

**RANCANG BANGUN SISTEM INFORMASI ADMINISTRASI
KEUANGAN SMA PLUS NURUL IMAN LELES BERBASIS
WEB MENGGUNAKAN METODE *SCRUM***

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Syarat Penulisan Skripsi Pada Program
Studi Sistem Informasi Institut Pendidikan Indonesia Garut

Oleh

**RIZWAN ALAMSYAH
NIM 20576021**



**SISTEM INFORMASI
FAKULTAS ILMU TERAPAN DAN SAINS
INSTITUT PENDIDIKAN INDONESIA GARUT**

2024

LEMBAR PENGESAHAN SKRIPSI

**RANCANG BANGUN SISTEM INFORMASI ADMINISTRASI
KEUANGAN SMA PLUS NURUL IMAN LELES BERBASIS
WEB MENGGUNAKAN METODE *SCRUM***

oleh

Rizwan Alamsyah
NIM 20576021

disetujui dan disahkan oleh:

Pembingbing I

Pembingbing II

Yopi Nugraha, M.Kom.
NIDN. 0412029202

Dinar Rahayu, S.ST., M.Kom.
NIDN. 0412109007

diketahui oleh

Ketua Program Studi
Sistem Informasi
Institut Pendidikan Indonesia Garut

Yopi Nugraha, M.Kom.
NIDN. 0412029202

LEMBAR PENGUJIAN SKRIPSI

**RANCANG BANGUN SISTEM INFORMASI ADMINISTRASI
KEUANGAN SMA PLUS NURUL IMAN LELES BERBASIS
WEB MENGGUNAKAN METODE *SCRUM***

oleh

Rizwan Alamsyah
NIM. 20576021

Skripsi ini telah diujikan pada tanggal 31 Mei 2024

Penguji I

Penguji II

Penguji III

Tedi Budiman, S.Si., M.Kom.
NIDN. 0410067601

Purnomo Sidiq, M.Kom.
NIDN. 0407088007

Dinar Rahayu, S.ST., M.Kom.
NIDN. 0412109007

diketahui oleh

Dekan
Fakultas Ilmu Terapan dan Sains
Institut Pendidikan Indonesia Garut

Ketua Program Studi
Sistem Informasi
Institut Pendidikan Indonesia Garut

Dr. Lida Amalia, M.Pd.
NIP. 19660214199403200

Yopi Nugraha M.Kom.
NIDN. 0412029202

MOTTO

“Apapun yang menjadi takdirmu, akan mencari jalannya menemukanmu”

Ali bin Abi Thalib

“Jika kamu takut pada sesuatu, jatuhkanlah dirimu ke dalamnya, karena perasaan takut yang hebat lebih kuat dari sesuatu yang kamu takuti”

Ali bin Abi Thalib

إِنَّمَا الْعُسْرُ يُسْرًا

“Sesungguhnya bersama kesulitan ada kemudahan”

Q.S Asy-Syarh:6

PERNYATAAN KEASLIAN SKRIPSI

Dengan ini penulis menyatakan bahwa skripsi yang berjudul “Rancang Bangun Sistem Informasi Administrasi Keuangan SMA Plus Nurul Iman Leles Berbasis *Web* Menggunakan Metode *Scrum*” adalah hasil karya penulis pribadi. Adapun terdapat tulisan yang penulis kutip dalam laporan ini sudah penulis tuliskan sesuai aturan kutipan yang berlaku. Apabila penulis melakukan plagiat, baik secara sengaja ataupun tidak, maka penulis menyatakan akan menarik skripsi yang penulis ajukan. Apabila kemudian hari penulis terbukti melakukan penyalinan atau meniru tulisan milik orang lain yang diklaim hasil pemikiran penulis sendiri, berarti penulis bersedia melepas semua hak yang telah diberikan oleh Institut Pendidikan Indonesia.

Garut, Maret 2024
Penulis,

Rizwan Alamsyah
NIM.20576021

ABSTRAK

SMA Plus Nurul Iman Leles telah menunjukkan pendekatan yang belum terkomputerisasi dalam pengelolaan administrasi keuangannya sejak awal berdiri. Proses pencatatan atau penginputan data keuangan masih dilakukan secara manual, menyebabkan berbagai permasalahan, termasuk keterlambatan dalam pencarian data dan kerapnya kehilangan data. Tujuan penelitian ini, untuk membuat sebuah sistem informasi administrasi keuangan berbasis website yang dapat mempermudah pengelolaan administrasi, khususnya dalam penginputan data keuangan dan pencarian data. Penelitian dilaksanakan untuk mengatasi permasalahan yang timbul dari sistem manual, seperti kesalahan pencatatan, kehilangan data, dan inefisiensi dalam proses pengelolaan data keuangan. Metode penelitian dan pengembangan yang digunakan adalah metode *Scrum*, yang menekankan kecepatan, fleksibilitas, dan hasil akhir yang terukur. Metode ini melibatkan serangkaian iterasi pendek yang memungkinkan untuk secara bertahap mengembangkan dan memperbaiki sistem berdasarkan umpan balik yang diterima. Dalam penelitian ini, penulis berhasil menemukan bahwa implementasi sistem informasi administrasi keuangan berbasis *website* dapat meningkatkan efektivitas dan efisiensi kerja dengan mengotomatisasi proses administrasi yang sebelumnya dilakukan secara manual. Temuan penelitian menunjukkan bahwa sistem ini tidak hanya mengurangi kesalahan pencatatan dan mempercepat proses pengelolaan data, tetapi juga memberikan transparansi dan akurasi yang lebih tinggi dalam pengelolaan keuangan sekolah. Selain itu, sistem ini juga mempermudah akses terhadap informasi keuangan, memungkinkan pengguna untuk dengan cepat mengakses dan memverifikasi data keuangan yang diperlukan.

Kata kunci: Sistem Informasi, Administrasi Keuangan, Metode *Scrum*, *Website*.

ABSTRACT

SMA Plus Nurul Iman Leles has shown an uncomputerized approach in managing its financial administration since its inception. The process of recording or inputting financial data is still carried out manually, causing various problems, including delays in data retrieval and frequent data loss. The purpose of this research is to create a website-based financial administration information system that can facilitate administrative management, especially in financial data input and data search. The research was carried out to overcome problems arising from manual systems, such as recording errors, data loss, and inefficiencies in the financial data management process. The research and development method used is the Scrum method, which emphasizes speed, flexibility, and measurable deliverables. This method involves a series of short iterations that make it possible to gradually develop and improve the system based on the feedback received. In this study, the author managed to find that the implementation of a website-based financial administration information system can increase work effectiveness and efficiency by automating administrative processes that were previously done manually. The findings of the study show that this system not only reduces recording errors and speeds up the data management process, but also provides greater transparency and accuracy in school financial management. In addition, the system also makes it easier to access financial information, allowing users to quickly access and verify the necessary financial data.

Keywords: *Information Systems, Financial Administration, Scrum Methods, Website.*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT, yang sudah melimpahkan rahmat dan nikmat-Nya, sehingga mampu menyelesaikan penulisan skripsi yang berjudul “Rancang Bangun Sistem Informasi Administrasi Keuangan SMA Plus Nurul Iman Leles Berbasis *Web* Menggunakan Metode *Scrum*”. Sholawat serta salam semoga selalu tercurah limpahkan kepada Nabi Muhammad SAW, beserta sahabat, keluarga, dan para pengikutnya.

Penulisan skripsi ini bertujuan memenuhi sebagian syarat supaya bisa memperoleh gelar Sarjana Komputer pada program studi Sistem Informasi, Fakultas Ilmu Terapan dan Sains, Institut Pendidikan Indonesia Garut.

Penulis menyadari bahwa, penulisan pada skripsi ini bukan karena keberhasilan individual saja. Dengan hal ini, penulis mengungkapkan terimakasih yang begitu besar kepada kedua orang tua yang selalu memberikan do’a dan motivasi. Selain itu, penulis mengucapkan terimakasih kepada yang terhormat:

1. Bapak Prof. Dr. Nizar Alam Hamdani, M.M., M.T., M.Si., M.Kom., selaku Rektor Institut Pendidikan Indonesia Garut.
2. Ibu Dr. Lida Amalia, M.Si., selaku Dekan Fakultas Ilmu Terapan dan Sains Institut Pendidikan Indonesia Garut.
3. Bapak Yopi Nugraha, M.Kom., selaku Ketua Program Studi Sistem Informasi Institut Pendidikan Indonesia Garut, sekaligus dosen pembimbing I yang senantiasa memberikan arahan, bimbingan dan serta telah meluangkan waktu dalam penulisan skripsi.
4. Ibu Dinar Rahayu, S.ST., M.Kom., selaku dosen pembimbing II yang senantiasa memberikan arahan, bimbingan dan serta telah meluangkan waktu dalam penulisan skripsi.
5. Semua Pihak SMA Plus Nurul Iman yang telah bersedia dijadikan sebagai objek penelitian.
6. Para dosen Program Studi Sistem Informasi yang sudah memberikan ilmu selama masa perkuliahan di Insitut Pendidikan Indonesia Garut.
7. Teman-teman satu angkatan Program Studi Sistem Informasi yang sudah merangkai cerita pada masa perkuliahan.

8. Para media, penulis atau pemilik jurnal maupun buku yang telah memberikan referensi bagi penulis.

Serta pihak-pihak yang tidak bisa disebutkan semuanya dan sudah membantu proses dalam penyelesaian skripsi ini dengan lancar dan baik

Akhir kata penulis sampaikan, semoga skripsi ini bisa memberikan banyak manfaat bagi pembaca sekalian pada umumnya, dan khususnya bagi pihak-pihak yang memiliki kepentingan.

Garut, Maret 2024

Rizwan Alamsyah
NIM.20576021

UCAPAN TERIMA KASIH

Skripsi ini penulis persembahkan untuk keluarga, terutama kepada orang tua yang selalu memberikan dukungan baik secara moril maupun materil. Mereka senantiasa mendorong semangat dan do'a dari awal sampai saat ini. Penulis mengucapkan banyak terima kasih, atas segala pengorbanan, cinta, dan perhatian yang selama ini diberikan.

Penulis ingin menyampaikan rasa terima kasih kepada para dosen pembimbing, yang memberikan bimbingan, ilmu, dan waktu dalam menyelesaikan penulisan skripsi ini. Terima kasih juga disampaikan kepada teman-teman yang selalu memberikan dukungan, motivasi, dan kebersamaan selama masa kuliah.

Mudah-mudahan skripsi ini bisa bermanfaat dan berkontribusi positif bagi para pembaca dan bisa membantu dalam pengembangan ilmu pengetahuan di masa mendatang. Penulis sadar, bahwa masih adanya kekurangan di dalam penulisan skripsi ini. Karena itu kritik dan sarannya yang konstruktif sangat diharapkan demi perbaikan di waktu yang akan datang.

DAFTAR TABEL

Tabel 2. 1 Use Case Diagram	16
Tabel 2. 2 Activity Diagram	18
Tabel 2. 3 Sequence Diagram	19
Tabel 2. 4 Class Diagram.....	20
Tabel 2. 5 Penelitian Terdahulu	26
Tabel 4. 1 Analisis Kelemahan Sistem	42
Tabel 4. 2 Fitur Aplikasi	44
Tabel 4. 3 Pembagian Sprint.....	45
Tabel 4. 4 Kelengkapan Sprint	45
Tabel 4. 5 User.....	57
Tabel 4. 6 Siswa.....	58
Tabel 4. 7 Guru	59
Tabel 4. 8 Kelas	59
Tabel 4. 9 Jurusan	60
Tabel 4. 10 Jabatan	60
Tabel 4. 11 Agama	60
Tabel 4. 12 Gender.....	61
Tabel 4. 13 Transaksi.....	61
Tabel 4. 14 Pembayaran.....	61
Tabel 4. 15 Histori Pembayaran	62
Tabel 4. 16 Pemasukan	62
Tabel 4. 17 Pengeluaran.....	63
Tabel 4. 18 Session	63
Tabel 4. 19 Kas Guru.....	64
Tabel 4. 20 Pengambilan Kas	64
Tabel 4. 21 Pengujian Fitur Login.....	89
Tabel 4. 22 Pengujian Fitur Verifikasi Email.....	90
Tabel 4. 23 Pengujian Fitur Reset Password	91
Tabel 4. 24 Pengujian Fitur Tambah Data Siswa.....	92
Tabel 4. 25 Pengujian Fitur Ubah Data Siswa.....	92
Tabel 4. 26 Pengujian Fitur Hapus Data Siswa	93

Tabel 4. 27 Pengujian Fitur Detail Data Siswa.....	94
Tabel 4. 28 Pengujian Fitur Pencarian Data Siswa.....	94
Tabel 4. 29 Pengujian Fitur Download Data Siswa.....	95
Tabel 4. 30 Pengujian Fitur Tambah Data Kelas	95
Tabel 4. 31 Pengujian Fitur Ubah Data Kelas	96
Tabel 4. 32 Pengujian Fitur Hapus Data Kelas.....	96
Tabel 4. 33 Pengujian Fitur Pencarian Data Kelas	97
Tabel 4. 34 Pengujian Fitur Tambah Data Jurusan	97
Tabel 4. 35 Pengujian Fitur Ubah Data Jurusan	98
Tabel 4. 36 Pengujian Fitur Hapus Data Jurusan.....	98
Tabel 4. 37 Pengujian Fitur Pencarian Data Jurusan	99
Tabel 4. 38 Pengujian Fitur Tambah Data Guru	99
Tabel 4. 39 Pengujian Fitur Ubah Data Guru	100
Tabel 4. 40 Pengujian Fitur Hapus Data Guru.....	100
Tabel 4. 41 Pengujian Fitur Detail Data Guru	101
Tabel 4. 42 Pengujian Fitur Pencarian Data Guru	101
Tabel 4. 43 Pengujian Fitur Download Data Guru	102
Tabel 4. 44 Pengujian Fitur Pembayaran Siswa	102
Tabel 4. 45 Pengujian Fitur Ubah Pembayaran Siswa.....	103
Tabel 4. 46 Pengujian Fitur Hapus Pembayaran Siswa	103
Tabel 4. 47 Pengujian Fitur Detail Pembayaran Siswa	104
Tabel 4. 48 Pengujian Fitur Pencarian Pembayaran Siswa	104
Tabel 4. 49 Pengujian Fitur Download Kwitansi Pembayaran Siswa	104
Tabel 4. 50 Pengujian Fitur Tambah Transaksi	105
Tabel 4. 51 Pengujian Fitur Ubah Transaksi.....	105
Tabel 4. 52 Pengujian Fitur Hapus Transaksi	106
Tabel 4. 53 Pengujian Fitur Pencarian Transaksi	106
Tabel 4. 54 Pengujian Fitur Tambah Pemasukan.....	107
Tabel 4. 55 Pengujian Fitur Ubah Pemasukan.....	107
Tabel 4. 56 Pengujian Fitur Hapus Pemasukan	107
Tabel 4. 57 Pengujian Fitur Pencarian Pemasukan.....	108
Tabel 4. 58 Pengujian Fitur Download Laporan Pemasukan	108

Tabel 4. 59 Pengujian Fitur Tambah Pengeluaran	108
Tabel 4. 60 Pengujian Fitur Ubah Pengeluaran	109
Tabel 4. 61 Pengujian Fitur Hapus Pengeluaran.....	109
Tabel 4. 62 Pengujian Fitur Pencarian Pengeluaran	110
Tabel 4. 63 Pengujian Fitur Download Laporan Pengeluaran.....	110
Tabel 4. 64 Pengujian Fitur Tambah Kas.....	110
Tabel 4. 65 Pengujian Fitur Ubah Kas.....	111
Tabel 4. 66 Pengujian Fitur Hapus Kas	111
Tabel 4. 67 Pengujian Fitur Pencarian Kas.....	111
Tabel 4. 68 Pengujian Fitur Download Kwitansi Pembayaran Kas	112
Tabel 4. 69 Pengujian Fitur Download Kwitansi Pengambilan Kas	112

