan database dalam memperbarui data akun pengguna secara bertahap setelah proses autentikasi dilakukan.

Activity Diagram Mengubah Akun



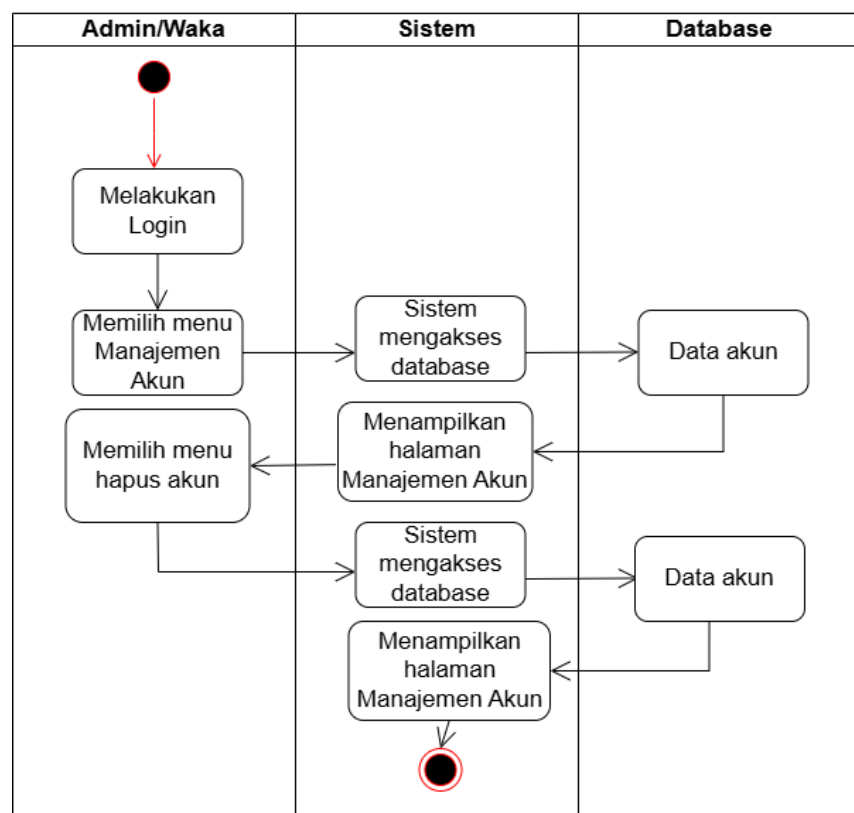
Gambar 38. Activity Diagram Mengubah Akun

26) Activity Diagram Menghapus Akun

Activity Diagram pada Gambar 39 menjelaskan alur proses admin dalam menghapus akun pengguna. Proses dimulai

ketika admin melakukan Login ke sistem. Setelah berhasil Login, admin memilih menu manajemen akun. Selanjutnya, sistem akan mengakses database untuk mengambil data akun yang sudah ada dan menampilkan halaman daftar akun. Setelah itu, admin memilih menu hapus akun, yang menyebabkan sistem mengakses database untuk menghapus data akun yang dipilih. Setelah data akun berhasil dihapus, sistem menampilkan halaman daftar akun yang telah diperbarui tanpa data akun yang telah dihapus. Diagram ini menunjukkan interaksi antara admin, sistem, dan database dalam menghapus akun pengguna secara bertahap dan dinamis setelah proses autentikasi dilakukan.

Activity Diagram Menghapus Akun

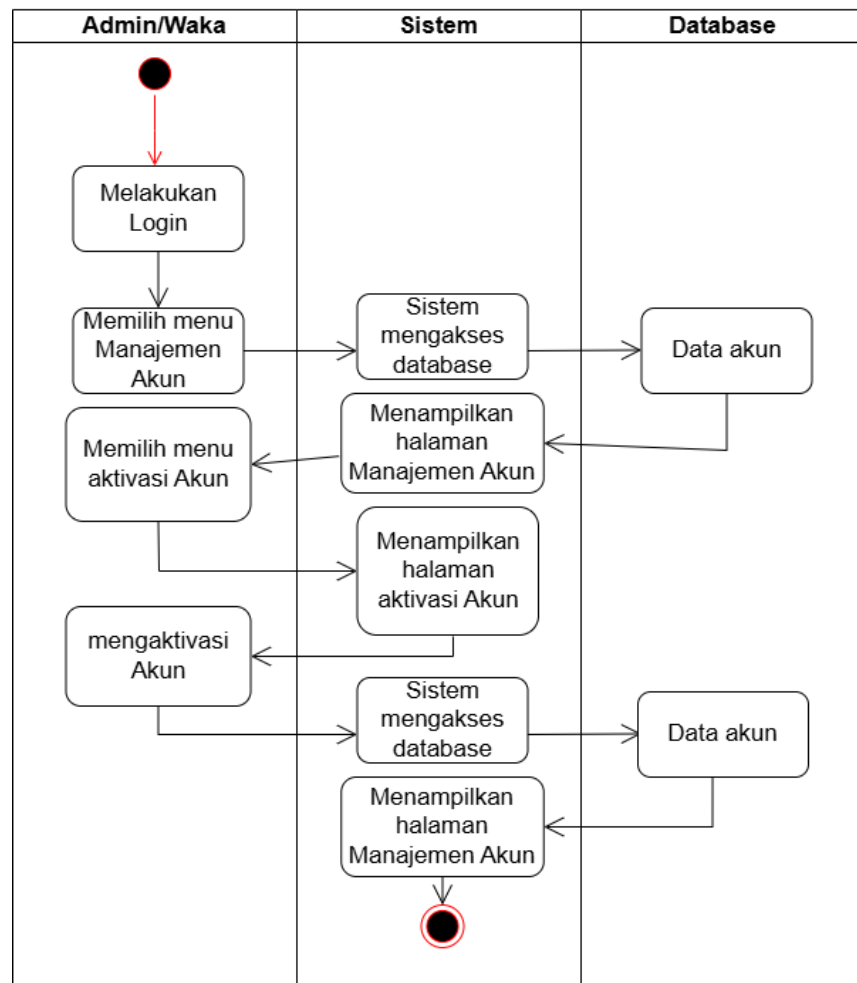


Gambar 39. Activity Diagram Menghapus Akun

27) *Activity Diagram* Aktivasi Akun

Activity Diagram pada Gambar 40 menjelaskan alur proses admin dalam mengaktivasi akun pengguna. Proses dimulai ketika admin melakukan Login ke sistem. Setelah berhasil Login, admin memilih menu manajemen akun. Selanjutnya, sistem akan mengakses database untuk mengambil data akun yang belum aktif dan menampilkan halaman daftar akun. Setelah itu, admin memilih akun yang akan diaktivasi, lalu sistem menampilkan halaman konfirmasi aktivasi. Setelah admin mengonfirmasi, sistem akan memperbarui status akun di database menjadi aktif. Setelah akun berhasil diaktivasi, sistem menampilkan halaman daftar akun yang telah diperbarui dengan status akun yang telah aktif. Diagram ini menunjukkan interaksi antara admin, sistem, dan database dalam proses aktivasi akun pengguna secara bertahap setelah proses autentikasi dilakukan.