DAFTAR GAMBAR

Gambar 1. 1	Data Pengguna Internet di Indonesia	1
Gambar 2. 1	Tahapan Metode Scrum	12
Gambar 3. 1	Struktur Organisasi SMA Plus Nurul Iman Leles.....	32
Gambar 3. 2	Teknik Pengambilan Data.....	35
Gambar 3. 3	Kerangka Berfikir	37
Gambar 3. 4	Metode Perancangan.....	40
Gambar 3. 5	Sistem yang Sedang Berjalan	41
Gambar 4. 1	Use Case Diagram	49
Gambar 4. 2	Activity Diagram Login.....	50
Gambar 4. 3	Activity Diagram Dashboard	51
Gambar 4. 4	Activity Diagram Siswa.....	51
Gambar 4. 5	Activity Diagram Guru	52
Gambar 4. 6	Activity Diagram Pencatatan	52
Gambar 4. 7	Activity Diagram Laporan.....	53
Gambar 4. 8	Activity Diagram Logout.....	53
Gambar 4. 9	Sequence Diagram Login	54
Gambar 4. 10	Sequence Diagram Dashboard.....	54
Gambar 4. 11	Sequence Diagram Siswa.....	55
Gambar 4. 12	Sequence Diagram Guru.....	55
Gambar 4. 13	Sequence Diagram Pencatatan.....	56
Gambar 4. 14	Sequence Diagram Laporan.....	56
Gambar 4. 15	Sequence Diagram Logout	57
Gambar 4. 16	Class Diagram.....	65
Gambar 4. 17	Desain Tampilan Halaman Login	68
Gambar 4. 18	Desain Tampilan Reset Password	68
Gambar 4. 19	Desain Tampilan Verifikasi Email	69
Gambar 4. 20	Desain Tampilan Dashboard	69
Gambar 4. 21	Desain Tampilan Menu siswa	70
Gambar 4. 22	Desain Tampilan Form Tambah Siswa	70
Gambar 4. 23	Desain Tampilan Menu Guru.....	71
Gambar 4. 24	Desain Tampilan Form Tambah Guru.....	71

Gambar 4. 25	Desain Tampilan Menu Pencatatan.....	72
Gambar 4. 26	Desain Tampilan Form Tambah Pemasukan.....	72
Gambar 4. 27	Desain Tampilan Menu Laporan.....	73
Gambar 4. 28	Desain Tampilan Menu Logout	73
Gambar 4. 29	Struktur Folder Front End.....	74
Gambar 4. 30	Kode Program Front End Login	75
Gambar 4. 31	Kode Program Front End Reset Password.....	75
Gambar 4. 32	Kode Program Front End Verifikasi Password	76
Gambar 4. 33	Kode Program Front End Dashboard	76
Gambar 4. 34	Kode Program Front End Siswa	77
Gambar 4. 35	Kode Program Front End Guru.....	77
Gambar 4. 36	Kode Program Front End Laporan	78
Gambar 4. 37	Kode Program Front End Pencatatan.....	78
Gambar 4. 38	Kode Program Back End Logout.....	79
Gambar 4. 39	Struktur Folder Back End	79
Gambar 4. 40	Kode Program Back End Repository.....	80
Gambar 4. 41	Kode Program Back End Service	80
Gambar 4. 42	Kode Program Back End Controller.....	81
Gambar 4. 43	Kode Program Back End Models	81
Gambar 4. 44	Kode Program Back End Middleware	82
Gambar 4. 45	Kode Program Back End Main.....	82
Gambar 4. 46	Kode Program Back End Initializers	83
Gambar 4. 47	Tampilan Halaman Login	83
Gambar 4. 48	Tampilan Halaman Reset Password.....	84
Gambar 4. 49	Tampilan Halaman Verifikasi	84
Gambar 4. 50	Tampilan Dashboard.....	85
Gambar 4. 51	Tampilan Halaman Siswa	85
Gambar 4. 52	Tampilan Form Tambah Siswa	86
Gambar 4. 53	Tampilan Halaman Guru.....	86
Gambar 4. 54	Tampilan Form Tambah Guru.....	87
Gambar 4. 55	Tampilan Halaman Pencatatan.....	87
Gambar 4. 56	Tampilan Form Tambah Pembayaran	88

Gambar 4. 57 Tampilan Laporan	88
Gambar 4. 58 Tampilan Halaman Logout	89

DAFTAR ISI

Lembar Pengesahan Skripsi	i
Lembar Pengujian Skripsi	ii
MOTTO.....	iii
PERNYATAAN KEASLIAN SKRIPSI	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR.....	vii
UCAPAN TERIMA KASIH	ix
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xiii
DAFTAR ISI	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	4
1.6. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	6
2.1. Rancang Bangun.....	6
2.2. Sistem	6
2.3. Informasi.....	8
2.4. Sistem Informasi.....	9
2.5. <i>Web</i>	10
2.6. <i>Black Box Testing</i>	10
2.7. <i>Metode Scrum</i>	11
2.8. <i>Golang</i>	13
2.9. <i>JavaScript</i>	14
2.10. <i>Next JS</i>	15
2.11. <i>Unified Modelling Language (UML)</i>	16
2.12. <i>Database</i>	21

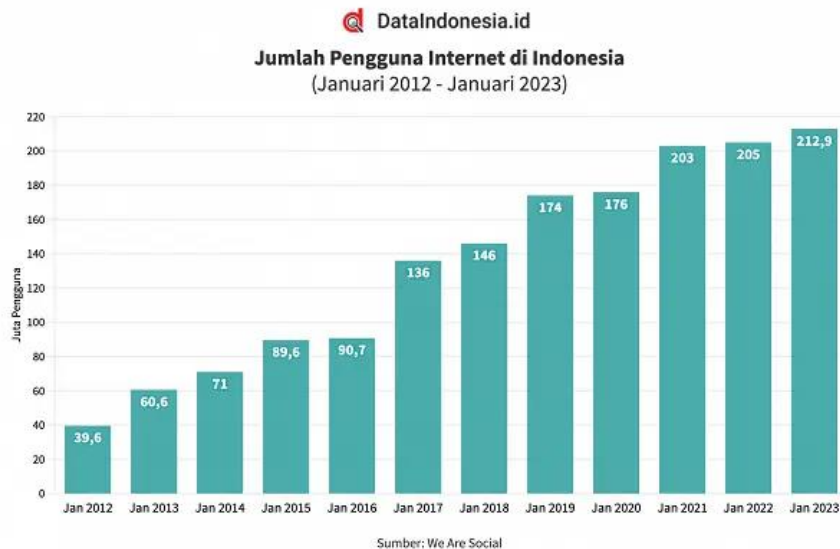
2.13.	<i>Data Base Management System</i>	22
2.14.	<i>PostgreSQL</i>	24
2.15.	<i>Application Programming Interface (API)</i>	25
2.16.	Penelitian Terdahulu	26
BAB III METODOLOGI PENELITIAN.....		31
3.1.	Objek Penelitian	31
3.1.1.	Profil Instansi	31
3.1.2.	Struktur Organisasi.....	31
3.2.	Metode dan Desain Penelitian	35
3.2.1.	Teknik Pengambilan Data	35
3.2.2.	Kerangka Berfikir.....	36
3.2.3.	Metode Perancangan	39
3.2.4.	Sistem yang Sedang Berjalan.....	41
BAB IV HASIL DAN PEMBAHASAN.....		42
4.1.	Analisis Sistem yang Sedang Berjalan.....	42
4.2.	Analisis Kelemahan Sistem yang Sedang Berjalan.....	42
4.3.	Identifikasi Kebutuhan Sistem	43
4.4.	Perencanaan Pembuatan Sistem	44
4.4.1.	Pembagian <i>Sprint</i>	45
4.4.2.	Kelengkapan <i>Sprint</i>	45
4.5.	Implementasi	48
4.5.1.	Desain Sistem.....	48
4.5.2.	Desain Kode Program	74
4.5.3.	Hasil Perancangan.....	83
4.6.	Pengujian Sistem	89
BAB V KESIMPULAN DAN SARAN.....		113
5.1.	Kesimpulan.....	113
5.2.	Saran.....	114
DAFTAR PUSTAKA		115
DAFTAR LAMPIRAN		120

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kemajuan teknologi yang pesat dan perubahan dinamis dalam era ini telah menciptakan transformasi besar-besaran, khususnya melalui perkembangan pada teknologi informasi. Perkembangan ini, memberikan banyak kemudahan dalam setiap aktivitas manusia. Hal ini dibuktikan dengan semakin meningkatnya penggunaan internet dari setiap tahunnya. Melansir dari DataIndonesia.id, jumlah pengguna internet saja mencapai 212,9 juta per Januari 2023 di Indonesia. Jumlah ini terhitung luar biasa, karena hampir 77% dari populasi Indonesia sudah menggunakan internet.



Gambar 1. 1 Jumlah Pengguna Internet di Indonesia
Sumber: dataindonesia.id

Penggunaan internet mencakup berbagai aspek, dan salah satunya adalah penggunaan *website* sebagai sarana akses informasi. *Website* merupakan salah satu bentuk utama dari konten *online* yang bisa diakses oleh pengguna internet. Melalui *website*, pengguna dapat mengakses berbagai informasi, mulai dari berita terkini, artikel ilmiah, tutorial, hingga informasi produk dan layanan. Selain itu, *website* juga dapat digunakan sebagai platform untuk berinteraksi, seperti forum diskusi atau komunitas *online*, yang memungkinkan pengguna untuk berbagi pengetahuan dan pengalaman.

Perkembangan internet seperti *website* memberikan pengaruh yang begitu signifikan di berbagai bidang, termasuk dalam lembaga pendidikan. Dalam ranah pendidikan, penerapan teknologi informasi mencakup sejumlah aspek, seperti pengelolaan keuangan, administrasi data kehadiran, dan manajemen gaji. Manajemen keuangan menjadi fokus utama, mengingat peran krusial keuangan dalam menjalankan operasional lembaga pendidikan secara efisien dan transparan. Penggunaan teknologi informasi dalam pengelolaan keuangan mencakup implementasi sistem otomatisasi, pelaporan keuangan *real-time*, dan integrasi data keuangan untuk memastikan akurasi dan keandalan informasi. Seluruhnya, integrasi teknologi informasi dalam lembaga pendidikan bertujuan meningkatkan efisiensi dan transparansi, menciptakan lingkungan belajar yang *modern* dan adaptif di era digital ini (Seta et al., 2021).

Namun, masih terdapat lembaga pendidikan yang menjalankan pengelolaan data keuangan secara manual seperti mengandalkan buku catatan tradisional atau menggunakan *microsoft excel*. Metode ini seringkali menimbulkan kesalahan penulisan data, yang disebabkan oleh kecenderungan *human error*. Hal ini terjadi karena proses penulisan dilakukan dengan kurangnya ketelitian saat penginputan data. Salah satu contoh lembaga pendidikan yang masih mengadopsi proses pengelolaan keuangan secara manual adalah SMA Plus Nurul Iman Leles.

SMA Plus Nurul Iman Leles, didirikan pada tahun 2011, telah menunjukkan pendekatan yang belum terkomputerisasi dalam pengelolaan administrasi keuangannya sejak awal berdiri. Proses pencatatan transaksi pembayaran dan penginputan data keuangan masih dilakukan secara manual, menyebabkan berbagai permasalahan, termasuk keterlambatan dalam pencarian data dan kerapnya kehilangan data. Kendala-kendala ini memberikan tantangan bagi manajemen dalam mengelola data keuangan, yang pada akhirnya memperlambat proses pengelolaan keuangan secara keseluruhan. Oleh karena itu, dibutuhkan pembangunan sebuah sistem yang bisa mempermudah pengelolaan administrasi, khususnya dalam penginputan data keuangan, dan pencarian data. Solusi yang diusulkan penulis yaitu pembuatan dan pembangunan sistem informasi administrasi keuangan berbasis *website*.

Berfokus pada permasalahan yang sudah dijelaskan, penulis tertarik untuk

menjalankan penelitian yang akan difokuskan pada perancangan dan pembangunan sistem informasi administrasi keuangan di SMA Plus Nurul Iman Leles. Penelitian ini akan diwujudkan dalam bentuk skripsi yang judul "Rancang Bangun Sistem Informasi Administrasi Keuangan SMA Plus Nurul Iman Leles Berbasis Web menggunakan Metode *Scrum*".

Dengan dibangunnya sistem informasi ini, semoga bisa membantu lembaga dalam proses pengelolaan keuangan yang lebih efektif, efisien dan modern

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka berikut merupakan rumusan permasalahannya:

1. Bagaimana merancang sistem informasi administrasi keuangan yang dapat membantu SMA Plus Nurul Iman Leles dalam proses mengelola keuangan yang lebih efektif dan efisien.
2. Bagaimana membangun sistem informasi sesuai dengan hasil perancangan.

1.3. Batasan Masalah

Dalam penelitian ini, penulis memberikan batasan supaya fokus tetap terjaga dan hasil yang diharapkan dapat dicapai secara optimal. Beberapa batasan yang diberlakukan meliputi:

1. Penelitian ini akan membatasi lingkup pada pembangunan sistem informasi administrasi keuangan mencakup pengelolaan keuangan, data siswa dan data guru di SMA Plus Nurul Iman Leles. Aspek lain di luar itu tidak akan dibahas secara mendalam.
2. Perancangan dan pembangunan sistem merujuk pada hasil penelitian yang telah disebutkan sebelumnya.
3. Sistem informasi ini hanya memiliki satu *role* yaitu admin (bendahara) yang mempunyai hak akses penuh terhadap sistem.
4. Tidak ada pembayaran atau transaksi elektronik.

1.4. Tujuan Penelitian

Adapun tujuan dari penulisan penelitian skripsi ini adalah sebagai berikut:

1. Membuat perancangan sistem informasi administrasi keuangan yang dapat membantu SMA Plus Nurul Iman Leles dalam mengelola keuangan yang

lebih efektif dan efisien.

2. Membangun sistem informasi sesuai dengan hasil perancangan.

1.5. Manfaat Penelitian

Adapun manfaat yang dapat diambil dari hasil penelitian ini adalah:

1. Bagi Penulis

Penelitian ini memberikan kontribusi signifikan terhadap pengembangan kompetensi penulis di bidang teknologi informasi. Selain itu, penelitian ini juga memberikan kontribusi akademis dengan menyumbangkan pengetahuan dan wawasan baru dalam pengembangan sistem informasi berbasis web serta penerapan metode *Scrum*, khususnya dalam konteks administrasi keuangan lembaga pendidikan.

2. Bagi Pengguna

Penelitian ini memberikan manfaat dalam meningkatkan efisiensi operasional dengan mempercepat dan menyederhanakan proses administrasi keuangan, menghasilkan penghematan waktu dan sumber daya. Penekanan pada akurasi informasi keuangan diharapkan memberikan dasar yang kuat untuk pengambilan keputusan strategis, menjamin kepastian dan keandalan dalam manajemen keuangan.

3. Bagi Lembaga

Dengan penerapan teknologi *modern* dalam administrasi keuangan, lembaga diharapkan dapat meraih peningkatan reputasi sebagai institusi yang progresif dan inovatif. Selain itu, penggunaan sistem informasi yang baik dapat memberikan keuntungan berupa peningkatan transparansi dalam pengelolaan keuangan, yang pada gilirannya memperkuat kepercayaan dari pihak-pihak yang terkait. Selanjutnya, hal ini juga menciptakan kesempatan bagi lembaga untuk menyesuaikan diri dengan era digital, meningkatkan kesiapan dalam menghadapi tantangan zaman dengan memanfaatkan teknologi informasi untuk meningkatkan efektivitas operasional secara keseluruhan.

1.6. Sistematika Penulisan

Skripsi yang berjudul “Rancang Bangun Sistem Informasi Administrasi Keuangan SMA Plus Nurul Iman Leles Berbasis Web Menggunakan Metode

Scrum” ini tersusun secara sistematis dalam beberapa bab dan setiap bab, saling berkaitan dan menjelaskan tentang sistem yang dibuat.

BAB I PENDAHULUAN

Bab ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini terdapat teori-teori yang dijadikan sebagai acuan dalam melakukan penelitian, dan penelitian terdahulu yang pastinya berkaitan dengan penelitian yang dilakukan.

BAB III METODOLOGI PENELITIAN

Bab ini berisi uraian tentang objek penelitian, metode dan desain penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi uraian terkait dengan hasil rancangan dan sistem yang dibangun sesuai dengan metodologi yang digunakan.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan penulisan skripsi, sekaligus sikap akhir penulis terkait permasalahan yang ada. Selain kesimpulan, penulis juga menyampaikan saran dan hasil pemikiran berupa rekomendasi yang diharapkan bisa bermanfaat.