Activity Diagram Aktivasi Akun

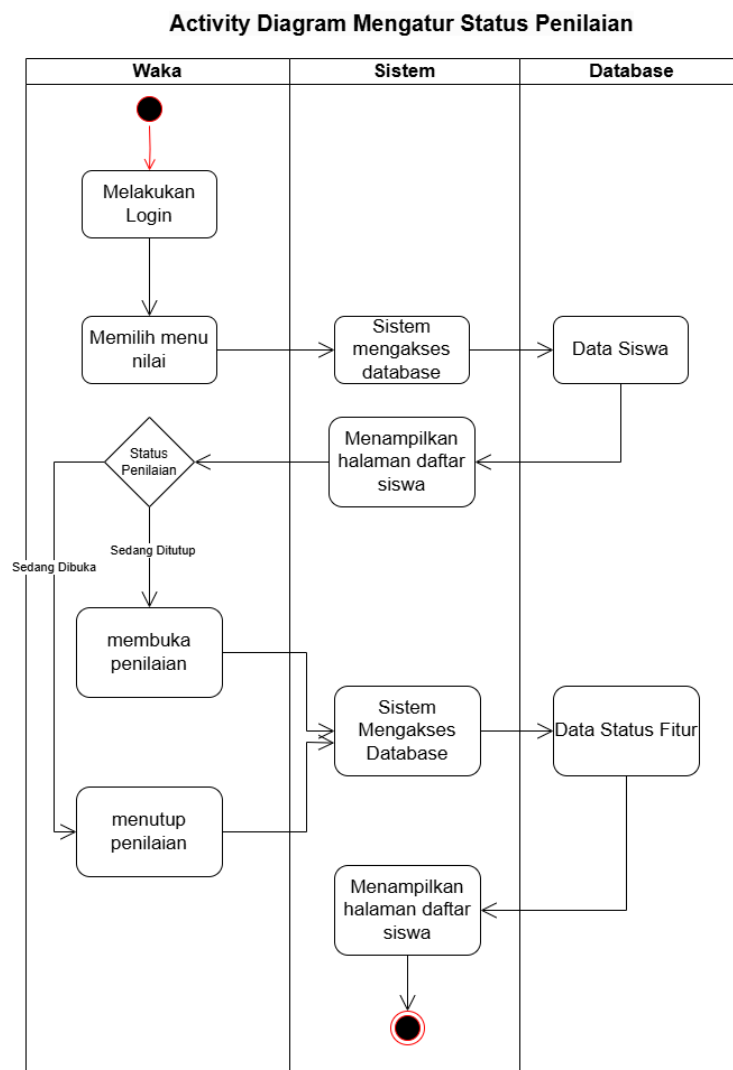


Gambar 40. Activity Diagram Aktivasi Akun

28) Activity Diagram Mengatur Status Penilaian

Activity Diagram pada Gambar 41 menjelaskan alur proses Waka dalam mengatur status penilaian. Proses dimulai ketika Waka melakukan Login ke sistem. Setelah berhasil Login, Waka memilih menu nilai untuk mengakses pengaturan status penilaian. Selanjutnya, sistem akan mengakses database untuk memeriksa status penilaian saat ini dan menampilkan halaman daftar siswa. Jika penilaian sedang ditutup, Waka dapat memilih untuk membuka penilaian, yang akan memodifikasi

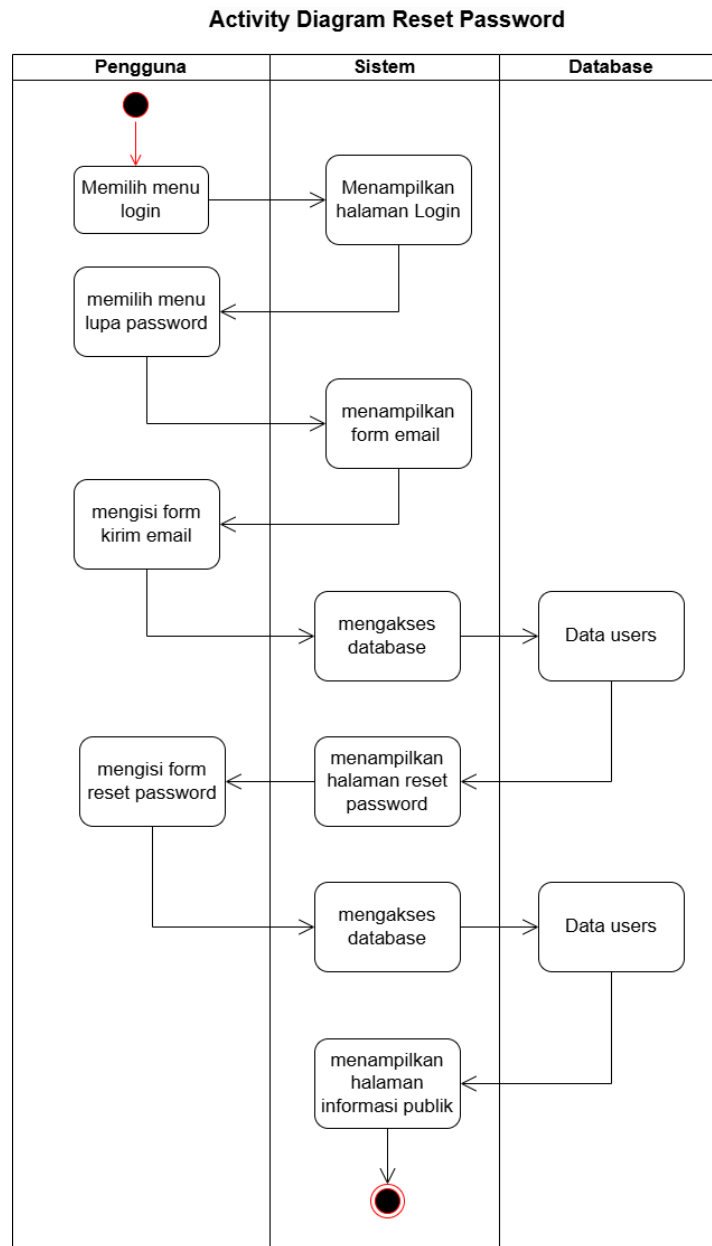
status penilaian menjadi terbuka dalam database, dan sistem akan menampilkan halaman daftar siswa yang telah diperbarui. Sebaliknya, jika penilaian sedang dibuka, Waka dapat memilih untuk menutup penilaian, yang akan mengubah status penilaian di database menjadi ditutup. Setelah itu, sistem akan kembali menampilkan halaman daftar siswa yang terbaru. Diagram ini menunjukkan interaksi antara Waka, sistem, dan database dalam proses pengaturan status penilaian secara bertahap setelah proses login dilakukan.



Gambar 41. Activity Diagram Mengatur Status Penilaian

29) *Activity Diagram Reset Password*

Activity Diagram pada Gambar 42 menjelaskan alur proses reset password oleh pengguna. Proses dimulai ketika Pengguna memilih menu login pada halaman utama. Setelah itu, pengguna memilih opsi lupa password untuk memulai proses pemulihan akses. Sistem kemudian menampilkan form untuk mengirim email, di mana pengguna harus mengisi form tersebut dengan alamat email yang terdaftar. Setelah itu, sistem mengakses database untuk memverifikasi data pengguna, kemudian menampilkan form reset password kepada pengguna. Pengguna kemudian mengisi form untuk mengatur ulang password mereka. Setelah pengisian selesai, sistem kembali mengakses database untuk memperbarui password pengguna, dan akhirnya menampilkan halaman informasi publik yang menandakan bahwa proses reset password telah berhasil. Diagram ini menunjukkan interaksi antara Pengguna, sistem, dan database dalam proses reset password secara bertahap setelah pengguna memilih opsi untuk mengatur ulang password.