BAB II

TINJAUAN PUSTAKA

2.1. Rancang Bangun

Perancangan adalah tahap penting dalam pembangunan program komputer. Tujuan adanya perancangan ini tentunya memberikan gambaran informasi yang jelas dan komprehensif kepada para pemrogram dan teknisi yang terlibat dalam pengembangan. Perancangan haruslah memberikan manfaat dan pemahaman sehingga dapat diimplementasikan dengan lancar. Ini melibatkan berbagai prosedur yang dipakai untuk mengubah hasil analisis dari sistem ke dalam kode pemrograman, dengan maksud menjelaskan secara rinci cara komponen sistem tersebut akan diimplementasikan (Novitasari et al., 2021).

Rancang bangun adalah kumpulan prosedur yang bertujuan untuk mengubah hasil analisis sistem menjadi pemrograman. Proses ini dilakukan untuk menggambarkan secara rinci bagaimana setiap komponen dalam sistem akan diimplementasikan (Studi Sistem Informasi Universitas Trilogi Jakarta Jl TMP Kalibata No et al., 2021).

Rancang bangun merupakan proses di mana suatu sistem dibangun dari awal atau dimodifikasi untuk memperbaiki bagian-bagian tertentu. Prosedur ini dapat melibatkan penciptaan sistem yang baru atau *maintenance* pada sistem yang lama, baik secara menyeluruh maupun hanya pada bagian-bagian tertentu (Putra et al., 2020).

Berdasarkan penjelasan ketiga teori di atas bisa disimpulkan bahwa perancangan adalah tahapan kunci dalam pengembangan program komputer yang bertujuan untuk menyampaikan gambaran informasi yang jelas dan komprehensif kepada para pemrogram dan teknisi. Pentingnya perancangan terletak pada kebermanfaatan dan kemudahan pemahaman, sehingga implementasi dapat berjalan dengan lancar. Proses ini melibatkan serangkaian prosedur yang memiliki tujuan mengubah hasil analisis sistem menjadi pemrograman, dengan fokus pada penjelasan detail mengenai implementasi komponen-komponen sistem.

2.2. Sistem

Sistem merupakan kesatuan yang berdasarkan elemen-elemen, bagian-bagian, atau faktor-faktor, yang telah diatur sedemikian rupa sehingga mampu berinteraksi

satu sama lain dan saling bergantung. Sistem memiliki beberapa karakteristik sebagai berikut (Shalludin & Maulida, 2016).

1. Komponen Sistem

Sebuah sistem terdiri dari berbagai elemen yang saling terhubung untuk membentuk suatu kesatuan. Elemen-elemen ini dapat berupa subsistem dengan karakteristik dan fungsi khusus yang mempengaruhi kinerja keseluruhan sistem.

2. Batasan Sistem

Sistem memiliki lingkup yang membedakannya dari entitas lain, dan keterbatasan ini membuatnya terlihat sebagai suatu entitas yang tak terpisahkan.

3. Lingkungan Luar Sistem

Lingkungan eksternal suatu sistem merujuk pada operasi sistem yang dipengaruhi oleh segala aspek di luar batasan sistem itu sendiri. Lingkungan ini dapat memberikan manfaat maupun kerugian bagi sistem. Lingkungan yang memberikan manfaat perlu dipertahankan dan dijaga karena merupakan sumber energi bagi sistem. Di sisi lain, lingkungan yang dapat menimbulkan kerugian perlu dikendalikan agar tidak mengganggu kinerja sistem.

4. Penghubung Sistem

Penghubung sistem adalah sarana yang menghubungkan sistem utama dengan subsistem lainnya, memfasilitasi aliran sumber daya dari satu subsistem ke yang lain.

5. Masukan Sistem

Sistem menerima energi melalui dua jenis input: input pemeliharaan, seperti maintenance, dan input sinyal. Sebagai contoh, dalam konteks sistem komputer, program digunakan sebagai input pemeliharaan untuk menjalankan operasi komputer, sementara data berfungsi sebagai input sinyal yang diubah menjadi informasi.

6. Keluaran Sistem

Energi yang telah diatur dan digabungkan menjadi *output* sistem yang memiliki fungsi tertentu, yang kemudian akan menjadi masukan bagi

subsistem lainnya. Sebagai contoh, dalam sistem informasi, *output* yang dihasilkan adalah informasi.

7. Pengolah Sistem

Sebelum dihasilkan sebagai *output*, *input* akan diproses oleh sistem. Sebagai contoh, dalam sistem akuntansi, data transaksi akan diolah menjadi laporan.

8. Sasaran Sistem

Setiap sistem memiliki tujuan tertentu. Tanpa tujuan yang jelas, pengoperasian sistem menjadi tidak efektif karena keberhasilan sistem diukur dari sejauh mana sistem tersebut bisa tepat sasaran atau tujuan yang ditetapkan.

Sistem adalah gabungan elemen yang saling terhubung, dirancang untuk memungkinkan aliran energi, materi, atau informasi guna mencapai tujuan tertentu. Istilah ini sering dipakai untuk menggambarkan interaksi berbagai entitas (Solahudin, 2021).

Sistem adalah kumpulan komponen, elemen, atau variabel yang telah diatur dengan baik dan dapat berinteraksi satu sama lain secara terstruktur dan saling memengaruhi, yang kesemuanya terintegrasi menjadi satu kesatuan (Faqih & Wahyudi, 2022).

Dari ketiga definisi di atas, bisa diambil kesimpulan bahwa sistem adalah suatu kesatuan berbagai elemen yang terkoneksi satu dengan yang lainnya, untuk memfasilitasi aliran, energi, informasi, atau materi supaya mencapai tujuan tertentu. Istilah ini sering dipakai untuk menggambarkan interaksi dari entitas-entitas tertentu.

2.3. Informasi

Informasi merupakan hasil olahan atau pengelompokan data, digunakan untuk membantu proses pengambilan keputusan. Nilai dari informasi ditentukan oleh manfaat yang diberikannya dibandingkan dengan biaya yang diperlukan untuk memperolehnya. Informasi dikatakan bernilai jika manfaat yang dihasilkan lebih terasa daripada banyaknya biaya yang dikeluarkan (Wijaya & Hendrastuty, 2022).

Informasi adalah hasil pengolahan data yang dibentuk untuk memberikan nilai dan makna yang lebih bermanfaat kepada penerimanya. Data, sebagai sumber

informasi, merupakan kumpulan fakta yang memberikan gambaran tentang peristiwa dan kesatuan nyata. Kejadian ini merupakan peristiwa yang terjadi pada waktu tertentu (Shalludin & Maulida, 2016)

Informasi merupakan hasil pengolahan atau pengelompokan data agar dapat digunakan dalam proses membuat keputusan. Penilaian terhadap informasi tergantung pada banyaknya manfaat dan biaya yang terkait dengannya; informasi dianggap berharga apabila manfaat yang diberikannya lebih besar ketimbang biaya yang dikeluarkan untuk memperolehnya (Faqih & Wahyudi, 2022).

Berdasarkan pengertian ketiga teori di atas, dapat ditarik kesimpulan bahwa informasi merupakan hasil pengolahan data yang telah diproses sehingga menjadi lebih bermanfaat dan memiliki makna yang lebih jelas. Data, sebagai sumber informasi, merupakan kumpulan fakta yang memberikan gambaran peristiwa dan kesatuan secara nyata terjadi pada waktu tertentu. Informasi memiliki nilai yang ditentukan oleh manfaat yang diberikannya dibandingkan dengan biaya yang diperlukan untuk memperolehnya. Informasi diklaim bernilai jika manfaatnya lebih besar daripada biaya yang dikeluarkan.

2.4. Sistem Informasi

Sistem informasi merupakan sistem dalam sebuah organisasi yang dirancang untuk mendukung operasional, manajemen, memenuhi kebutuhan pengolahan, dan aktivitas strategis organisasi. Sistem ini bertujuan untuk menyediakan informasi yang bermanfaat bagi tingkatan manajemen di semua tingkat dalam organisasi, serta dapat memberikan laporan- yang dibutuhkan bagi pihak eksternal (Novitasari et al., 2021).

Sistem informasi merupakan bagian dari struktur organisasi yang menghubungkan kebutuhan untuk memproses transaksi, dengan fungsi organisasi yang bersifat manajerial, serta aktivitas strategis organisasi. (Wijaya & Hendrastuty, 2022)

Sistem bisa didefinisikan melalui pendekatan prosedural atau komponen. Dalam pendekatan prosedural, sistem dapat dijelaskan sebagai serangkaian prosedur yang memiliki tujuan tertentu (Oktaviani et al., 2020).

Dari ketiga definisi tersebut, dapat ditarik kesimpulan bahwa sistem informasi dalam sebuah organisasi menghubungkan kebutuhan pengolahan dengan operasi

manajerial dan aktivitas strategis. Tujuan utamanya adalah menyediakan informasi yang bermanfaat bagi manajemen di berbagai tingkatan serta menghasilkan laporan yang dibutuhkan oleh pihak eksternal.

2.5. Web

Sebuah situs web adalah kumpulan halaman online yang terhubung dengan topik tertentu, seringkali dilengkapi berbagai berkas multimedia seperti video, gambar dan lain-lain. Situs web ini diterbitkan di internet dan dapat diakses melalui domain atau sebuah *Uniform Resource Locator* yang bisa diakses oleh pengguna internet dengan memasukkan alamatnya di *browser web* (Novitasari et al., 2021).

Sebuah situs *web* adalah kumpulan halaman yang digunakan untuk menampilkan beragam informasi, seperti teks, gambar diam atau bergerak, animasi, atau kombinasi dari keduanya. Halaman-halaman ini membentuk struktur yang saling terhubung dan terkoneksi ke jaringan halaman lainnya (Wijaya & Hendrastuty, 2022).

Web atau situs adalah kumpulan halaman yang menampilkan berbagai media informasi. Informasi ini bisa berupa tetap atau berubah-ubah, membentuk struktur yang terhubung di mana setiap halaman terhubung melalui tautan di internet (Oktaviani et al., 2020)

Dari definisi tersebut, dapat diambil kesimpulan bahwa sebuah situs web adalah gabungan halaman yang digunakan untuk menampilkan beragam informasi. Ini mengatur struktur yang saling terkait dan terhubung dengan jaringan halaman lainnya. Situs web ini diterbitkan di internet dan dapat diakses melalui domain atau *Uniform Resource Locator* yang bisa diakses oleh pengguna internet dengan memasukkan alamatnya di browser web.

2.6. Black Box Testing

Metode *blackbox testing* adalah sebuah pendekatan percobaan yang terpusat pada fungsional *software*. Pada metode ini, pengujian dilakukan dengan mendefinisikan serangkaian kondisi masukan dan mengujinya berdasarkan spesifikasi fungsional yang telah ditentukan. Proses berbasis *black box testing* dilakukan dengan mengoperasikan program yang telah dikembangkan mengisi semua formulir sesuai dengan spesifikasi fungsional yang telah ditetapkan (Shadiq et al., 2021).

Metode uji coba *black box* berpusat pada kebutuhan fungsional *software*. Dalam metode ini, pengembang dapat membuat serangkaian kondisi masukan yang mencakup semua persyaratan fungsional dari program tersebut. Uji coba *black box* bukanlah pengganti uji coba *white box*, melainkan pendekatan tambahan untuk menemukan kesalahan dengan cara yang berbeda, selain dari metode *white box* (Utomo et al., 2020).

Black Box Testing adalah jenis pengujian di mana perangkat lunak diuji tanpa memperhatikan struktur internalnya, hanya berdasarkan hasil dari data uji yang dimasukkan. (Hendri et al., 2020).

Kesimpulannya bahwa *black box testing* merupakan jenis pengujian yang fokus pada spesifikasi fungsional *software*. Dalam pendekatan ini, pengujian dilakukan dengan menetapkan masukan dan menguji program berdasarkan spesifikasi fungsionalnya. Pendekatan ini memungkinkan pengembang untuk membuat masukan yang mencakup semua syarat fungsional program. Pengujian ini tidak menggantikan *white box testing*, melainkan menjadi pendekatan tambahan untuk menemukan kesalahan dengan cara yang berbeda, tanpa memeriksa struktur internal atau implementasi *software*.

2.7. Metode Scrum

Scrum merupakan salah satu pendekatan rekayasa *software* yang mengadopsi prinsip *agile*. Ini menyoroti pentingnya kerja sama tim, pengembangan produk secara bertahap, dan proses berulang untuk mencapai tujuan akhir. *Scrum* kerangka kerja yang dapat mengatasi permasalahan kompleks yang selalu berubah, serta dianggap mampu menghasilkan produk berkualitas tinggi yang sesuai dengan kebutuhan pengguna secara inovatif dan efisien (Aryanata Andipradana & Kristoko Dwi Hartomo, 2021).

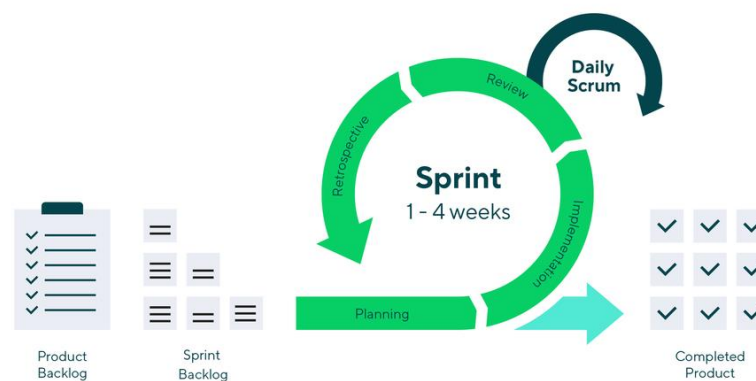
Scrum adalah proses pengembangan produk atau layanan yang inovatif, yang berjalan secara berulang. *Scrum* adalah kerangka kerja metodologi yang memberikan fleksibilitas dalam mengontrol dan mengelola persyaratan serta pengembangan perangkat lunak (Muhamad Rizky & Yuni Sugiarti, 2021).

Scrum adalah sebuah kerangka kerja yang dirancang untuk menangani pekerjaan kompleks yang terus berubah. Kerangka kerja ini digunakan untuk

mengatasi masalah-masalah adaptif yang kompleks, serta mendorong kreativitas dan inovasi (Wahyu Adi Prabowo & Citra Wiguna, 2021).

Scrum adalah kerangka kerja responsif yang digunakan dalam pembuatan *software* untuk mengurus produk atau aplikasi. Kerangka kerja ini berpusat pada strategi dan fleksibilitas dalam bekerja sebagai satu tim guna bersama-sama mencapai tujuan (Sri Puji Utami et al., 2022)

Secara umum, metode *scrum* merupakan pendekatan yang populer dalam pengembangan perangkat lunak karena menekankan kecepatan, fleksibilitas, kolaborasi tim, dan hasil akhir yang terukur.



Gambar 2. 1 Tahapan Metode *Scrum*

Berikut merupakan tahapan dalam pengembangan aplikasi menggunakan metode *Scrum*.

1. *Product Backlog*

Product log adalah list kebutuhan yang perlu diketahui dan dipahami saat ini terkait dengan produk yang sedang dikembangkan. Ini mencakup informasi tentang fitur, fungsionalitas, dan kebutuhan lain yang harus dipenuhi oleh produk tersebut. Product log berfungsi sebagai panduan bagi tim pengembangan dalam mengidentifikasi dan memprioritaskan pekerjaan yang perlu dilakukan untuk memenuhi kebutuhan produk (Wahyu Adi Prabowo & Citra Wiguna, 2021)

2. *Sprint Planning*

Sprint planning adalah tahap dimana membuat perencanaan yang akan dikerjakan sesuai prioritas dari *product backlog* (Setiadana, 2021).

3. *Sprint*

Sprint adalah tahap implementasi dalam metodologi pengembangan perangkat lunak *scrum*. Dalam *sprint*, mulai bekerja untuk menghasilkan sebuah *increment* yang merupakan potongan fungsionalitas dari produk yang siap untuk diserahkan kepada pengguna akhir. Setiap *Sprint* memiliki waktu yang tetap, biasanya dalam rentang waktu satu sampai empat minggu, yang di mana selama periode tersebut tim berfokus pada penyelesaian pekerjaan yang telah dipilih dari *product backlog* (Raditya Aryaputra & Kristoko Dwi Hartomo, 2023).