Gambar 42. Activity Diagram Reset Password

E. Use Case Description

Deskripsi setiap *Use Case* diagram adalah sebagai berikut:

Tabel 5. *Use Case Description* untuk Melakukan Login

Use Case Name: Melakukan Login	ID: UC-01
Actor: Siswa, Guru, Admin, Waka	
Preconditions: Siswa, Guru, dan Admin tidak sudah Login kedalam sistem	

Normal Course: 1. Siswa, Guru, dan Admin Mengakses halaman <i>Login</i> 2. Sistem menampilkan halaman <i>Login</i> 3. Siswa, Guru, dan Admin mengisi form <i>Login</i> 4. Sistem meminta data user kepada <i>database</i> 5. <i>Database</i> memberikan hasil pencarian data user kepada sistem 6. Sistem memvalidasi apakah data benar atau tidak 7. jika benar, sistem menampilkan halaman beranda
Postconditions: Siswa, Guru, dan Admin berhasil melakukan <i>Login</i> dan dapat menggunakan fitur fitur yang ada di dalam sistem

Tabel 6. *Use Case Description* untuk Melihat Data Siswa

Use Case Name: Melihat Data Siswa	ID: UC-02
Actor: Siswa, Guru, Admin, Waka	
Preconditions: Siswa, Guru, dan Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Siswa, Guru, dan Admin melakukan <i>Login</i> 2. Siswa, Guru, dan Admin memilih menu data siswa 3. Sistem meminta data siswa kepada <i>database</i> 4. <i>Database</i> memberikan data siswa kepada sistem 5. Sistem menampilkan halaman data siswa	
Postconditions: Siswa, Guru, dan Admin berhasil membuka halaman data siswa	

Tabel 7. *Use Case Description* untuk Melihat Data Guru dan Pegawai

Use Case Name: Melihat Data Guru dan Pegawai	ID: UC-03
Actor: Siswa, Guru, Admin, Waka	
Preconditions: Siswa, Guru, dan Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Siswa, Guru, dan Admin melakukan <i>Login</i> 2. Siswa, Guru, dan Admin memilih menu data guru dan pegawai 3. Sistem meminta data guru dan pegawai kepada <i>database</i> 4. <i>Database</i> memberikan data guru dan pegawai kepada sistem 5. Sistem menampilkan halaman data guru dan pegawai	
Postconditions: Siswa, Guru, dan Admin berhasil membuka halaman data guru dan pegawai	

Tabel 8. *Use Case Description* untuk Melihat Riwayat Administrasi

Use Case Name: Melihat Riwayat Administrasi	ID: UC-04
Actor: Siswa, Guru, Admin, Waka	
Preconditions: Siswa, Guru, dan Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Siswa, Guru, dan Admin melakukan <i>Login</i> 2. Siswa, Guru, dan Admin memilih menu riwayat administrasi 3. Sistem meminta data siswa kepada <i>database</i> 4. <i>Database</i> memberikan data siswa kepada sistem 5. Sistem menampilkan halaman daftar siswa 6. Siswa, Guru, dan Admin memilih siswa 7. Sistem meminta data riwayat administrasi kepada <i>database</i> 8. <i>Database</i> memberikan data riwayat administrasi kepada sistem 9. Sistem menampilkan halaman riwayat administrasi	
Postconditions: Siswa, Guru, dan Admin berhasil membuka halaman riwayat administrasi	

Tabel 9. *Use Case Description* untuk Melihat Jadwal Mata Pelajaran

Use Case Name: Melihat Jadwal Mata Pelajaran	ID: UC-05
Actor: Siswa, Guru, Admin, Waka	
Preconditions: Siswa, Guru, dan Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Siswa, Guru, dan Admin melakukan <i>Login</i> 2. Siswa, Guru, dan Admin memilih menu jadwal mata pelajaran 3. Sistem meminta data jadwal mata pelajaran kepada <i>database</i> 4. <i>Database</i> memberikan data jadwal mata pelajaran kepada sistem 5. Sistem menampilkan halaman jadwal mata pelajaran	
Postconditions: Siswa, Guru, dan Admin berhasil membuka halaman jadwal mata pelajaran	

Tabel 10. *Use Case Description* untuk Melihat Nilai

Use Case Name: Melihat Nilai	ID: UC-06
Actor: Siswa, Guru, Admin, Waka	
Preconditions: Siswa, Guru, dan Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Siswa, Guru, dan Admin melakukan <i>Login</i> 2. Siswa, Guru, dan Admin memilih menu nilai 3. Sistem meminta data siswa kepada <i>database</i> 4. <i>Database</i> memberikan data siswa kepada sistem 5. Sistem menampilkan halaman daftar siswa	

6. Siswa, Guru, dan Admin memilih siswa 7. Sistem meminta data nilai kepada <i>database</i> 8. <i>Database</i> memberikan data nilai kepada sistem 9. Sistem menampilkan halaman nilai
Postconditions: Siswa, Guru, dan Admin berhasil membuka halaman nilai

Tabel 11. *Use Case Description* untuk Melihat Kemajuan Belajar

Use Case Name: Melihat Kemajuan Belajar	ID: UC-07
Actor: Siswa, Guru, Admin, Waka	
Preconditions: Siswa, Guru, dan Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Siswa, Guru, dan Admin melakukan <i>Login</i> 2. Siswa, Guru, dan Admin memilih menu kemajuan belajar 3. Sistem meminta data siswa kepada <i>database</i> 4. <i>Database</i> memberikan data siswa kepada sistem 5. Sistem menampilkan halaman daftar siswa 6. Siswa, Guru, dan Admin memilih siswa 7. Sistem meminta data kemajuan belajar kepada <i>database</i> 8. <i>Database</i> memberikan data kemajuan belajar kepada sistem 9. Sistem menampilkan halaman kemajuan belajar	
Postconditions: Siswa, Guru, dan Admin berhasil membuka halaman kemajuan belajar	

Tabel 12. *Use Case Description* untuk Menambah Nilai

Use Case Name: Menambah Nilai	ID: UC-08
Actor: Guru, Admin, Waka	
Preconditions: Guru, dan Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Guru, dan Admin melakukan <i>Login</i> 2. Guru, dan Admin memilih menu nilai 3. Sistem meminta data siswa kepada <i>database</i> 4. <i>Database</i> memberikan data siswa kepada sistem 5. Sistem menampilkan halaman daftar siswa 6. Guru, dan Admin memilih siswa 7. Sistem meminta data nilai kepada <i>database</i> 8. <i>Database</i> memberikan data nilai kepada sistem 9. Sistem menampilkan halaman nilai 10. Guru, dan Admin memilih menu tambah nilai 11. Sistem menampilkan halaman tambah nilai 12. Guru, dan Admin mengisi form tambah nilai	

13. Sistem mengirim data nilai ke <i>database</i>
14. <i>Database</i> menyimpan data nilai
15. Sistem menampilkan halaman nilai
Postconditions: Guru, dan Admin berhasil menambahkan nilai siswa

Tabel 13. *Use Case Description* untuk Mengubah Nilai

Use Case Name: Mengubah Nilai	ID: UC-09
Actor: Guru, Admin, Waka	
Preconditions: Guru, dan Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Guru, dan Admin melakukan <i>Login</i> 2. Guru, dan Admin memilih menu nilai 3. Sistem meminta data siswa kepada <i>database</i> 4. <i>Database</i> memberikan data siswa kepada sistem 5. Sistem menampilkan halaman daftar siswa 6. Guru, dan Admin memilih siswa 7. Sistem meminta data nilai kepada <i>database</i> 8. <i>Database</i> memberikan data nilai kepada sistem 9. Sistem menampilkan halaman nilai 10. Guru, dan Admin memilih menu ubah nilai 11. Sistem menampilkan halaman ubah nilai 12. Guru, dan Admin mengisi form ubah nilai 13. Sistem mengirim data nilai ke <i>database</i> 14. <i>Database</i> menyimpan data nilai 15. Sistem menampilkan halaman nilai	
Postconditions: Guru, dan Admin berhasil mengubah nilai siswa	

Tabel 14. *Use Case Description* untuk Menghapus Nilai

Use Case Name: Menghapus Nilai	ID: UC-10
Actor: Guru, Admin, Waka	
Preconditions: Guru, dan Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Guru, dan Admin melakukan <i>Login</i> 2. Guru, dan Admin memilih menu nilai 3. Sistem meminta data siswa kepada <i>database</i> 4. <i>Database</i> memberikan data siswa kepada sistem 5. Sistem menampilkan halaman daftar siswa 6. Guru, dan Admin memilih siswa 7. Sistem meminta data nilai kepada <i>database</i> 8. <i>Database</i> memberikan data nilai kepada sistem	

9. Sistem menampilkan halaman nilai 10. Guru, dan Admin memilih menu hapus nilai 11. Sistem mengirim perintah hapus nilai ke <i>database</i> 12. <i>Database</i> menghapus data nilai 13. Sistem menampilkan halaman nilai
Postconditions: Guru, dan Admin berhasil menghapus nilai siswa

Tabel 15. *Use Case Description* untuk Menambah Jadwal Mata Pelajaran

Use Case Name: Menambah Jadwal Mata Pelajaran	ID: UC-11
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu jadwal mata pelajaran 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data jadwal mata pelajaran kepada sistem 5. Sistem menampilkan halaman jadwal mata pelajaran 6. Admin memilih menu tambah jadwal mata pelajaran 7. Sistem menampilkan halaman tambah jadwal mata pelajaran 8. Admin menambah jadwal mata pelajaran 9. Sistem mengirim data jadwal mata pelajaran kepada <i>database</i> 10. <i>Database</i> menyimpan data jadwal mata pelajaran 11. Sistem menampilkan halaman jadwal mata pelajaran	
Postconditions: Admin berhasil menambahkan jadwal mata pelajaran	

Tabel 16. *Use Case Description* untuk Mengubah Jadwal Mata Pelajaran

Use Case Name: Mengubah Jadwal Mata Pelajaran	ID: UC-12
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu jadwal mata pelajaran 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data jadwal mata pelajaran kepada sistem 5. Sistem menampilkan halaman jadwal mata pelajaran 6. Admin memilih menu ubah jadwal mata pelajaran 7. Sistem menampilkan halaman ubah jadwal mata pelajaran 8. Admin mengubah jadwal mata pelajaran 9. Sistem mengirim data jadwal mata pelajaran kepada <i>database</i>	

10. <i>Database</i> menyimpan data jadwal mata pelajaran
11. Sistem menampilkan halaman jadwal mata pelajaran
Postconditions: Admin berhasil mengubah jadwal mata pelajaran

Tabel 17. *Use Case Description* untuk Menghapus
Jadwal Mata Pelajaran

Use Case Name: Menghapus Jadwal Mata Pelajaran	ID: UC-13
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu jadwal mata pelajaran 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data jadwal mata pelajaran kepada sistem 5. Sistem menampilkan halaman jadwal mata pelajaran 6. Admin memilih menu hapus jadwal mata pelajaran 7. Sistem mengirim perintah hapus data kepada <i>database</i> 8. <i>Database</i> menghapus data jadwal mata pelajaran 9. Sistem menampilkan halaman jadwal mata pelajaran	
Postconditions: Admin berhasil menghapus jadwal mata pelajaran	

Tabel 18. *Use Case Description* untuk Menambah
Data Siswa

Use Case Name: Menambah Data Siswa	ID: UC-14
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu data siswa 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data siswa kepada sistem 5. Sistem menampilkan halaman data siswa 6. Admin memilih menu tambah data siswa 7. Sistem menampilkan halaman tambah data siswa 8. Admin menambah data siswa 9. Sistem mengirim data siswa kepada <i>database</i> 10. <i>Database</i> menyimpan data siswa 11. Sistem menampilkan halaman data siswa	
Postconditions: Admin berhasil menambah data siswa	

Tabel 19. *Use Case Description* untuk Mengubah Data Siswa

Use Case Name: Mengubah Data Siswa	ID: UC-15
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu data siswa 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data siswa kepada sistem 5. Sistem menampilkan halaman data siswa 6. Admin memilih menu ubah data siswa 7. Sistem menampilkan halaman ubah data siswa 8. Admin mengubah data siswa 9. Sistem mengirim data siswa kepada <i>database</i> 10. <i>Database</i> menyimpan data siswa 11. Sistem menampilkan halaman data siswa	
Postconditions: Admin berhasil mengubah data siswa	

Tabel 20. *Use Case Description* untuk Menghapus Data Siswa

Use Case Name: Menghapus Data Siswa	ID: UC-16
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu data siswa 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data siswa kepada sistem 5. Sistem menampilkan halaman data siswa 6. Admin memilih menu hapus data siswa 7. Sistem mengirim perintah hapus data kepada <i>database</i> 8. <i>Database</i> menghapus data siswa 9. Sistem menampilkan halaman data siswa	
Postconditions: Admin berhasil menghapus data siswa	

Tabel 21. *Use Case Description* untuk Menambah Data Guru dan Pegawai

Use Case Name: Menambah Data Guru dan Pegawai	ID: UC-17
Actor: Admin, Waka	

Preconditions: Admin harus sudah <i>Login</i> kedalam sistem
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu data guru dan pegawai 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data guru dan pegawai kepada sistem 5. Sistem menampilkan halaman data guru dan pegawai 6. Admin memilih menu tambah data guru dan pegawai 7. Sistem menampilkan halaman tambah data guru dan pegawai 8. Admin menambah data guru dan pegawai 9. Sistem mengirim data guru dan pegawai kepada <i>database</i> 10. <i>Database</i> menyimpan data guru dan pegawai 11. Sistem menampilkan halaman data guru dan pegawai
Postconditions: Admin berhasil menambah data guru dan pegawai

Tabel 22. *Use Case Description* untuk Mengubah Data Guru dan Pegawai

Use Case Name: Mengubah Data Guru dan Pegawai	ID: UC-18
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu data guru dan pegawai 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data guru dan pegawai kepada sistem 5. Sistem menampilkan halaman data guru dan pegawai 6. Admin memilih menu ubah data guru dan pegawai 7. Sistem menampilkan halaman ubah data guru dan pegawai 8. Admin mengubah data guru dan pegawai 9. Sistem mengirim data guru dan pegawai kepada <i>database</i> 10. <i>Database</i> menyimpan data guru dan pegawai 11. Sistem menampilkan halaman data guru dan pegawai	
Postconditions: Admin berhasil mengubah data guru dan pegawai	

Tabel 23. *Use Case Description* untuk Menghapus Data Guru dan Pegawai

Use Case Name: Menghapus Data Guru dan Pegawai	ID: UC-19
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i>	

2. Admin memilih menu data guru dan pegawai 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data guru dan pegawai kepada sistem 5. Sistem menampilkan halaman data guru dan pegawai 6. Admin memilih menu hapus data guru dan pegawai 7. Sistem mengirim perintah hapus data kepada <i>database</i> 8. <i>Database</i> menghapus data guru dan pegawai 9. Sistem menampilkan halaman data guru dan pegawai
Postconditions: Admin berhasil menghapus data guru dan pegawai

Tabel 24. *Use Case Description* untuk Menambah Riwayat Administrasi

Use Case Name: Menambah Riwayat Administrasi	ID: UC-20
Actor: Admin, Waka	
Preconditions: Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu riwayat administrasi 3. Sistem meminta data siswa kepada <i>database</i> 4. <i>Database</i> memberikan data siswa kepada sistem 5. Sistem menampilkan halaman daftar siswa 6. Admin memilih siswa 7. Sistem meminta data riwayat administrasi kepada <i>database</i> 8. <i>Database</i> memberikan data riwayat administrasi kepada sistem 9. Sistem menampilkan halaman riwayat administrasi 10. Admin memilih menu tambah riwayat administrasi 11. Sistem menampilkan halaman tambah riwayat administrasi 12. Admin mengisi form tambah riwayat administrasi 13. Sistem mengirim data riwayat administrasi ke <i>database</i> 14. <i>Database</i> menyimpan data riwayat administrasi 15. Sistem menampilkan halaman riwayat administrasi	
Postconditions: Admin berhasil menambahkan riwayat administrasi siswa	