4. *Sprint Review*

Sprint Review merupakan pertemuan yang dilakukan untuk memperlihatkan kepada pemangku kepentingan apa yang telah dicapai selama *sprint* dan mendapatkan *feed back* dari mereka. Tujuan nya adalah untuk mengevaluasi pekerjaan yang telah diselesaikan dan membuat perubahan yang diperlukan pada *Product Backlog* (Sri Puji Utami et al., 2022).

2.8. *Golang*

Golang bahasa pemrograman yang dibuat Ken Thompson, Robert Griesemer, dan Rob Pike dari *Google* pada tahun 2009, terkenal karena kecepatannya, keandalannya, skalabilitas, dan kesederhanaannya (Priadi & Idris, n.d.).

Golang terdapat penggunaan konkurensi dalam pemrograman sistem dengan implementasi yang mudah. Selain itu, *Golang* menggunakan proses *built-in garbage collector (Goroutines)* yang efektif untuk mengelola sampah memori, memastikan sistemnya memiliki penanganan yang baik terhadap *garbage collection*. Dengan keandalan dan kecepatannya dalam skala besar, *golang* termasuk bahasa pemrograman yang ringan (Kristanto, A, 2020)

Golang adalah bahasa pemrograman yang dirancang untuk memberikan lingkungan pengembangan perangkat lunak modern yang efisien, sederhana, dan mudah digunakan (Nawaf Naofal et al., 2023)

Golang (Go Language) adalah bahasa pemrograman *opensource* yang dikembangkan Rob Pike, Robert Griesemer, Ken Thompson, dan banyak kontributor lainnya di komunitas pengembang yang bersifat terbuka. Bahasa

pemrograman ini dirancang untuk mempermudah pengembangan perangkat lunak yang efisien dan skalabel, terutama dalam lingkungan yang membutuhkan kinerja tinggi (Pamungkas & Setiaji, n.d.).

Bahasa pemrograman ini adalah bahasa yang open-source yang memiliki gaya sintaksis mirip dengan bahasa C dan C++. Bahasa ini dikembangkan oleh *Google* dan memiliki beberapa keunggulan, seperti sintak yang sederhana sehingga mudah dipelajari, manajemen memori yang ringan dan baik secara sintaksis, serta proses kompilasi yang cepat (Ni Kadek Dwi Sabrina et al., 2023).

Berdasarkan ketiga teori di atas dapat ditarik kesimpulan bahwa, *Go*, atau *Golang*, merupakan pemrograman yang dibuat pada tahun 2009 oleh *google*. Bahasa ini terkenal karena kecepatan, kehandalan, skalabilitas, dan kesederhanaannya. *Golang* menggunakan proses *built-in garbage collector* (*Goroutines*) yang efektif untuk mengelola sampah memori, memastikan sistemnya memiliki penanganan yang baik terhadap garbage collection. Dengan keandalan dan kecepatannya dalam skala besar, *Go* termasuk bahasa pemrograman yang bersifat *clean code* agar tidak membebani sistem.

2.9. JavaScript

JavaScript merupakan pemrograman yang dapat mengimplementasi fitur-fitur yang canggih pada sebuah *website*. Selain itu, *JavaScript* juga merupakan bagian penting dari teknologi *web* standar (Salim et al., 2022).

JavaScript lebih dinamis dibandingkan dengan pemrograman lain. Peran *JavaScript* dalam pengembangan *website* adalah menyediakan interaktivitas dan pengalaman pengguna yang lebih responsif melalui efek animasi dan penanganan *event* yang dinamis (Putra et al., 2020).

Javascript adalah bahasa pemrograman yang pada awalnya dikembangkan untuk beroperasi di dalam *browser*. Namun, seiring dengan perkembangan, *Javascript* tidak terbatas hanya pada lingkungan *browser*, tetapi juga dapat digunakan di berbagai platform lainnya (Wahyudi, 2021).

JavaScript adalah Bahasa pemrograman yang banyak digunakan untuk pengembangan web adalah bahasa pemrograman yang berperan sebagai *Client Side Programming Language*. Artinya, pemrosesannya terjadi di sisi client, yaitu di web browser. *JavaScript* digunakan untuk menciptakan halaman web yang aktif dan

bergerak, memungkinkan pengguna berinteraksi dengan konten secara langsung, yang berarti dapat merespons terhadap tindakan pengguna tanpa perlu memuat ulang halaman (Ananda & Nama, 2024).

JavaScript adalah pemrograman yang sangat populer untuk pembuatan web aplikasi saat ini. Pemrograman ini pertama kali dibuat sekitar tahun 1995 dan sudah dipakai oleh banyak pengembang untuk membuat berbagai jenis aplikasi *webiste*. Salah satu alasan utama kepopuleran *JavaScript* adalah karena kompatibilitasnya yang luas dengan hampir semua *browser*, memungkinkan penggunaannya di berbagai platform. Selain itu, *JavaScript* juga dianggap relatif mudah dipelajari, terutama bagi pemula, karena lebih sederhana daripada bahasa pemrograman lain seperti *Java*, *C#*, dan *C++* (Yefta Christian & Hengky, 2023).

Berdasarkan teori tersebut disimpulkan bahwa *JavaScript* adalah bahasa pemrograman yang awalnya dikembangkan untuk menyokong fitur-fitur kompleks pada website, terutama di lingkungan *browser*. Namun, seiring perkembangannya, *JavaScript* telah meluas ke berbagai platform lainnya di luar browser. Selain itu, *JavaScript* juga merupakan bagian penting dari teknologi web standar.

2.10. Next JS

Next.js merupakan kerangka kerja yang fleksibel yang memungkinkan pembuatan aplikasi secara cepat. Untuk menciptakan tampilan pengguna (*Admin Interface*) yang dinamis, *Next.js* membutuhkan *ReactJs* sebagai pustaka *JavaScript* (Baehaqi et al., 2023).

Next.js merupakan kerangka kerja yang terkenal dalam pengembangan *web* aplikasi karena dapat menyederhanakan proses *render disisi* server dan pembuatan situs tetap, serta meningkatkan fungsionalitas sistem (Susanto et al., 2023).

Next.js adalah sebuah kerangka kerja *ReactJS* yang digunakan untuk mengembangkan aplikasi web di sisi klien. Saat ini, *Next.js* telah digunakan oleh banyak *website* di seluruh dunia. Beberapa perusahaan ternama seperti *Nike*, *Netflix*, dan *Playstation* telah mengadopsi *Next.js* untuk penggunaan mereka. (Ananda & Nama, 2024).

Next.js merupakan sebuah kerangka kerja *JavaScript* yang melengkapi fitur-fitur dari *React*. Fokus utamanya adalah pada pengembangan aplikasi web yang responsif dan kuat dengan menyediakan berbagai fitur yang dapat meningkatkan

pengalaman pengguna serta mempermudah pengembangan aplikasi web secara efisien. Salah satu keunggulan *Next.js* adalah kemampuannya dalam menangani *Rendering Sisi Server* (SSR) dan Pembuatan Situs Statis (SSG) (Hajjul Ikram & Muhammad Ardiansyah, 2024).

Next.js adalah sebuah framework *JavaScript* yang dapat dipakai untuk mengembangkan *web* aplikasi *fullstack* secara cepat. Dalam pengembangannya, beberapa aspek penting sangat dipertimbangkan. Salah satunya adalah proses rendering *website* dilakukan di server untuk memungkinkan halaman web ditampilkan lebih cepat di browser (Ilham Taufiq et al., 2023)

Berdasarkan pengertian kedua teori di atas kesimpulannya adalah *Next.js* sebuah kerangka kerja yang populer dan fleksibel dalam pengembangan *web* aplikasi. Dengan kemampuannya untuk menyederhanakan proses pembuatan antarmuka pengguna yang interaktif dan cepat, serta dukungan terhadap *React* sebagai library *JavaScript*, *Next.js* menjadi pilihan yang kuat dalam mengembangkan aplikasi web yang dapat diandalkan.

2.11. Unified Modelling Language (UML)

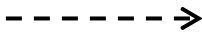
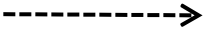
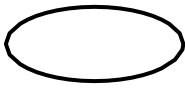
Unified Modeling Language merupakan standar industri untuk visualisasi, perancangan, dan dokumentasi perangkat lunak. UML memberikan standar untuk merancang model sistem. Terdapat berbagai diagram di dalam UML, yaitu (Hartono, 2020).

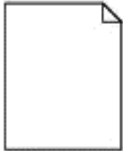
1. Use Case Diagram

Use case diagram digunakan untuk mengilustrasikan fungsi yang diinginkan dari suatu sistem, dengan fokus pada apa yang sistem lakukan daripada bagaimana melakukan hal tersebut. Setiap *use case* mewakili interaksi antara aktor (pengguna atau sistem eksternal) dan sistem tersebut.

Tabel 2. 1 Use Case Diagram

Simbol	Keterangan
	Seseorang, entitas, atau sistem lain yang berinteraksi dengan sistem informasi yang dibuat sendiri, meskipun diwakili oleh simbol

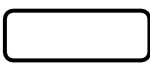
Simbol	Keterangan
<i>Actor</i>	aktor berupa gambar orang, umumnya diidentifikasi menggunakan kata benda di awal nama aktornya.
 <i>dependency</i>	Hubungan di mana modifikasi pada suatu elemen yang tergantung padanya tidak mempengaruhi elemen yang bersifat mandiri (<i>independent</i>).
 <i>generalization</i>	Hubungan di mana objek turunan (<i>descendant</i>) mengadopsi perilaku dan struktur data dari objek yang lebih tinggi, yaitu objek leluhur (<i>ancestor</i>).
 <i>include</i>	Menegaskan bahwa penggunaan sumber-sumber eksplisit dalam kasus penggunaan.
 <i>extend</i>	Mengarahkan use case target untuk memperluas fungsi yang sama dengan use case sumber pada titik tertentu yang ditentukan.
 <i>Association</i>	Menspesifikasikan paket yang menampilkan sistem dengan batasan tertentu.
 <i>Use Case</i>	Deskripsi dari urutan langkah-langkah yang diterapkan oleh sistem untuk mencapai hasil yang dapat diukur bagi seorang pelaku.
	Interaksi antara berbagai aturan dan elemen bekerja bersama untuk

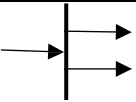
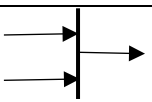
Simbol	Keterangan
<i>Collaborations</i>	menghasilkan perilaku yang kompleks melebihi dari sekadar jumlah aturan dan elemen yang terlibat
 <i>Note</i>	Elemen fisik saat aplikasi berjalan.

2. Activity Diagram

Activity diagram mengilustrasikan aliran berbagai aktivitas dalam sebuah sistem yang direncanakan. Mereka memperlihatkan bagaimana setiap aktivitas diawali, keputusan yang mungkin akan diambil, dan bagaimana aktivitas berakhir.

Tabel 2. 2 Activity Diagram

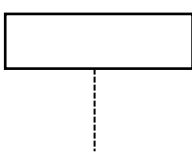
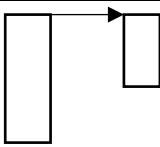
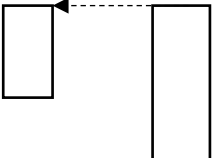
Simbol	Keterangan
 <i>Start Point</i>	Membuat diagram aktivitas yang dimulai dengan menunjukkan alur kerja atau sistem dengan simbol lingkaran solid.
 <i>End Point</i>	Menggambarkan akhir dari sebuah presentasi dengan menunjukkan alur kerja sistem dalam diagram aktivitas, yang ditandai dengan lingkaran di sekelilingnya.
 <i>Activity</i>	mempresentasikan kinerja dari beberapa perilaku dalam suatu alur kerja

Simbol	Keterangan
 <i>Fork (Percabangan)</i>	menunjukkan aktivitas yang dilakukan secara bersamaan.
 <i>Join (Penggabungan)</i>	Mengintegrasikan dua kegiatan yang berjalan bersama menjadi satu.
 <i>Decision Point</i>	Menentukan saat di mana suatu kegiatan membagi alirannya menjadi beberapa cabang

3. *Sequence Diagram*

Sequence diagram digunakan untuk mengilustrasikan ubungan antara komponen dalam sistem, seperti pengguna, tampilan, dan komponen lainnya, dengan menggunakan urutan pesan yang ditampilkan sepanjang waktu. Ini adalah diagram dengan sumbu waktu vertikal dan sumbu horizontal yang menunjukkan objek-objek yang terlibat.

Tabel 2. 3 *Sequence Diagram*

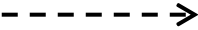
Simbol	Keterangan
 <i>Lifeline</i>	Objek masuk, antarmuka yang berinteraksi satu sama lain
 <i>Message Entry</i>	Spesifikasi komunikasi antar objek yang mencakup pengiriman pesan secara asinkron mengenai kejadian atau aktivitas yang terjadi.
	Spesifikasi untuk komunikasi antar objek yang melibatkan pemanggilan metode dengan fokus

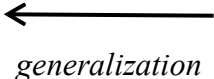
Simbol	Keterangan
<i>Message to Self</i>	pada hasil nilai kembalian yang direkam dari aktivitas yang berlangsung

4. *Class Diagram*

Class adalah sebuah diagram yang, saat diinstansiasi, menciptakan sebuah objek. Ini merupakan elemen kunci dalam pengembangan dan desain berorientasi objek. *Class* menggambarkan karakteristik (atribut/properti) dari sebuah sistem dan menyediakan metode untuk mengelola karakteristik tersebut (metode/fungsi).

Tabel 2. 4 *Class Diagram*

Simbol	Keterangan
<i>Class</i>	Simbol ini mewakili sebuah kelas dalam struktur sistem penulisan di mana penggunaan spasi tidak diperbolehkan. Simbol tersebut terdiri dari tiga bagian: bagian pertama adalah nama kelas, bagian tengah menentukan properti atau atribut dari kelas tersebut, dan bagian terakhir menentukan metode yang dimiliki setiap kelas.
 <i>dependency</i>	Kadang-kadang, sebuah kelas memanfaatkan kelas lain. Biasanya, penggunaan ketergantungan menunjukkan bahwa suatu kelas melakukan operasi dengan menggunakan kelas lainnya. Ketergantungan ini ditunjukkan

Simbol	Keterangan
	dengan sebuah panah dengan titik-titik di antaranya..
 <i>generalization</i>	Jika suatu kelas tidak dapat beroperasi secara independen dan harus menjadi bagian dari kelas lain, ini menunjukkan bahwa kelas tersebut memiliki hubungan komposisi dengan kelas yang lebih tinggi. Dalam konteks ini, hubungan komposisi digambarkan sebagai garis solid dengan ujung berbentuk panah kecil.
 <i>Association</i>	Hubungan antara kelas dalam konteks umum sering kali disertai dengan asosiasi yang juga mencakup <i>multiplicity</i> .

2.12. Database

Database merupakan kumpulan data terintegrasi dan terorganisir yang disimpan dalam suatu wadah, sehingga memungkinkan informasi untuk dicari, diakses, dan dikelola dengan cepat (Priadi & Idris, n.d.).

Database adalah sebuah kumpulan data, terkait secara logis dan disimpan bersama, dibuat untuk memberi kebutuhan informasi suatu organisasi (Raharjo, 2021).

Basis data merupakan kumpulan data yang terkait yang disusun secara terstruktur untuk memudahkan penggunaannya dalam memenuhi berbagai kebutuhan. Data-data ini disimpan dalam format elektronik sehingga dapat diakses dan dimanfaatkan dengan cepat (Putra et al., 2020).

Berdasarkan teori tersebut disimpulkan bahwa, *database* merupakan kumpulan data yang terhubung dan terorganisir secara logis yang disimpan dalam

suatu wadah. Hal ini memungkinkan informasi untuk dicari, diakses, dan dikelola dengan efisien sesuai dengan kebutuhan informasi suatu organisasi.

2.13. Data Base Management System

Sistem Manajemen *Database* merupakan sekumpulan data yang terhubung dan berbagai program yang digunakan untuk memproses data tersebut. Adapun fungsi dan pelayanan DBMS sebagai berikut (Raharjo, 2021).

1. *Data Storage Management*

DBMS membentuk kerangka kerja yang penting untuk menyimpan data dalam struktur fisik dari sebuah basis data. Ini memberikan cara untuk mengatur penyimpanan data secara permanen.