Tabel 25. *Use Case Description* untuk Mengubah Riwayat Administrasi

Use Case Name: Mengubah Riwayat Administrasi	ID: UC-21
Actor: Admin, Waka	
Preconditions: Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu riwayat administrasi	

3. Sistem meminta data siswa kepada <i>database</i> 4. <i>Database</i> memberikan data siswa kepada sistem 5. Sistem menampilkan halaman daftar siswa 6. Admin memilih siswa 7. Sistem meminta data riwayat administrasi kepada <i>database</i> 8. <i>Database</i> memberikan data riwayat administrasi kepada sistem 9. Sistem menampilkan halaman riwayat administrasi 10. Admin memilih menu ubah riwayat administrasi 11. Sistem menampilkan halaman ubah riwayat administrasi 12. Admin mengisi form ubah riwayat administrasi 13. Sistem mengirim data riwayat administrasi ke <i>database</i> 14. <i>Database</i> menyimpan data riwayat administrasi 15. Sistem menampilkan halaman riwayat administrasi
Postconditions: Admin berhasil mengubah riwayat administrasi siswa

Tabel 26. *Use Case Description* untuk Menghapus Riwayat Administrasi

Use Case Name: Menghapus Riwayat Administrasi	ID: UC-22
Actor: Admin, Waka	
Preconditions: Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu riwayat administrasi 3. Sistem meminta data siswa kepada <i>database</i> 4. <i>Database</i> memberikan data siswa kepada sistem 5. Sistem menampilkan halaman daftar siswa 6. Admin memilih siswa 7. Sistem meminta data riwayat administrasi kepada <i>database</i> 8. <i>Database</i> memberikan data riwayat administrasi kepada sistem 9. Sistem menampilkan halaman riwayat administrasi 10. Admin memilih menu hapus riwayat administrasi 11. Sistem mengirim perintah hapus riwayat administrasi ke <i>database</i> 12. <i>Database</i> menghapus data riwayat administrasi 13. Sistem menampilkan halaman riwayat administrasi	
Postconditions: Admin berhasil menghapus riwayat administrasi siswa	

Tabel 27. *Use Case Description* untuk Melakukan Logout

Use Case Name: Melakukan Logout	ID: UC-23
Actor: Admin, Siswa, Guru, Waka	
Preconditions: Admin, Siswa, Guru harus <i>Login</i> kedalam sistem	
Normal Course:	

1. Admin, Siswa, dan Guru melakukan <i>Login</i> 2. Admin, Siswa, dan Guru memilih menu Logout 3. Sistem menampilkan konfirmasi logout 4. Admin, Siswa, dan Guru menekan tombol Logout 5. Sistem menghapus sesi 6. Sistem menampilkan halaman informasi publik
Postconditions: Admin, Siswa, dan Guru berhasil keluar dari sistem

Tabel 28. *Use Case Description* untuk Menambah Akun

Use Case Name: Menambah Akun	ID: UC-24
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu manajemen akun 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data akun kepada sistem 5. Sistem menampilkan halaman manajemen akun 6. Admin memilih menu tambah data akun 7. Sistem menampilkan halaman tambah akun 8. Admin menambah akun 9. Sistem mengirim data akun kepada <i>database</i> 10. <i>Database</i> menyimpan data akun 11. Sistem menampilkan halaman manajemen akun	
Postconditions: Admin berhasil menambah akun	

Tabel 29. *Use Case Description* untuk Mengubah Data Akun

Use Case Name: Mengubah Data Akun	ID: UC-25
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu manajemen akun 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data akun kepada sistem 5. Sistem menampilkan halaman manajemen akun 6. Admin memilih menu ubah data akun 7. Sistem menampilkan halaman ubah data akun 8. Admin mengubah data akun 9. Sistem mengirim data akun kepada <i>database</i> 10. <i>Database</i> menyimpan data akun	

11. Sistem menampilkan halaman manajemen akun
Postconditions: Admin berhasil mengubah data akun

Tabel 30. *Use Case Description* untuk Menghapus Data Akun

Use Case Name: Menghapus Akun	ID: UC-26
Actor: Admin, Waka	
Preconditions: Admin harus sudah <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu manajemen akun 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data akun kepada sistem 5. Sistem menampilkan halaman data akun 6. Admin memilih menu hapus data akun 7. Sistem mengirim perintah hapus data kepada <i>database</i> 8. <i>Database</i> menghapus data akun 9. Sistem menampilkan halaman manajemen akun	
Postconditions: Admin berhasil menghapus data akun	

Tabel 31. *Use Case Description* untuk Aktivasi Akun

Use Case Name: Aktivasi Akun	ID: UC-27
Actor: Admin, Waka	
Preconditions: Admin harus <i>Login</i> kedalam sistem	
Normal Course: 1. Admin melakukan <i>Login</i> 2. Admin memilih menu manajemen akun 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data akun kepada sistem 5. Sistem menampilkan halaman manajemen akun 6. Admin memilih menu aktivasi akun 7. Sistem menampilkan halaman aktivasi akun 8. Admin mengaktivasi akun 9. Sistem mengirim data kepada <i>database</i> 10. <i>Database</i> menyimpan data akun 11. Sistem menampilkan halaman manajemen akun	
Postconditions: Admin berhasil mengaktivasi akun	

Tabel 32. *Use Case Description* Mengatur Status Penilaian

Use Case Name: Mengatur Status Penilaian	ID: UC-28
Actor: Waka	
Preconditions: Waka harus <i>Login</i> kedalam sistem	
Normal Course: 1. Waka melakukan <i>Login</i> 2. Waka memilih menu nilai 3. Sistem meminta data kepada <i>database</i> 4. <i>Database</i> memberikan hasil pencarian data siswa kepada sistem 5. Sistem menampilkan halaman daftar siswa 6. Waka mengatur status penilaian 7. Sistem mengakses <i>database</i> 8. <i>Database</i> menyimpan status penilaian 9. Sistem menampilkan halaman daftar siswa	
Postconditions: Waka berhasil mengatur status penilaian	

Tabel 33. *Use Case Description* Reset Password

Use Case Name: Reset Password	ID: UC-29
Actor: Admin, Guru, Siswa, Waka	
Preconditions: Admin, Guru, Siswa, Waka tidak harus <i>Login</i> kedalam sistem	
Normal Course: 1. Admin, Guru, Siswa, Waka memilih menu nilai 2. Sistem menampilkan halaman Login 3. Admin, Guru, Siswa, Waka memilih menu lupa password 4. Sistem menampilkan form kirim email 5. Admin, Guru, Siswa, Waka mengisi form kirim email 6. Sistem mengakses <i>database</i> 7. <i>Database</i> mencari email yang di inputkan 8. Sistem menampilkan halaman reset password 9. Admin, Guru, Siswa, Waka mengisi form reset password 10. Sistem mengakses <i>database</i> 11. Data user di perbarui 12. Sistem menampilkan halaman informasi publik	
Postconditions: Admin, Guru, Siswa, Waka berhasil mengubah password	

F. Strategi Analisis Kebutuhan

Pemetaan Kebutuhan dengan Proses *As-Is* dan *To-Be-System*

Berikut ini adalah pemetaan kebutuhan (baik fungsional

maupun nonfungsional) dengan proses *As-Is* dan proses *To-Be*.

Tabel 34. Matriks Kebutuhan Fungsional *As-Is* dan *To-Be*

Kebutuhan Fungsional		
No.	Proses berjalan (<i>As-Is</i>)	Sistem yang akan dibuat (<i>To-Be</i>)
1.	Admin mengelola informasi publik (Sejarah singkat, galeri kegiatan, prestasi, ekstrakurikuler) secara manual menggunakan pengumuman sederhana, dokumen kertas atau diupdate di media sosial.	Admin dapat mengelola dan memperbarui informasi publik (Sejarah singkat, galeri kegiatan, prestasi, ekstrakurikuler) melalui sistem yang terintegrasi dan dapat diakses secara online.
2.	Siswa, guru dan admin mendapatkan informasi (Sejarah singkat, galeri kegiatan, prestasi, ekstrakurikuler, visi misi, hymne, struktur organisasi) hanya melalui pengumuman langsung atau informasi yang terbatas pada media sosial.	Siswa, guru, dan admin dapat mengakses informasi publik (Sejarah singkat, galeri kegiatan, prestasi, ekstrakurikuler, visi misi, hymne, struktur organisasi) kapan saja dan dari mana saja melalui <i>website</i> yang menyediakan informasi terstruktur dan transparan.
3.	Guru dan admin mencatat dan mengolah nilai secara manual menggunakan kertas atau spreadsheet, yang rentan terhadap kesalahan dan membutuhkan waktu lebih lama untuk pengolahan data.	Guru dan admin dapat langsung menginput, mengubah, atau menghapus nilai melalui sistem, guna mengurangi risiko kesalahan dan meningkatkan efisiensi.

4.	Siswa hanya dapat melihat nilai, ranking, kemajuan belajar dan <i>raport</i> pada saat penerimaan rapor akhir semester, dan tidak dapat memantau hasil belajar secara <i>real-time</i>	Siswa dapat memantau kemajuan belajar, nilai, <i>raport</i> , dan <i>ranking</i> siswa secara langsung dapat diakses dengan cepat melalui sistem yang terintegrasi, memungkinkan pemantauan <i>real-time</i> , akurat dan transparansi yang lebih tinggi.
5.	Admin mengelola data akademik, seperti data siswa, data guru pegawai, jadwal pelajaran dan riwayat administrasi secara manual menggunakan dokumen fisik atau spreadsheet sederhana, yang rentan terhadap kesalahan dan memakan waktu.	Admin dapat mengelola data akademik (data siswa, data guru pegawai, jadwal pelajaran dan riwayat administrasi) secara terintegrasi dalam sistem berbasis <i>web</i> , yang memastikan data lebih aman, akurat, dan dapat diakses dengan cepat.
6.	Guru dan admin mengelola <i>ranking</i> , kemajuan belajar dan <i>raport</i> kelas secara manual yang memakan lebih banyak waktu	Guru dan admin bisa langsung melihat data kemajuan belajar, <i>ranking</i> dan <i>raport</i> , karena berdasarkan nilai, <i>ranking</i> kelas, <i>ranking</i> angkatan, kemajuan belajar dan data <i>raport</i> akan otomatis terhitung dan ter- <i>update</i> data nya secara <i>real-time</i>
7.	Siswa dan guru hanya dapat melihat data akademik seperti data siswa, guru, jadwal pelajaran dan riwayat administrasi secara terbatas, tidak ada sistem yang memungkinkan untuk mengakses data akademik kapan saja.	Siswa dan guru dapat melihat data akademik secara langsung melalui sistem berbasis <i>web</i> kapan saja dan di mana saja

Tabel 35. Matriks Kebutuhan Non-Fungsional *As-Is* dan *To-Be*

Kebutuhan Non-Fungsional		
No.	Proses berjalan (<i>As-Is</i>)	Sistem yang akan dibuat (<i>To-Be</i>)
1.	Persyaratan Operasional	
	- Sistem pengelolaan data akademik hanya tersedia dalam bentuk manual atau semi-digital, seperti menggunakan dokumen cetak atau aplikasi desktop lokal.	Sistem harus dapat diakses secara <i>online</i> baik melalui <i>website</i> .
	- Pengguna hanya dapat mengakses sistem di komputer yang tersedia di ruang administrasi sekolah, tanpa dukungan perangkat <i>mobile</i> atau akses <i>online</i> .	Sistem harus dapat dijalankan di perangkat <i>smartphone</i> , tablet, maupun komputer.
2.	Persyaratan Kinerja	
	- Proses pengelolaan nilai dan data akademik membutuhkan waktu lama karena dilakukan secara manual atau menggunakan perangkat lunak sederhana tanpa optimasi fitur.	Sistem harus mampu memproses penggunaan fitur, seperti akses data akademik dan unggahan nilai pembelajaran, dalam waktu tidak lebih dari 30 detik per permintaan.
3.	Persyaratan Keandalan	
	- Layanan tidak dapat diakses kapanpun, hanya	Sistem harus memiliki <i>uptime</i> 99.9% untuk memastikan

	dapat diakses saat jam kerja saja.	ketersediaan layanan yang konsisten bagi pengguna.
4.	Persyaratan Keselamatan	
	- Belum ada Sistem informasi akademik di sekolah.	Enkripsi data harus diterapkan untuk semua data sensitif seperti <i>password</i> akun pengguna.
5.	Persyaratan Keamanan	
	- Otorisasi belum dilakukan oleh sistem, sehingga terbilang lebih rentan.	Sistem harus memastikan bahwa setiap fitur hanya dapat diakses oleh pengguna yang memiliki peran dan otorisasi sesuai dengan hak akses yang telah ditentukan.
6.	Persyaratan Kegunaan	
	- Proses masih dilakukan secara manual atau belum terdigitalisasi, sehingga belum ada antar muka sistem.	Antarmuka pengguna harus intuitif dan mudah digunakan, dengan desain yang responsif untuk berbagai perangkat.

3. Coding (Pengkodean)

Sistem informasi akademik (SIAKAD) di SMPN 29 Bandar Lampung dibangun menggunakan *framework* CodeIgniter 4 dengan bahasa pemrograman PHP. Sistem ini terdiri dari empat modul utama, yaitu modul Informasi Publik, Data Akademik, Pembelajaran, dan Evaluasi. Proses pengembangan sistem dibagi berdasarkan modul, di mana saya bertanggung jawab dalam mengerjakan modul Data Akademik dan Pembelajaran, sedangkan modul Informasi Publik dan Evaluasi dikembangkan oleh Enjelita Aini Natasya. Pembagian tugas ini

dilakukan untuk mempercepat proses pengembangan dan memastikan setiap modul dapat berjalan secara optimal.

4. *Testing* (Pengujian)

Tahap pengujian merupakan langkah terakhir dalam metode Extreme Programming dan memiliki peran penting dalam memastikan kualitas sistem. Pada tahap ini, pengguna dilibatkan secara langsung untuk menilai apakah sistem yang dikembangkan telah memenuhi kebutuhan yang diharapkan. Dalam pelaksanaan pengujian pada penelitian ini, digunakan dua metode, yaitu Black Box Testing dan User Acceptance Testing (UAT). Black Box Testing difokuskan pada pengujian fungsional sistem dengan cara memberikan input dan mengamati output tanpa memperhatikan struktur internal program. Sedangkan UAT dilakukan dengan melibatkan pengguna untuk mengevaluasi apakah sistem telah berjalan sesuai dengan tujuan dan dapat diterima untuk digunakan dalam lingkungan sebenarnya. Skenario pengujian Black Box dan UAT dapat dilihat pada tabel 36 dan 37.