2. Manajemen transaksi

Transaksi adalah rangkaian tindakan yang dilakukan oleh program aplikasi pada sebuah database untuk mengakses atau mengubah konten *database*. Oleh karena itu, sistem manajemen basis data (DBMS) harus menjamin bahwa semua perubahan yang terkait dengan suatu transaksi entah dilaksanakan sepenuhnya atau tidak ada yang dilakukan sama sekali.

3. Layanan integritas

Integritas database merujuk pada keakuratan dan kesesuaian data yang tersimpan, yang menjadi krusial dalam sistem *database* yang fokus pada transaksi. Oleh karena itu, DBMS harus menawarkan cara untuk memastikan bahwa data yang ada dalam database serta modifikasi data mengikuti kebijakan yang telah ditetapkan.

4. Manajemen pencadangan dan pemulihan

DBMS dilengkapi dengan sistem untuk menangani berbagai jenis kegagalan yang mencegah hilangnya data. Mekanisme pemulihan DBMS memastikan bahwa *database* dapat dikembalikan ke keadaan yang konsisten setelah terjadi transaksi yang gagal karena crash sistem, kegagalan media, kesalahan perangkat keras atau lunak, mati listrik, dan sejenisnya.

5. *Concurrency control services*

DBMS memungkinkan berbagi data di antara banyak pengguna dengan menyediakan mekanisme untuk mengatur akses bersamaan ke *database*.

Sistem ini bertanggung jawab untuk menjaga konsistensi *database* dan integritas data, sehingga memastikan bahwa pembaruan *database* dilakukan dengan benar saat banyak pengguna melakukan pembaruan secara bersamaan.

6. Manajemen manipulasi data

Sistem manajemen basis data (DBMS) memungkinkan pengguna untuk mengambil, memperbarui, dan menghapus data yang ada dalam *database*, serta untuk menambahkan data baru ke dalamnya. Ini melibatkan penggunaan bahasa manipulasi data (DML) untuk mengatur operasi-operasi tersebut.

7. Manajemen katalog sistem

Sistem manajemen basis data (DBMS) memiliki kamus data atau katalog sistem yang berfungsi untuk menyimpan deskripsi dari item-item data. Katalog sistem ini dapat diakses oleh pengguna dan digunakan untuk menyimpan informasi tentang data, yang sering disebut sebagai metadata.

8. Otorisasi

DBMS melindungi basis data dari akses yang tidak sah, baik disengaja maupun tidak disengaja, dengan menyediakan cara untuk memastikan bahwa hanya pengguna yang memiliki izin yang dapat mengakses basis data. Ini menghasilkan sistem keamanan yang menjamin keamanan pengguna dan menjaga privasi data dalam basis data.

9. Layanan utilitas

DBMS menyediakan berbagai alat bermanfaat bagi administrator basis data (DBA) dan perancang basis data dalam seluruh proses pembuatan, implementasi, pemantauan, dan pemeliharaan basis data. Alat-alat ini mendukung DBA dalam mengelola basis data dengan lebih efisien.

10. Akses database dan antarmuka pemrograman aplikasi

Setiap sistem manajemen basis data (DBMS) menyediakan antarmuka yang memungkinkan aplikasi untuk berinteraksi dengan layanan yang disediakan oleh DBMS tersebut.

11. Layanan independensi data

DBMS harus memfasilitasi operasi program aplikasi tanpa memerlukan pengetahuan mendalam tentang struktur database yang sebenarnya.

12. Layanan definisi data

Sistem manajemen basis data menggabungkan definisi data dari skema eksternal, skema konseptual, skema internal, dan semua pemetaan yang terkait sebagai sumber informasi.

2.14. *PostgreSQL*

PostgreSQL adalah sebuah sistem manajemen basis data yang dikembangkan oleh *Department of Computer Science* di *University of California, Berkeley*. Database ini didukung banyak platform dan tersedia dengan lisensi gratis. *PostgreSQL* merupakan sistem basis data relasional yang mendukung penggunaan perintah-perintah *SQL* dan *PL/pgSQL (Procedural Language)*. Selain itu, *PostgreSQL* memiliki komunitas yang besar, dokumentasi yang komprehensif, dan mendukung berbagai bahasa pemrograman seperti *C++*, *Java*, *Perl*, *PHP*, *Python*, dan lainnya (Aji Bayu Prasetyo & Alfa Susetyo, 2022).

PostgreSQL adalah sistem DBMS relasional yang berorientasi objek (ORDBMS) open source. *PostgreSQL* mendukung bahasa kueri terstruktur (*SQL*) dengan kemampuan transaksi, *subquery*, *trigger*, dan fungsi lainnya. *PostgreSQL* memiliki fungsi sebagai tempat pengolahan dan penyimpanan data dalam *database* dengan fleksibel, serta memiliki kegunaan seperti manipulasi data seperti *insert*, *update*, *delete* (Ilham Taufiq et al., 2023).

PostgreSQL adalah DBMS yang tersedia sebagai perangkat lunak terbuka berdasarkan perjanjian lisensi BSD. Sistem basis data ini memungkinkan pengguna untuk memproses data dalam tabel yang terhubung dengan tabel lain dan dapat diubah secara bebas sesuai kebutuhan. Fitur tambahan yang terdapat dalam *PostgreSQL* diantaranya *PGPool*, *PGCluster*, *Slony*, *DB Mirror*, dan sebagainya (Gladys Wahyu Khairunnisa et al., 2022).

PostgreSQL, sebagai DBMS *relational* terbuka, telah mengalami pertumbuhan yang meningkat secara global. Selain memudahkan penggunaan, *PostgreSQL* juga terkenal karena kemampuan skalabilitas yang fleksibel dan

efisiensi yang tinggi (Yosef Jeffri Silvanus Nahak & Hindriyanto Dwi Purnomo, 2023).

PostgreSQL, yang juga dikenal sebagai Postgres, adalah salah satu basis data populer dikenal karena kemampuan untuk menangani skala besar, fleksibilitas, dan performa yang optimal. Arsitektur standarnya memfasilitasi pengguna untuk mengimplementasikan SQL mereka sendiri, terutama dalam pembuatan fungsi (Firda Rosiana Tanjung & Liptia Venica, 2023).

Dapat disimpulkan bahwa *PostgreSQL* adalah sebuah sistem manajemen basis data relasional yang mendukung berbagai platform dan tersedia dengan lisensi gratis. *Database* ini menyediakan dukungan penuh terhadap perintah *SQL* serta bahasa *prosedural PL/pgSQL*. Selain itu, *PostgreSQL* didukung oleh komunitas yang luas, dokumentasi yang komprehensif, serta integrasi dengan berbagai *bahasa* pemrograman seperti *C++*, *Java*, *Perl*, *PHP*, *Python*, dan banyak lainnya.

2.15. Application Programming Interface (API)

Application Programming Interface (API) merupakan *interface* aplikasi yang terdiri dari kumpulan instruksi yang disimpan dalam sebuah pustaka. Instruksi-instruksi tersebut memberitahu bagaimana sebuah aplikasi bisa berinteraksi dengan aplikasi yang lain untuk berbagi data (Priadi & Idris, n.d.).

API adalah penghubung di antara perangkat lunak yang memungkinkan dua aplikasi untuk berkomunikasi. Ketika kita menggunakan aplikasi seperti *WhatsApp* untuk mengirim pesan atau memeriksa kondisi cuaca, kita sebenarnya sedang memanfaatkan API untuk berinteraksi dengan sistem yang ada. (Salim et al., 2022).

Application Programming Interface (API) merupakan *interface* yang terdiri dari berbagai instruksi yang tersusun dalam sebuah pustaka atau perpustakaan. Fungsinya adalah untuk memungkinkan perangkat lunak berkomunikasi dan bertukar informasi dengan perangkat lunak lainnya dengan menyediakan serangkaian data dan fungsi (Ni Kadek Dwi Sabrina et al., 2023).

Application Programming Interface (API) adalah sebuah panduan yang menjelaskan elemen-elemen penting dalam pengembangan perangkat lunak, seperti antarmuka, kelas, fungsi, dan struktur yang dibutuhkan. Fungsinya sebagai perantara aplikasi atau situs web yang dibuat oleh pengembang dengan berbagai layanan yang tersedia. API memungkinkan akses dan pemanfaatan fungsionalitas

dari sistem lain, baik dalam lingkungan yang sama maupun lintas platform (Hajjul Ikram & Muhammad Ardiansyah, 2024).

Application Programming Interface (API) merupakan perangkat yang mengintegrasikan aplikasi untuk bertukar data. Fitur serupa tidak perlu dibuat ulang karena telah ada API yang menyediakannya. Contohnya, integrasi login menggunakan Google dan pengiriman pesan melalui Gmail. Salah satu desain arsitektur dalam API yaitu REST API yang menyediakan pertukaran informasi antara client dan server (Siti Sauda & M Barokah, 2022).

Berdasarkan teori tersebut disimpulkan bahwa, API (*Application Programming Interface*) merupakan antarmuka yang dapat berinteraksi antara dua aplikasi berbeda. Dengan menggunakan API, aplikasi dapat saling berkomunikasi, berbagi data, dan mengintegrasikan fungsionalitas satu sama lain. Misalnya, kita bisa menggunakan API *WhatsApp* untuk mengirim pesan atau API cuaca untuk menampilkan perkiraan cuaca.

2.16. Penelitian Terdahulu

Pada penelitian terdahulu ini, akan dijelaskan hasil dari beberapa penelitian yang pernah dilakukan orang lain sebagai sumber acuan dalam penulisan penelitian ini. Pemilihan topik penelitian terdahulu, relevan dengan topik yang dibahas dalam penulisan penelitian ini. Sehingga diharapkan mampu memberikan penjelasan dan dijadikan sebagai sumber referensi penulisan penelitian.

Tabel 2. 5 Penelitian Terdahulu

No	Penulis/Tahun	Judul Penelitian	Metode	Hasil	Perbedaan
1	Bagus Seta Inba Cipta, Pangestu Prima Darajat, Farchan Aziz mubarak 2021	Rancang Bangun Sistem Administrasi Keuangan Sekolah Berbasis Website Menggunakan <i>Framework Laravel</i> di SMP 1	<i>Waterfall</i>	Menghasilkan sebuah sistem administrasi keuangan	Metode yang digunakan, beberapa fitur dalam sistem, tempat penelitian,

No	Penulis/Tahun	Judul Penelitian	Metode	Hasil	Perbedaan
		Lukman Hakim Pakisaji			teknologi yang digunakan
2	Asrul Azhari Muin, Hermita 2020	Rancang Bangun Sistem Informasi Administrasi Keuangan Sekolah dan Penjadwalan Mata Pelajaran Berbasis <i>Web</i> pada Sma Negeri 1 Ambalawi	<i>Waterfall</i>	Menghasilkan sebuah sistem informasi administrasi keuangan	Metode yang digunakan, beberapa fitur dalam sistem, tempat penelitian, teknologi yang digunakan
3	Aldi prasetyo, Julianto Iemantara, Edo Yonatan Koentjoro 2019	Rancang Bangun Sistem Informasi Administrasi Keuangan Sekolah Berbasis <i>Web</i> pada Sman 1 Dawarblandong	<i>Waterfall</i>	Menghasilkan sebuah sistem informasi administrasi keuangan	Metode yang digunakan, beberapa fitur dalam sistem, tempat penelitian, teknologi yang digunakan
4	Yolanda Sherley Novitasari, Qadhi Jafar	Rancang Bangun Sistem Informasi Media Pembelajaran	<i>Extreme Programming</i>	Menghasilkan sebuah sistem informasi	Metode yang digunakan, sistem yang

No	Penulis/Tahun	Judul Penelitian	Metode	Hasil	Perbedaan
	Adrian, Wita Kurnia 2021	Berbasis <i>Website</i> (Studi Kasus: Bimbingan Belajar <i>De Potlood</i>)		media pembelajaran	dibuat, beberapa fitur dalam sistem, tempat penelitian, teknologi yang digunakan
5	Hafiz Riyadli, Arliyana, Fariez Eka Saputra 2020	Rancang Bangun Sistem Informasi Keuangan Berbasis <i>Web</i>	<i>Research and Development</i>	Menghasilkan sebuah sistem informasi keuangan	Metode yang digunakan, beberapa fitur dalam sistem, tempat penelitian, teknologi yang digunakan
6	I Gusti Ngurah Swala Putra, I Putu Satwika, I Gede Juliana Eka Putra 2020	Rancang Bangun Sistem Informasi Manajemen Administrasi Desa berbasis <i>Web</i> Menggunakan <i>Framework Laravel</i>	<i>Waterfall</i>	Menghasilkan sebuah sistem informasi manajemen administrasi desa	Metode yang digunakan, Sistem yang dibuat, fitur dalam sistem, tempat penelitian,

No	Penulis/Tahun	Judul Penelitian	Metode	Hasil	Perbedaan
					teknologi yang digunakan
7	Anjab Izdihar Hanisworo, Anggi Zafia 2022	Rancang Bangun Sistem Administrasi Keuangan berbasis <i>Website</i> (Studi Kasus: MTS Al-Ittihaad Darusa'adaah Pasir Kidul)	<i>SCRUM</i>	Menghasilkan sebuah sistem administrasi keuangan	Beberapa fitur dalam sistem, tempat penelitian, teknologi yang digunakan
8	Risa Anggraini 2021	Rancang Bangun Sistem Informasi Administrasi Pengelolaan Dana Masjid berbasis <i>Web</i> (Studi Kasus: Masjid Al-Muttaqin)	<i>Extreme Programming</i>	Menghasilkan sebuah sistem informasi administrasi pengelolaan dana	Metode yang digunakan, fitur dalam sistem, tempat penelitian, teknologi yang digunakan
9	Eva Oktaviani, Rodianto, Sari Noviana, Nawassyarif 2020	Rancang Bangun Sistem Informasi untuk Meningkatkan Tata Kelola Administrasi Surat Menyurat	<i>Waterfall</i>	Menghasilkan sebuah sistem informasi administrasi Surat Menyurat	Metode yang digunakan, sistem yang dibuat, fitur dalam sistem,

No	Penulis/Tahun	Judul Penelitian	Metode	Hasil	Perbedaan
					tempat penelitian, teknologi yang digunakan
10	Dini Destiani Siti Fatimah, Yeni Paryatin, Nisa Nurhasanah 2022	Rancang Bangun Sistem Informasi Pelayanan Administrasi Surat Desa berbasis <i>Web</i> Menggunakan Metode <i>Unified Approach</i>	<i>Unified Approach</i>	Menghasilkan sebuah sistem pelayanan administrasi Surat	Metode yang digunakan, sistem yang dibuat, fitur dalam sistem, tempat penelitian, teknologi yang digunakan

BAB III METODOLOGI PENELITIAN

3.1. Objek Penelitian

Objek penelitian ini dilakukan di Sekolah Menengah Atas Swasta yang bernama SMA Plus Nurul Iman Leles berlokasi di Desa Cipancar, Kecamatan Leles, Kabupaten Garut, Provinsi Jawa Barat.

3.1.1. Profil Instansi

SMA Plus Nurul Iman Leles merupakan sebuah lembaga pendidikan menengah atas yang didirikan sejak tahun 2011, berada di bawah naungan yayasan AL-MUAWANAH, dengan dedikasi yang kuat terhadap pendidikan berkualitas. Sekolah ini, telah berperan sebagai salah satu lembaga yang memberikan kesempatan belajar dan berkembang bagi generasi muda di daerah pedesaan yang ada di Kabupaten Garut.

Adapun Visi dan Misi dari SMA Plus Nurul Iman leles sebagai berikut.

1. Visi

Menciptakan kualitas lulusan yang mandiri, berbudi pekerti luhur dan religius.

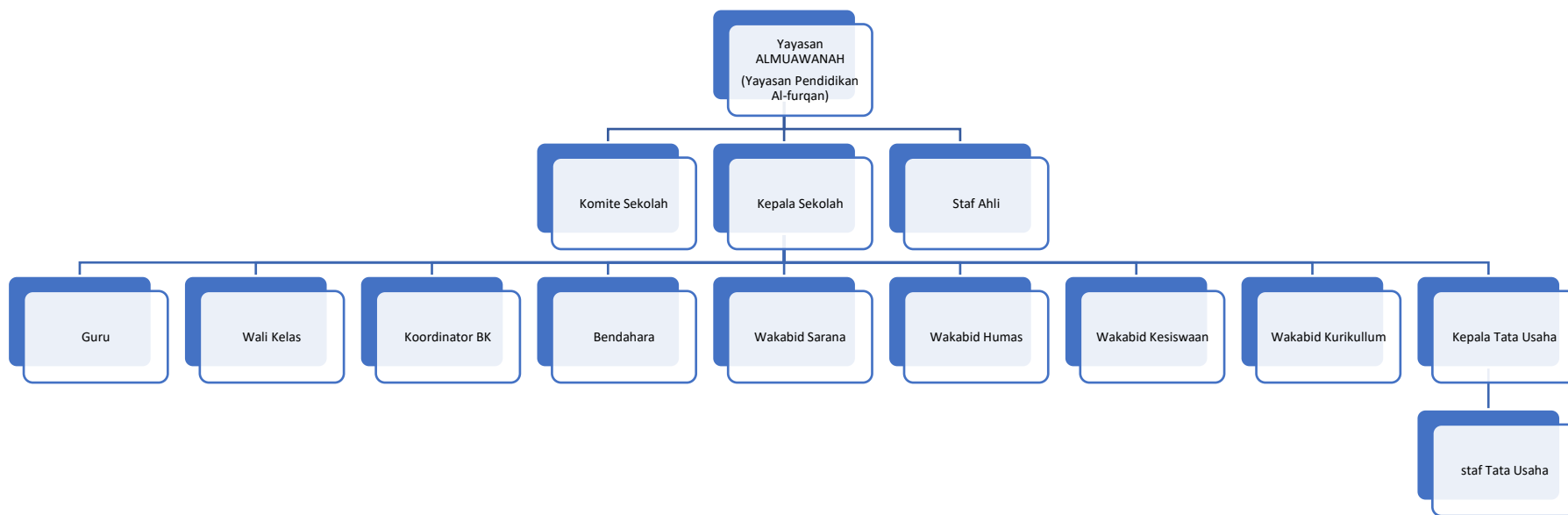
2. Misi

Adapun misi dari SMA Plus Nurul Iman sebagai berikut:

- a. Meningkatkan proses belajar mengajar.
- b. Meningkatkan mutu pendidikan dan unggul dalam prestasi.
- c. Melaksanakan akhlak budi pekerti.
- d. Mewujudkan personal yang sehat dan terampil beretos kerja.
- e. Memberdayakan sumber daya pendidikan secara efektif dan efisien.
- f. Menempatkan guru yang profesional dan berdedikasi tinggi.

3.1.2. Struktur Organisasi

SMA Plus Nurul Iman Leles memiliki struktur organisasi yang meliputi ketua yayasan sebagai pemimpin dan penentu kebijakan tertinggi. Kepala sekolah, komite, staff ahli, dan wakil kepala sekolah sebagai pengelola sekolah secara keseluruhan. Terdapat juga koordinator bidang, guru mata pelajaran, staf tata usaha yang mendukung operasional sekolah. Adapun struktur organisasi bisa dilihat pada gambar 3.1 berikut.



Gambar 3. 1 Struktur Organisasi SMA Plus Nurul Iman Leles

Sumber: Bapak Cecep Somantri, S.Pd. (Staff Tata Usaha) SMA Plus Nurul Iman Leles Tahun 2024

Adapun berikut ini merupakan penjelasan terkait tugas dan kewenangan masing-masing jabatan.

1. Yayasan
 - a. Menyediakan fasilitas sekolah
 - b. Memelihara fasilitas sekolah
 - c. Melakukan pengangkatan dan pemberhentian kepala sekolah
 - d. Terlibat dalam pengambilan keputusan
 - e. Melakukan Pengawasan kepatuhan hukum
2. Komite Sekolah
 - a. Menetapkan kebijakan dan program sekolah
 - b. Memberikan masukan dan saran kepada kepala sekolah
 - c. Mengawasi pelaksanaan kebijakan dan program sekolah
 - d. Melakukan evaluasi kinerja kepala sekolah
3. Kepala Sekolah
 - a. Merencanakan, mengorganisir, dan mengelola operasional sekolah
 - b. Menetapkan kebijakan dan program sekolah
 - c. Mengawasi dan memimpin staf sekolah
 - d. Bertanggung jawab akan keselamatan dan kedisiplinan siswa
4. Staf Ahli
 - a. Memberikan saran dan masukan kepada kepala sekolah
 - b. Membantu kepala sekolah dalam pengambilan keputusan strategis
 - c. Melakukan analisis dan evaluasi terhadap kebijakan dan program sekolah
5. Guru
 - a. Merencanakan dan melaksanakan proses pembelajaran
 - b. Mengevaluasi perkembangan siswa
 - c. Melibatkan diri dalam kegiatan sekolah, seperti rapat guru dan kegiatan ekstrakurikuler
6. Wali Kelas
 - a. Membimbing dan mengawasi siswa di dalam kelas
 - b. Melakukan evaluasi dan pelaporan perkembangan siswa

- c. Menjalin komunikasi dengan orangtua siswa terkait perkembangan siswa
 - d. Mengkoordinasikan kegiatan dan tugas siswa di kelas
7. Koordinator Bimbingan Konselling
- a. Memberikan bimbingan dan konselling kepada siswa
 - b. Melakukan kegiatan penyuluhan dan orientasi untuk siswa
 - c. Membantu siswa dalam pemecahan masalah pribadi dan akademik
8. Bendahara
- a. Mengelola penerimaan dan pengeluaran dana
 - b. Menyusun laporan keuangan secara berkala
 - c. Membuat dan memelihara catatan keuangan yang akurat dan teratur
 - d. Berpartisipasi dalam penyusunan anggaran sekolah
9. Wakil Kepala Bidang Sarana
- a. Mengelola dan memelihara fasilitas sekolah
 - b. Mengkoordinasikan perawatan dan perbaikan fasilitas
 - c. Mengatur penggunaan ruang kelas dan fasilitas lainnya
10. Wakil Kepala Bidang Humas
- a. Mengelola hubungan dengan masyarakat, termasuk media masa
 - b. Menyusun dan menyebarkan informasi terkait sekolah
 - c. Mengkoordinasikan kegiatan promosi dan publisitas sekolah
11. Wakil Kepala Bidang Kesiswaan
- a. Mengawasi kesejahteraan dan perkembangan siswa
 - b. Mengkoordinasikan kegiatan-kegiatan ekstrakurikuler
 - c. Menangani masalah kedisiplinan siswa
12. Wakil Kepala Bidang Kurikulum
- a. Merencanakan, mengembangkan dan mengevaluasi kurikulum sekolah
 - b. Mengawasi implementasi kurikulum sekolah
 - c. Melakukan evaluasi dan pemantauan terhadap kegiatan pembelajaran
13. Kepala Tata Usaha
- a. Mengelola administrasi sekolah termasuk keuangan dan kepegawaian
 - b. Mengkoordinasikan kegiatan-kegiatan administrasi dan operasional sekolah

14. Staf Tata Usaha

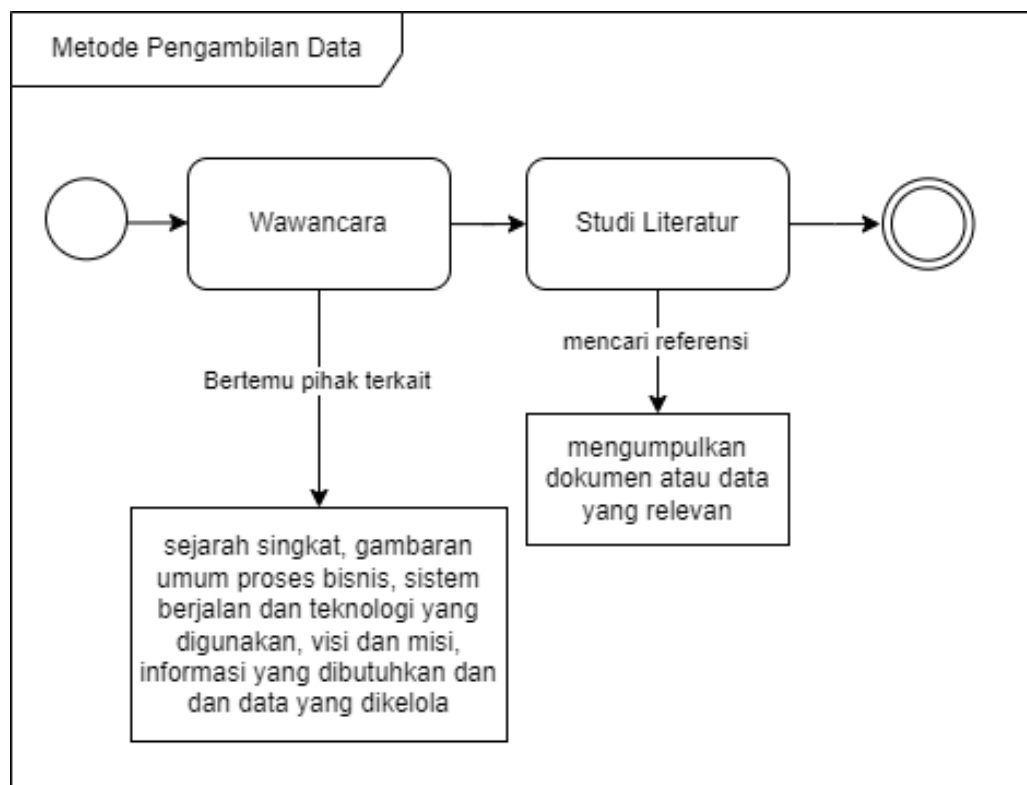
- Melakukan tugas-tugas administratif seperti pengarsipan, pengelolaan surat menyurat dan penerimaan tamu
- Membantu kepala tata usaha dalam tugas-tugas administratif
- Mengelola inventaris sekolah
- Menyediakan dukungan administratif kepada staf dan guru

3.2. Metode dan Desain Penelitian

Metode dan desain penelitian adalah serangkaian langkah sistematis yang digunakan untuk merencanakan, melaksanakan, dan mengevaluasi sebuah penelitian. Metode ini mencakup teknik-teknik dan prosedur-prosedur untuk mengumpulkan, menganalisis data, dan menarik kesimpulan dari hasil penelitian. Metode dan desain penelitian yang dipakai penulis yaitu sebagai berikut.

3.2.1. Teknik Pengambilan Data

Untuk menghasilkan berbagai data yang dibutuhkan dari objek penelitian, maka dipakai beberapa metode pengambilan atau pengambilan data, yaitu sebagai berikut:



Gambar 3. 2 Teknik Pengambilan Data

Berikut uraian dari masing-masing proses metode pengumpulan data.

1. Wawancara

Pada tahapan wawancara, penulis melakukan diskusi atau tanya jawab secara langsung dengan para *stakeholder* yang ada di SMA Plus Nurul Iman Leles. Wawancara yang dilakukan menghasilkan informasi sebagai berikut.

- a. Sejarah singkat atau profil.
- b. Sistem yang sedang berjalan.
- c. Visi dan misi.
- d. Gambaran umum tentang proses bisnis yang dilakukan.
- e. Sistem atau teknologi yang saat ini digunakan.
- f. Informasi yang dibutuhkan oleh pemegang kebijakan.
- g. Data yang dikelola, dan sebagainya.

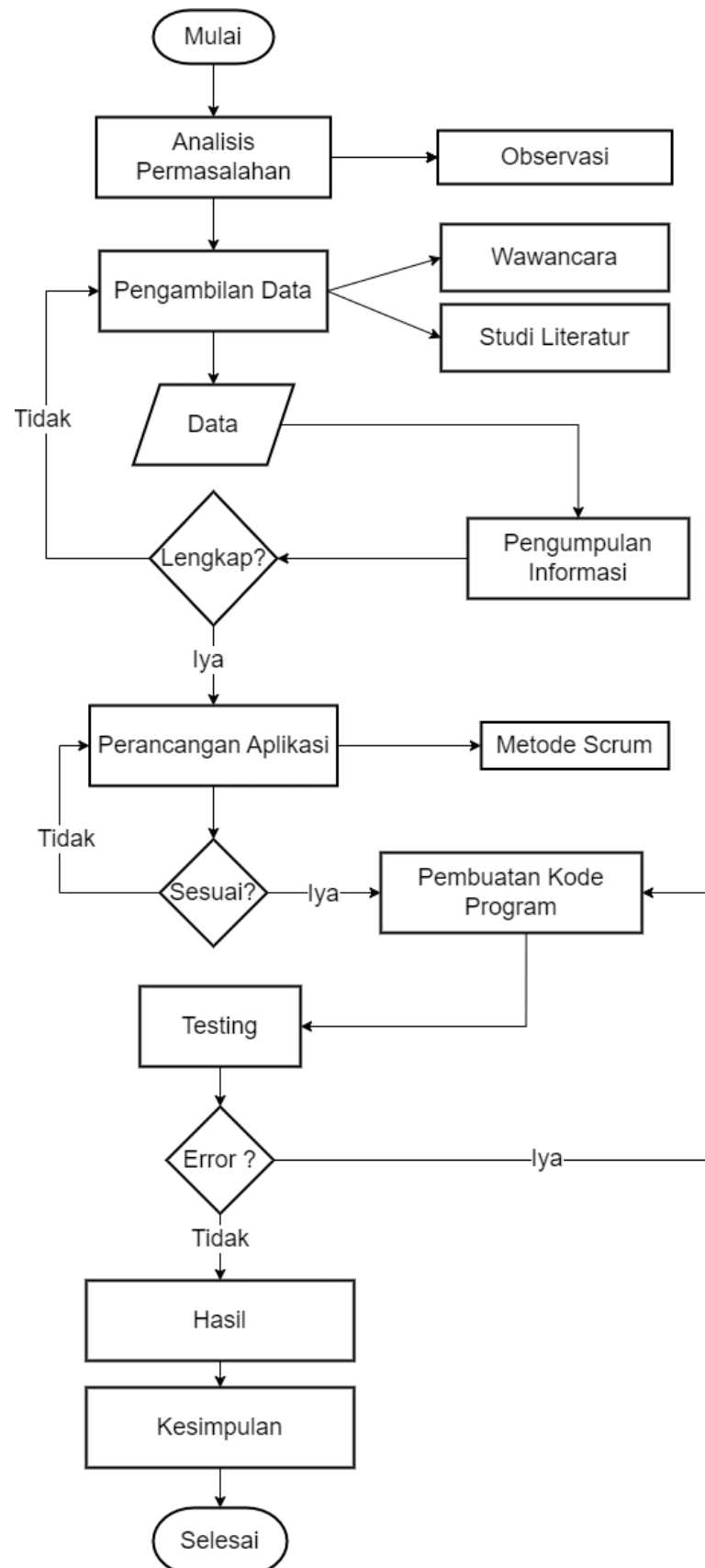
2. Studi Literatur

Selanjutnya, dilakukan studi literatur. Kegiatan ini melibatkan pengumpulan dan perbandingan beberapa referensi dokumen yang relevan dalam pembuatan laporan. Hal ini bertujuan untuk menentukan kesetaraan, kelebihan, dan kekurangan setiap referensi, serta menjadikan referensi tersebut sebagai acuan yang *valid*. Pada tahapan ini, referensi diambil dari jurnal, skripsi, dan sumber literatur terpercaya

Dengan menjalankan tahapan-tahapan ini, penelitian ini dapat mengumpulkan data yang valid dan akurat mengenai administrasi keuangan di SMA Plus Nurul Iman Leles serta memberikan landasan yang kuat untuk menyusun laporan yang komprehensif

3.2.2. Kerangka Berfikir

Kerangka berfikir adalah struktur konseptual yang digunakan untuk memandu pemikiran dan tindakan, baik dalam pemecahan masalah, analisis situasi, maupun pengembangan perangkat lunak. Di bawah merupakan kerangka berfikir yang digunakan penulis.



Gambar 3. 3 Kerangka Berfikir

Berikut penulis uraikan kerangka berfikir dari tahap awal sampai akhir sesuai dengan pada gambar 3.3.

1. Analisis Permasalahan

Tahapan pertama yang dilakukan adalah analisis masalah. Pada tahap ini, penulis menganalisis dengan mendatangi langsung (observasi) tempat atau objek penelitian untuk memahami permasalahan yang sedang dihadapi dan mendapatkan pemahaman yang lebih baik tentang konteks masalah. Dengan demikian, analisis permasalahan menjadi landasan yang kuat untuk merumuskan solusi yang tepat dan efektif.