Tabel 36. Skenario Black Box testing

No.	Kelas Pengujian	ID Kasus Pengujian	Kasus Pengujian	Hasil Yang diharapkan
1.	Login	UC01-TC01	Mengakses halaman Login	Menampilkan form Login
		UC01-TC02	Memasukan username dan password benar	Berhasil masuk ke dalam sistem
		UC01-TC03	Memasukan username dan password salah	Menampilkan pesan bahwa username dan password salah

2.	Logout	UC23-TC01	Menekan menu Logout	Menampilkan konfirmasi Logout
		UC23-TC02	Menekan tombol konfirmasi logout	Berhasil keluar dari sistem
3.	Menu Data Siswa	UC02-TC01	Menekan menu data siswa	Menampilkan halaman data siswa
		UC014-TC01	Menekan tombol tambah data siswa	Menampilkan form tambah data siswa
		UC014-TC02	Mengisi data siswa dengan input valid	Data berhasil disimpan dan menampilkan semua data siswa
		UC014-TC03	Mengisi form dengan data kosong	Sistem menampilkan pesan error validasi
		UC015-TC01	Menekan tombol edit data siswa	Menampilkan form edit data siswa
		UC015-TC02	Mengedit data dengan input valid	Data berhasil disimpan dan menampilkan semua data siswa
		UC015-TC03	Mengedit data dengan mengosongkan field	Sistem menampilkan pesan error validasi
		UC016-TC01	Menekan tombol hapus data siswa	Data berhasil dihapus dan menampilkan semua data siswa
4.	Menu Data Guru dan Pegawai	UC03-TC01	Menekan menu data guru dan pegawai	Menampilkan halaman data guru dan pegawai
		UC017-TC01	Menekan tombol tambah data guru	Menampilkan form tambah data guru

		UC017-TC02	Mengisi data guru dengan input valid	Data berhasil disimpan dan menampilkan semua data guru dan pegawai
		UC017-TC03	Mengisi form tambah data guru dengan data kosong	Sistem menampilkan pesan error validasi
		UC018-TC01	Menekan tombol edit data guru	Menampilkan form edit data guru
		UC018-TC02	Mengedit data guru dengan input valid	Data berhasil disimpan dan menampilkan semua data guru dan pegawai
		UC018-TC03	Mengedit data guru dengan mengosongkan field	Sistem menampilkan pesan error validasi
		UC019-TC01	Menekan tombol hapus data guru	Data berhasil dihapus dan menampilkan semua data guru dan pegawai
		UC017-TC04	Menekan tombol tambah data pegawai	Menampilkan form tambah data pegawai
		UC017-TC05	Mengisi data pegawai dengan input valid	Data berhasil disimpan dan menampilkan semua data guru dan pegawai

		UC017-TC06	Mengisi form tambah data pegawai dengan data kosong	Sistem menampilkan pesan error validasi
		UC018-TC04	Menekan tombol edit data pegawai	Menampilkan form edit data pegawai
		UC018-TC05	Mengedit data pegawai dengan input valid	Data berhasil disimpan dan menampilkan semua data guru dan pegawai
		UC018-TC06	Mengedit data pegawai dengan mengosongkan field	Sistem menampilkan pesan error validasi
		UC019-TC02	Menekan tombol hapus data pegawai	Data berhasil dihapus dan menampilkan semua data guru dan pegawai
5.	Menu Jadwal Mata Pelajaran	UC05-TC01	Menekan menu jadwal mata pelajaran	Menampilkan halaman jadwal mata pelajaran
		UC11-TC01	Menekan tombol tambah jadwal mata pelajaran	Menampilkan form tambah jadwal mata pelajaran
		UC11-TC02	Mengisi data jadwal mata pelajaran dengan input valid	Data berhasil disimpan dan menampilkan semua jadwal mata pelajaran
		UC11-TC03	Mengisi form dengan data kosong	Sistem menampilkan pesan error validasi

		UC12-TC01	Menekan tombol edit jadwal mata pelajaran	Menampilkan form edit jadwal mata pelajaran
		UC12-TC02	Mengedit data dengan input valid	Data berhasil disimpan dan menampilkan semua jadwal mata pelajaran
		UC12-TC03	Mengedit data dengan mengosongkan field	Sistem menampilkan pesan error validasi
		UC13-TC01	Menekan tombol hapus jadwal mata pelajaran	Data berhasil dihapus dan menampilkan semua jadwal mata pelajaran
6.	Menu Riwayat Administrasi	UC04-TC01	Menekan menu riwayat administrasi	Menampilkan data siswa pada halaman riwayat administrasi
		UC04-TC02	Memilih nama salah satu siswa pada halaman riwayat administrasi	Menampilkan data riwayat administrasi siswa yang telah dipilih
		UC20-TC01	Menekan tombol tambah riwayat administrasi	Menampilkan form tambah riwayat administrasi
		UC20-TC02	Mengisi data riwayat administrasi dengan input valid	Data berhasil disimpan dan menampilkan semua riwayat administrasi
		UC20-TC03	Mengisi form dengan data kosong	Sistem menampilkan pesan error validasi

		UC21-TC01	Menekan tombol edit riwayat administrasi	Menampilkan form edit riwayat administrasi
		UC21-TC02	Mengedit data dengan input valid	Data berhasil disimpan dan menampilkan semua riwayat administrasi
		UC21-TC03	Mengedit data dengan mengosongkan field	Sistem menampilkan pesan error validasi
		UC22-TC01	Menekan tombol hapus riwayat administrasi	Data berhasil dihapus dan menampilkan semua riwayat administrasi
7.	Menu Nilai	UC06-TC01	Menekan menu nilai	Menampilkan data siswa pada halaman nilai
		UC06-TC02	Memilih nama salah satu siswa pada halaman nilai	Menampilkan data nilai siswa yang telah dipilih
		UC08-TC01	Menekan tombol tambah nilai	Menampilkan form tambah nilai
		UC08-TC02	Mengisi data nilai dengan input valid	Data berhasil disimpan dan menampilkan semua nilai
		UC08-TC03	Mengisi form dengan data kosong	Sistem menampilkan pesan error validasi
		UC09-TC01	Menekan tombol edit nilai	Menampilkan form edit nilai

		UC09-TC02	Mengedit data dengan input valid	Data berhasil disimpan dan menampilkan semua nilai
		UC09-TC03	Mengedit data dengan mengosongkan field	Sistem menampilkan pesan error validasi
		UC10-TC01	Menekan tombol hapus nilai	Data berhasil dihapus dan menampilkan semua nilai
8.	Menu Kemajuan Belajar	UC01-TC01	Menekan menu kemajuan belajar	Menampilkan data siswa pada halaman kemajuan belajar
			Memilih nama salah satu siswa pada halaman kemajuan belajar	Menampilkan data kemajuan belajar siswa yang telah dipilih
9.	Menu Manajemen Akun	UC24-TC01	Menekan menu manajemen akun	Menampilkan halaman manajemen akun
		UC24-TC02	Menekan tombol tambah akun	Menampilkan form tambah akun
		UC24-TC03	Mengisi data akun dengan input valid	Data berhasil disimpan dan menampilkan semua data akun
		UC24-TC04	Mengisi form dengan data kosong	Sistem menampilkan pesan error validasi
		UC25-TC01	Menekan tombol edit data akun	Menampilkan form edit data akun

		UC25-TC02	Mengedit data dengan input valid	Data berhasil disimpan dan menampilkan semua data akun
		UC25-TC03	Mengedit data dengan mengosongkan field	Sistem menampilkan pesan error validasi
		UC26-TC01	Menekan tombol hapus data akun	Data berhasil dihapus dan menampilkan semua data akun

Tabel 37. Skenario User Acceptance Testing

No.	Pernyataan	Skala
1.	Penggunaan warna tulisan dengan latar belakang sudah sesuai.	1-5
2.	Menu atau fitur yang tersedia di Sistem Informasi Akademik ini mudah dipahami.	1-5
3.	Waktu yang dibutuhkan untuk mengoperasikan sistem ini cukup efisien.	1-5
4.	Fitur-fitur dan tombol yang disediakan dalam sistem berfungsi dengan baik.	1-5
5.	Sistem ini membantu meningkatkan efektivitas kerja pengguna.	1-5
6.	Respon sistem saat digunakan terasa cepat dan stabil.	1-5
7.	Sistem memiliki tampilan yang menarik dan tidak membosankan.	1-5
8.	Sistem mampu berfungsi dengan baik di berbagai perangkat atau browser yang berbeda.	1-5

3.2.3. Tahap Penulisan Laporan

Tahap penulisan laporan bertujuan untuk menyampaikan hasil dari penelitian yang telah dilakukan serta mendokumentasikan seluruh rangkaian kegiatan secara menyeluruh. Laporan ini diharapkan dapat membantu pembaca dalam memahami proses yang telah dilalui dan menjadi acuan bagi penelitian di masa mendatang.