2. Pengumpulan atau Pengambilan Data

Pada tahap ini, melakukan pengambilan data untuk mendapatkan informasi atau fakta yang sesuai untuk tujuan penelitian atau analisis. Data dikumpulkan melalui metode seperti, wawancara dan studi literatur dengan maksud menghasilkan informasi yang akurat dan dapat dipercaya. Proses pengumpulan data akan dilakukan dengan cermat sesuai dengan metode penelitian yang telah direncanakan.

3. Pengumpulan Informasi

Selanjutnya penulis akan mengumpulkan informasi dengan cara mengolah semua data yang sudah dikumpulkan sebelumnya. Informasi adalah data yang telah diolah melalui tahap analisis dengan metode yang telah ditetapkan. Informasi ini diharapkan dapat menambah pemahaman yang lebih dalam atau dapat digunakan untuk membuat keputusan atau rekomendasi. Jika informasi yang di dapat belum sesuai, maka akan kembali melakukan pengumpulan data.

4. Perancangan Aplikasi

Pada tahap perancangan aplikasi, penulis mulai menguraikan bagaimana aplikasi akan dibangun sesuai dengan kebutuhan yang didapat dari informasi yang telah dianalisis sebelumnya. Perancangan ini mencakup pemilihan teknologi, arsitektur sistem, antarmuka pengguna, dan rincian implementasi yang diperlukan. Tahap perancangan ini penting untuk memastikan bahwa aplikasi yang akan dibangun sesuai dengan tujuan dan kebutuhan yang telah ditetapkan sebelumnya. Hasil rancangan ini, akan

dikomunikasikan terlebih dahulu dengan pihak terkait. Apabila dirasa ada yang kurang maka akan dilakukan perancangan kembali.

5. Pembuatan Kode Program

Pada tahap ini, penulis membuat kode program sesuai konsep dan desain yang telah dirancang sebelumnya, supaya bisa diimplementasikan menjadi kode yang dapat dieksekusi oleh komputer. Tahap ini membutuhkan pengetahuan yang baik tentang bahasa pemrograman yang dipakai, struktur data, algoritma, dan prinsip pemrograman yang baik dan benar. Proses pembuatan kode program memang bisa menjadi tahapan yang cukup sulit karena memerlukan ketelitian dan kejelitan dalam menulis program agar sesuai dengan perancangan yang telah dibuat.

6. *Testing*

Melakukan pengujian setiap fitur yang ada pada aplikasi. Hal ini bertujuan untuk memastikan bahwasannya aplikasi sudah berjalan normal atau tidak. Pengujian ini mencakup pengujian fungsionalitas, keamanan, kinerja, dan keandalan aplikasi. Apabila masih terdapat *error* atau *bug*, maka akan dilakukan pemeriksaan dan pengkodean kembali.

7. Hasil

Setelah aplikasi menjalani tahap pengujian supaya bisa memastikan bahwa, aplikasi berjalan normal dan memenuhi kebutuhan yang telah ditetapkan, maka aplikasi tersebut dapat diintegrasikan dan dimanfaatkan oleh pengguna sesuai dengan keperluan yang diinginkan.

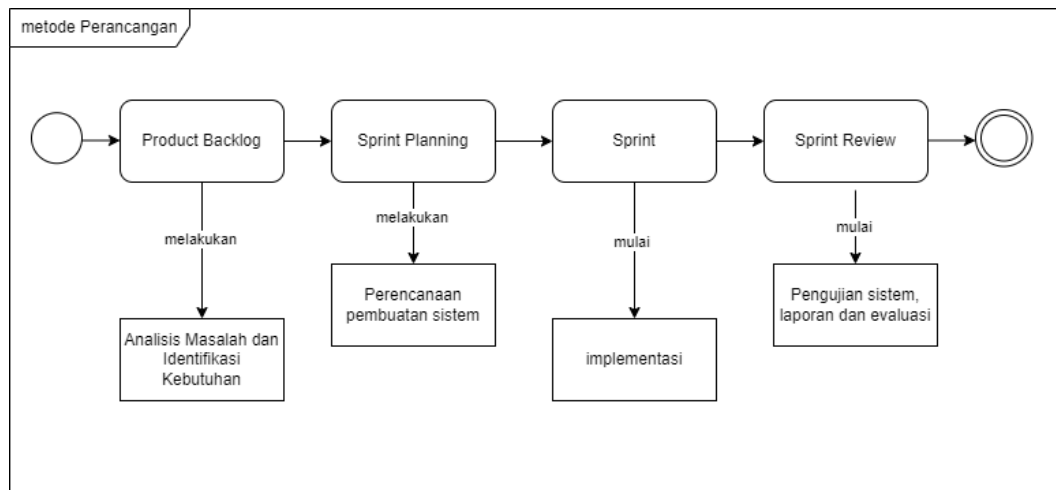
8. Kesimpulan

Tahap ini merupakan bagian akhir dari semua tahapan yang ada. Dimana pada bagian ini hanya berisi kesimpulan tentang seluruh hal yang terkait dengan penulisan laporan serta berisi sikap akhir yang disampaikan oleh penulis.

3.2.3. Metode Perancangan

Dalam merancang sistem informasi, penulis menggunakan metode perancangan *scrum* untuk menghasilkan solusi yang baik. Metode perancangan ini membantu penulis dalam merancang struktur sistem informasi dengan lebih terstruktur dan terorganisir. Dengan demikian, sistem informasi yang dihasilkan

akan lebih mudah dipahami, diimplementasikan, dan dipelihara. Adapun tahapan dari metode perancangan ini bisa dilihat pada gambar 3.4.



Gambar 3. 4 Metode Perancangan

1. *Product Backlog*

Pada tahap ini, penulis akan menganalisis sistem yang sedang berjalan untuk mengidentifikasi masalah atau kekurangan yang mungkin ada dalam sistem tersebut. Selain itu, penulis juga akan melakukan identifikasi terhadap kebutuhan apa saja, untuk meningkatkan atau mengatasi masalah yang ada dan disusun menjadi *product backlog*. Analisis ini penting untuk memastikan bahwa solusi yang akan diimplementasikan nantinya akan sesuai dengan kebutuhan dan dapat menyelesaikan masalah yang ada dalam sistem.

2. *Sprint Planning*

Pada tahap ini, penulis akan menyusun *product backlog* menjadi bentuk *sprint* dengan cara memilih item-item dari *product backlog* dan membuat rencana bagaimana pekerjaan akan diselesaikan dalam sprint tersebut.

3. *Sprint*

Tahap selanjutnya adalah *sprint*. Pada tahap ini, penulis akan mengimplementasikan setiap item dalam *sprint planning* menjadi fungsionalitas yang siap untuk dipresentasikan.

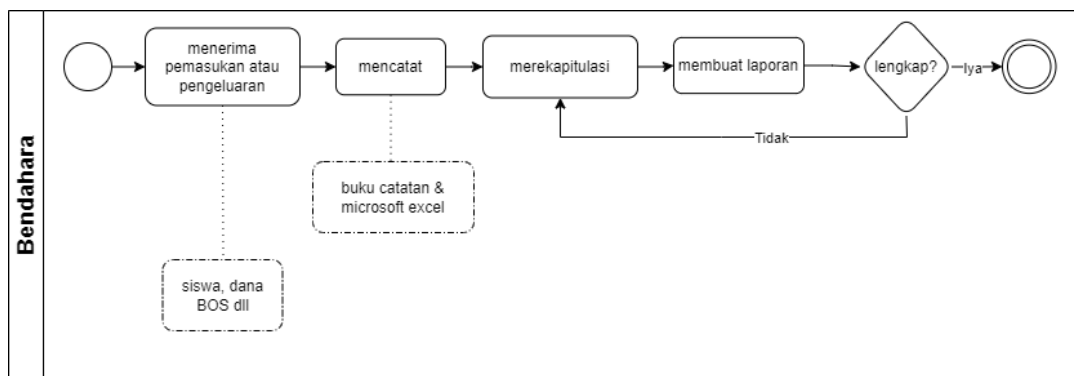
4. *Sprint Review*

Pada *sprint review*, penulis akan melakukan pengujian sekaligus mempresentasikan hasil pembuatan sistem kepada pihak sekolah. Hal ini

bertujuan supaya pihak sekolah bisa mengetahui perkembangan proyek, memberikan umpan balik, dan memastikan bahwa hasil yang dicapai sesuai dengan yang diharapkan.

3.2.4. Sistem yang Sedang Berjalan

SMA Plus Nurul Iman Leles masih menggunakan sistem administrasi keuangan yang dilakukan dengan cara manual menggunakan buku catatan dan *Microsoft Excel*. Sistem yang sedang berjalan bisa dilihat pada gambar 3.5.



Gambar 3. 5 Sistem yang Sedang Berjalan

Adapun penjelasan dari sistem yang sedang berjalan sebagai berikut.

Pertama-tama, bendahara sekolah akan menerima transaksi baik itu pemasukan ataupun pengeluaran. Pemasukan diperoleh dari berbagai sumber, seperti biaya sekolah dari siswa dan dana Bantuan Operasional Sekolah (BOS), sedangkan pengeluaran mencakup pembelian Alat Tulis Kantor (ATK) dan pembayaran gaji guru. Setelah itu, dilakukan pencatatan menggunakan media seperti buku dan *Microsoft Excel*. Kemudian, sewaktu-waktu pihak sekolah akan merekapitulasi transaksi tersebut dan membuat laporan akhir. Apabila laporan akhir tidak lengkap maka akan dilakukan rekapitulasi ulang.

BAB IV HASIL DAN PEMBAHASAN

4.1. Analisis Sistem yang Sedang Berjalan

Secara garis besar, sistem yang berjalan di SMA Plus Nurul Iman Leles masih menggunakan sistem administrasi keuangan yang dilakukan dengan cara manual menggunakan buku catatan dan *Microsoft Excel*.

4.2. Analisis Kelemahan Sistem yang Sedang Berjalan

Untuk mengidentifikasi lebih dalam terkait masalah yang terjadi, maka analisis kelemahan dilakukan pada sistem. Penulis menganalisis berdasarkan pada beberapa aspek seperti dalam kinerja, informasi, pelayanan dan efisiensi. Adapun hasil analisis kelemahan sistem bisa dilihat pada tabel 4.1 berikut.

Tabel 4. 1 Analisis Kelemahan Sistem

Jenis Analisis	Sistem Lama	Solusi
Kinerja	Keterlambatan dalam proses pencatatan dan pelaporan keuangan karena dilakukan secara manual. Potensi kesalahan dalam pengolahan data dan perhitungan keuangan.	Mengimplementasikan sistem basis data untuk menyimpan dan mengelola data keuangan secara terstruktur. Menggunakan perangkat lunak akuntansi yang terintegrasi untuk mempercepat proses pencatatan dan pelaporan keuangan.
Informasi	Kesulitan dalam mengakses dan memperoleh informasi keuangan secara cepat dan akurat. Risiko kehilangan atau kerusakan pada data karena penyimpanan manual di buku catatan.	Memigrasikan penyimpanan data ke format digital yang aman dan terstruktur. Menggunakan sistem manajemen basis data (DBMS) seperti <i>PostgreSQL</i> untuk mengelola data keuangan.
Efisiensi (<i>Effeciency</i>)	Proses yang lambat dan rentan terhadap kesalahan	Mengotomatisasi proses pencatatan dan pelaporan keuangan menggunakan

Jenis Analisis	Sistem Lama	Solusi
	mengakibatkan kurang efisiennya pengelolaan keuangan. Keterbatasan dalam menyediakan informasi keuangan secara cepat dan tepat waktu.	perangkat lunak yang sesuai. Menggunakan sistem yang dapat menghasilkan laporan keuangan secara otomatis untuk meningkatkan efisiensi.
Pelayanan (<i>Service</i>)	Kemungkinan terjadinya keterlambatan dalam pelayanan kepada pihak yang membutuhkan informasi keuangan. Tidak adanya aksesibilitas yang memadai terhadap informasi keuangan untuk semua pihak yang berkepentingan.	Menerapkan sistem yang menyediakan akses informasi keuangan dengan <i>real-time</i> untuk pihak yang berwenang. Memberikan pelatihan kepada staf terkait dengan penggunaan sistem baru untuk memastikan pelayanan yang lebih baik.

4.3. Identifikasi Kebutuhan Sistem

Identifikasi kebutuhan ditujukan untuk memahami dan mendefinisikan apa yang harus dicapai oleh sistem yang akan dibuat. Berikut ini adalah daftar kebutuhan dari sistem yang akan dibangun.

1. Kebutuhan Informasi

Adapun kebutuhan informasi yang menjadi dorongan untuk pembangunan sistem informasi ini, sebagai berikut:

- a. Kebutuhan informasi data keuangan secara lengkap seperti laporan keuangan tahunan dan bulanan.
- b. Kebutuhan informasi terkait detail transaksi yang mencakup pembayaran atau penerimaan dan pengeluaran
- c. Kebutuhan akan proses bisnis yang sedang berjalan terkait dengan administrasi keuangan mulai dari pencatatan transaksi hingga laporan keuangan.
- d. Kebutuhan informasi terkait kebijakan dan regulasi pengelolaan keuangan di sekolah, termasuk standar pelaporan keuangan yang harus dipenuhi.

2. Kebutuhan Perangkat Lunak

Software yang dibutuhkan untuk merancang dan membangun sistem informasi ini antara lain:

- a. *IDE Visual Studio Code*
- b. *Draw.io*
- c. *PostgreSQL*
- d. *Web Browser*
- e. *Docker*

3. Pengguna Sistem

Pengguna sistem informasi administrasi keuangan ini hanya ada Admin (bendahara) yang memiliki hak akses penuh terhadap sistem dalam kata lain sebagai pengelola data keuangan.

4. Kebutuhan fitur dalam aplikasi

Fitur aplikasi yang dibutuhkan pada sistem bisa dilihat pada tabel 4.2 di bawah ini.

Tabel 4. 2 Fitur Aplikasi

No	Item	Skala Prioritas
1.	Halaman <i>Login</i>	Tinggi
2.	Halaman <i>Reset Password</i>	Tinggi
3.	<i>Dashboard</i>	Rendah
4.	Halaman Siswa	Sedang
5.	Halaman Guru	Sedang
6.	Halaman Pencatatan	Tinggi
7.	Halaman Laporan	Tinggi

4.4. Perencanaan Pembuatan Sistem

Perencanaan dibuat dengan mengidentifikasi item-item dari *product backlog* yang akan dikerjakan selama Sprint berikutnya. Penulis juga akan menentukan bagaimana cara menyelesaikan pekerjaan tersebut dan merencanakan kebutuhan sumber daya yang diperlukan.

4.4.1. Pembagian *Sprint*

Dalam perancangan *sprint planning* kali ini, penulis membuat tujuh *sprint* yang akan dieksekusi. Seluruh *sprint* ini bertujuan untuk merampungkan pengembangan fitur-fitur utama yang telah diprioritaskan dalam *Product Backlog*. Adapun pembagian *sprint* telah penulis susun pada tabel 4.3 di bawah ini.

Tabel 4. 3 Pembagian *Sprint*

No	<i>Sprint</i>	<i>Product Backlog</i>
1.	<i>Sprint 1</i>	Perancangan Sistem
2.	<i>Sprint 2</i>	<i>Login, Verifikasi Email, Reset Password</i>
3.	<i>Sprint 3</i>	Data Siswa, Data Kelas, Data Jurusan
4.	<i>Sprint 4</i>	Data Guru
5.	<i>Sprint 5</i>	Pembayaran Siswa, Pemasukan, Transaksi, Pengeluaran, Kas Guru
6.	<i>Sprint 6</i>	Laporan Pemasukan dan Pengeluaran, Laporan Siswa, Laporan Guru
7.	<i>Sprint 7</i>	<i>Dashboard</i>

4.4.2. Kelengkapan *Sprint*

Bagian ini memberikan informasi mengenai kelengkapan setiap *sprint* yang sudah disusun sebelumnya. Hal ini dibuat untuk memudahkan dalam menentukan fitur atau *task* apa saja yang akan dikerjakan pada setiap *sprint*. Adapun kelengkapan *sprint* bisa dilihat pada tabel 4.4 di bawah ini.