V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berikut adalah beberapa kesimpulan utama dari penelitian ini:

1. Penelitian ini berhasil mengembangkan sistem informasi akademik berbasis web di SMPN 29 Bandar Lampung yang mempermudah akses informasi pembelajaran secara cepat dan real-time. Sistem ini mendukung transparansi belajar dengan memungkinkan siswa, guru, dan orang tua mengakses informasi pembelajaran secara jelas dan terbuka.
2. Sistem ini mempermudah pengelolaan data oleh staf administrasi sekolah, sehingga dapat meningkatkan efisiensi dalam pemantauan data akademik. Hal ini mendukung pengambilan keputusan yang lebih tepat dan memungkinkan seluruh pemangku kepentingan untuk mengakses data akademik secara terbuka dan *real-time*.

5.2. Saran

Berdasarkan implementasi sistem yang telah dikembangkan, terdapat beberapa saran untuk pengembangan lebih lanjut agar sistem dapat menjadi lebih baik:

1. Menambahkan fitur kehadiran siswa, agar sistem tidak hanya fokus pada data penilaian, tetapi juga mendukung pengelolaan kegiatan belajar mengajar secara lebih menyeluruh.
2. Melakukan pengujian yang lebih mendalam terhadap seluruh sistem untuk memastikan kualitas dan keandalannya.

DAFTAR PUSTAKA

- Alfian, A. N., Rifai, Y., Wahyuni, R., Rika, A., Lestari, I. A., & Putri, D. I. (2024). *User Acceptance Test dan Penerapan Model ADDIE pada Media Pembelajaran Interaktif Pengenalan Profesi*. 265–271.
- Ardiansah, T., Rahmanto, Y., & Amir, Z. (2023). Penerapan Extreme Programming Dalam Sistem Informasi Akademik SDN Kuala Teladas. *Journal of Information Technology, Software Engineering and Computer Science (ITSECS)*, 1(2), 44–51. <https://doi.org/10.58602/itsecs.v1i2.25>
- Chamida, M. A., Susanto, A., & Latubessy, A. (2021). Analisa User Acceptance Testing Terhadap Sistem Informasi Pengelolaan Bedah Rumah Di Dinas Perumahan Rakyat Dan Kawasan Permukiman Kabupaten Jepara. *Indonesian Journal of Technology, Informatics and Science (IJTIS)*, 3(1), 36–41. <https://doi.org/10.24176/ijtis.v3i1.7531>
- Dristyan, F., Syafnur, A., & Tinggi Manajemen Informatika dan Komputer Royal, S. (2021). SISTEM INFORMASI GEOGRAFIS PENYEBARAN COVID-19 DI KABUPATEN ASAHAN MENGGUNAKAN FRAMEWORK CODEIGNITER 4. In *Journal of Science and Social Research* (Issue 3). <http://jurnal.goretanpena.com/index.php/JSSR>
- Hakim, Z., & Meilina, P. (2022). Sistem Informasi Akademik Berbasis Webiste (Studi Kasus : Smpit Avicenna). *JUST IT : Jurnal Sistem Informasi, Teknologi Informasi Dan Komputer*, 12(3), 32–37.
- Jacobson, L., & Booch, J. (2021). The Unified Modeling Language Reference Manual. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).

- Kustiawan, D., Cholifah, W. N., Destriana, R., & Heriyani, N. (2022). Rancang Bangun Sistem Informasi Akuntansi Pengelolaan Koperasi Menggunakan Metode Extreme Programming. *Jurnal Teknologi Dan Informasi*, 12(1), 78–92. <https://doi.org/10.34010/jati.v12i1.6756>
- Lukman Santoso, & Juni Amanullah. (2022). Pengembangan Sistem Informasi Akademik Berbasis Website Menggunakan Metode Rapid Application Development (Rad). *Elkom : Jurnal Elektronika Dan Komputer*, 15(2), 250–259. <https://doi.org/10.51903/elkom.v15i2.943>
- Muhamad Solahudin. (2021). *Rancang Bangun Sistem Informasi Akademik Sekolah (SIAS) Berbasis Website*.
- Putri Irianti, A., Kurnia, W., & Penulis Korespondensi Submitted, N. (2023). *Sistem Informasi Perpustakaan Berbasis Website pada MAN 2 Bandar Lampung*. x, 192–197. <https://doi.org/10.33365/jtsi.v4i2.2573>
- Rahmadani, T. P., Siswanto, A., Yani, H., Masgo, & Santoso. (2022). Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM) Perancangan Sistem Informasi Akademik Berbasis Web Pada SMP N 1 Muaro Jambi. *Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)*, 2(2), 305.
- Ramdany, S. (2024). Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web. *Journal of Industrial and Engineering System*, 5(1). <https://doi.org/10.31599/2e9afp31>
- Sahat, P., & Sihombing, M. (2023). *Pengembangan Sistem Informasi Akademik Berbasis Web di SMP Negeri 3 Sidikalang*. 4(2), 33–36.
- Sari, N., & Cahyani, D. (2022). Perancangan Sistem Informasi Monitoring Sertifikat Menggunakan Extreme Programming. *Jurnal Ilmiah Computer Science*, 1(1), 1–6. <https://doi.org/10.58602/jics.v1i1.1>
- Setiawan, A., Samsugi, S., & Alita, D. (2023). Rancang Bangun Sistem Informasi Akademik SMK Taman Siswa 1 Tanjung Karang Berbasis Web. *Jurnal*

Informatika Dan Rekayasa Perangkat Lunak, 4(1), 53–59.

<https://doi.org/10.33365/jatika.v4i1.2465>

Sitanggang Rianto, Urian Dachi Teddy, & Manurung H G Immanuel. (2022).

Rancang Bangun Sistem Penjualan Tanaman Hiasberbasis Web

Menggunakan Php Dan Mysql. *Tekesnos*, 4(1), 84–90.

Sutabri. (2021). *Sistem Informasi Akademik Berbasis Web Untuk Sma Bina Warga*

1 Palembang Sistem Informasi Akademik Berbasis Web Untuk Sma Bina

Warga 1 Palembang.

Syaebani, A., Tyasmala, D. V., Maulani, R., Utami, E. D., & Wahyuni, S. N.

(2021). Pengembangan Sistem Informasi Pelayanan Surat Menyurat (Sira)

Berbasis Website Dengan Menggunakan Framework Codeigniter. *Journal of Information System Management (JOISM)*, 3(2), 59–65.

<https://doi.org/10.24076/joism.2021v3i2.446>

Uminingsih, Nur Ichsanudin, M., Yusuf, M., & Suraya, S. (2022). Pengujian

Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode

Black Box Testing Bagi Pemula. *STORAGE: Jurnal Ilmiah Teknik Dan Ilmu*

Komputer, 1(2), 1–8. <https://doi.org/10.55123/storage.v1i2.270>