Tabel 4. 4 Kelengkapan *Sprint*

No	<i>Sprint</i>	
1.	<i>Sprint 1</i>	Keterangan
	Pembuatan UML	Pembuatan <i>Use Case Diagram</i>
		Pembuatan <i>Activity Diagram</i>
		Pembuatan <i>Sequence Diagram</i>

No	<i>Sprint</i>	
	Pembuatan skema basis data	Pembuatan <i>Entity Relationship Diagram</i>
	Desain Aplikasi	Implementasi tampilan Halaman <i>Login</i>
		Implementasi tampilan halaman <i>Verifikasi Email</i>
		Implementasi tampilan halaman <i>Reset Password</i>
		Implementasi tampilan halaman <i>Dashboard</i>
		Implementasi tampilan halaman Siswa
		Implementasi tampilan halaman <i>Stakeholders</i>
		Implementasi tampilan halaman Pencatatan
		Implementasi tampilan halaman Laporan Keuangan
		Implementasi tampilan <i>Logout</i>
2.	<i>Sprint 2</i>	Keterangan
	<i>Login</i>	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
	<i>Verifikasi email</i>	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
	<i>Reset Password</i>	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
3.	<i>Sprint 3</i>	Keterangan
	Fitur data siwa	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>

No	<i>Sprint</i>	
	Fitur data kelas	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
	Fitur data jurusan	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
4.	<i>Sprint 4</i>	
	Fitur Guru	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
5.	<i>Sprint 5</i>	
	Fitur pembayaran siswa	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
	Fitur pemasukan	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
	Fitur transaksi	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
	Fitur Pengeluaran	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
	Fitur Kas Guru	Implementasi kode program (<i>Front-End</i>)

No	<i>Sprint</i>	
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
6.	<i>Sprint 6</i>	Keterangan
	Fitur laporan keuangan	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
	Fitur laporan siswa	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
	Fitur laporan guru	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>
7.	<i>Sprint 7</i>	Keterangan
	<i>Dashboard</i>	Implementasi kode program (<i>Front-End</i>)
		Implementasi kode program (<i>Back-End</i>)
		<i>Unit Testing</i>

4.5. Implementasi

Berikut ini merupakan implementasi dari setiap item yang telah disusun, selama proses perencanaan sistem informasi administrasi keuangan di SMA Plus Nurul Iman Leles.

4.5.1. Desain Sistem

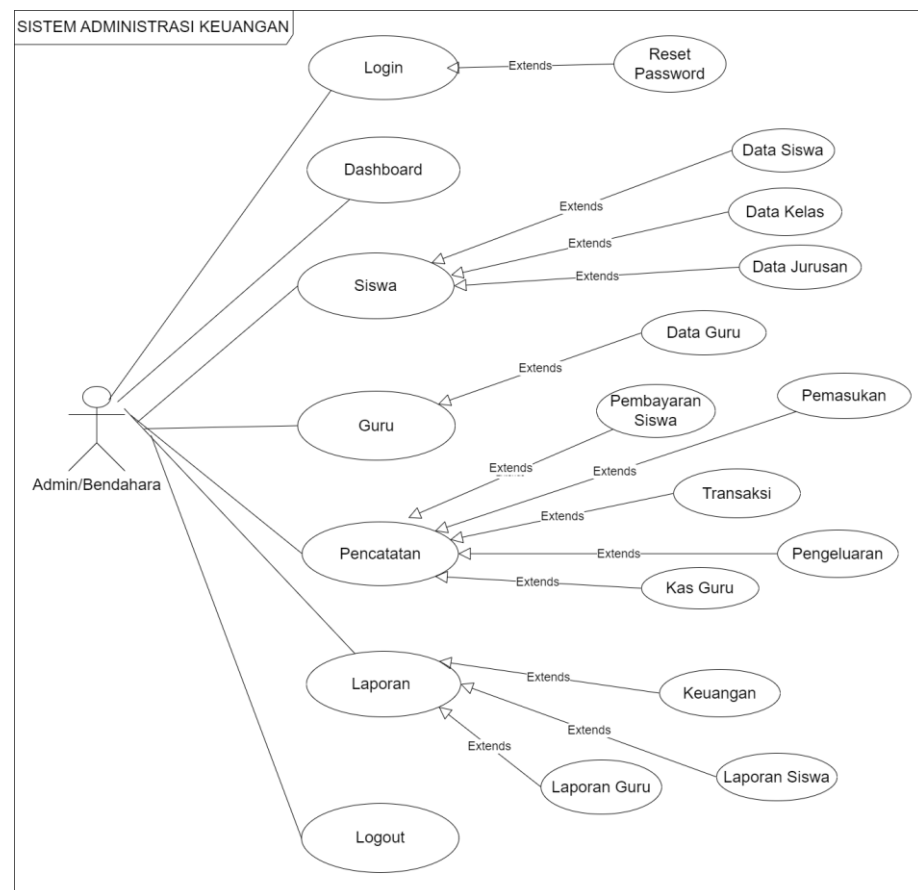
Sistem informasi administrasi keuangan SMA Plus Nurul Iman Leles, memiliki desain sistem yang terdiri dari desain proses, desain basis data, dan desain *interface*, sebagai bahan perancangan fitur-fitur pelaporan keuangan yang lengkap dan mudah digunakan. Berikut merupakan desain sistem yang telah dibuat.

1. Desain Proses

Pada desain proses, penulis menggunakan *Unified Modelling Language* (UML) untuk memberikan gambaran proses pada sistem informasi ini. Adapun diagram yang dipakai yaitu *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram*.

a. *Use Case Diagram*

Berikut ini merupakan *usecase diagram* pada sistem informasi yang akan dibangun.



Gambar 4. 1 *Use Case Diagram*

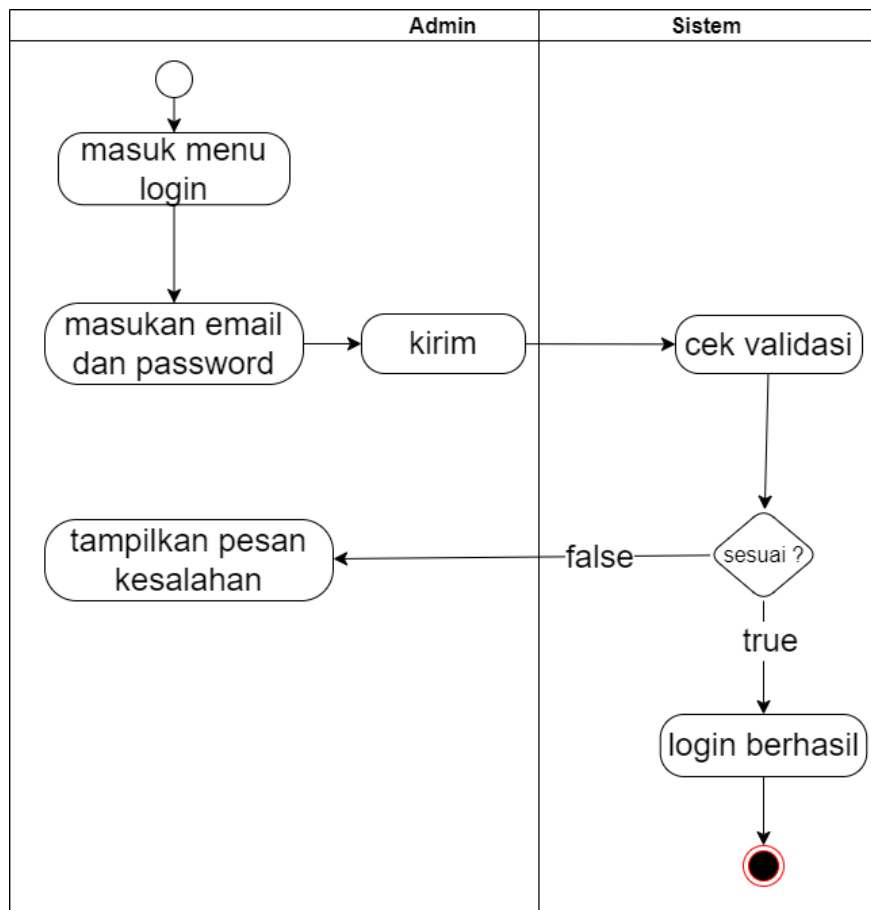
Pada gambar tersebut, *Admin* (bendahara) dapat mengakses menu *login*, *dashboard*, *siswa*, *guru*, *pencatatan*, *laporan* dan *logout*. Beberapa menu mempunyai sub-menu lagi, seperti pada menu *pencatatan* yang memiliki sub-menu *pembayaran siswa*, *pemasukan*, *transaksi*, *pengeluaran* dan *kas guru*, yang dilengkapi fitur untuk melakukan CRUD (*Create*, *Read*, *Update*, *Delete*) data.

b. *Activity Diagram*

Berikut ini merupakan *Activity Diagram* pada sistem informasi yang akan dibangun.

1). *Activity Diagram Login*

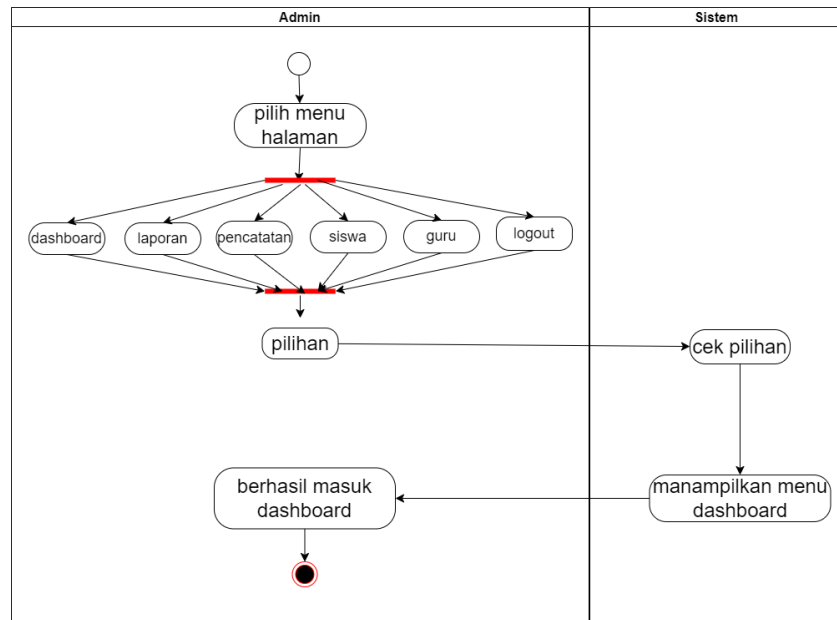
Berdasarkan gambar di bawah ini, proses ini dimulai dengan memasuki halaman login, menginput email dan *password*, kemudian sistem akan memvalidasi kecocokan. Jika sesuai maka *login* berhasil, jika tidak maka akan muncul pesan kesalahan.



Gambar 4. 2 *Activity Diagram Login*

2). *Activity Diagram Dashboard*

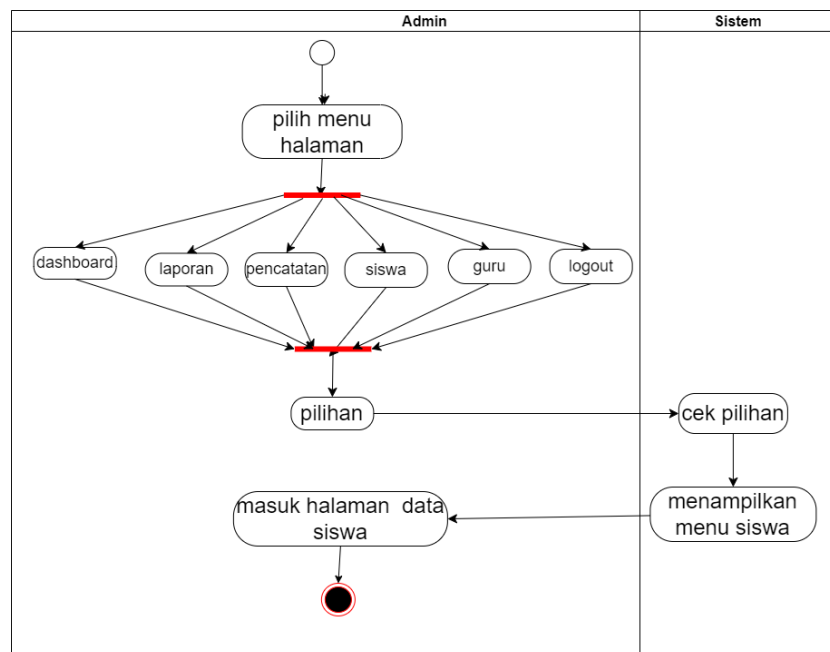
Proses ini berhubungan dengan proses *login*, yang dimana apabila berhasil maka secara otomatis akan diarahkan ke menu *dashboard* atau jika *Admin* sedang berada di halaman lain, *Admin* tinggal memilih menu *dashboard* maka akan langsung diarahkan ke menu tersebut.



Gambar 4. 3 *Activity Diagram Dashboard*

3). *Activity Diagram Siswa*

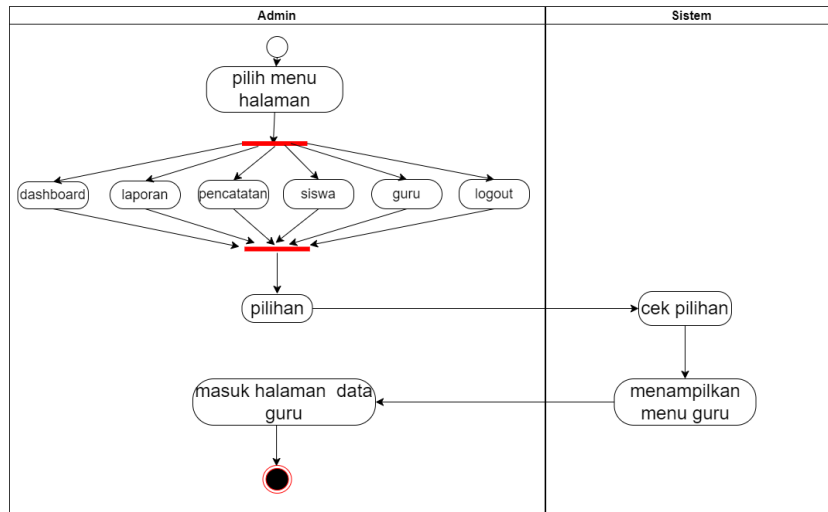
Proses ini diawali ketika berada di menu halaman, pengguna memilih menu siswa. Sistem akan mengecek pilihan dan menampilkan menu siswa. Di sini *Admin* bisa memanipulasi data seperti mengubah, menambah, menghapus, dan melihat secara detail. Selain itu, bisa juga melihat histori pembayaran persiswa, melihat report pembayaran dan mendownloadnya.



Gambar 4. 4 *Activity Diagram Siswa*

4). *Activity Diagram Guru*

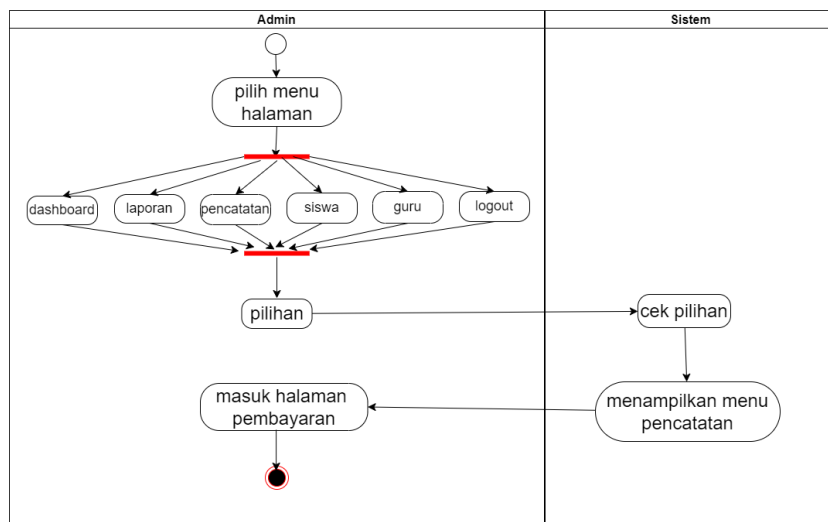
Proses ini diawali ketika berada di menu halaman, pengguna memilih menu guru. Sistem akan mengecek pilihan dan menampilkan menu tersebut. Di sini pengguna bisa memanipulasi data seperti menambah, mengubah, menghapus, dan melihat secara detail.



Gambar 4. 5 *Activity Diagram Guru*

5). *Activity Diagram Pencatatan*

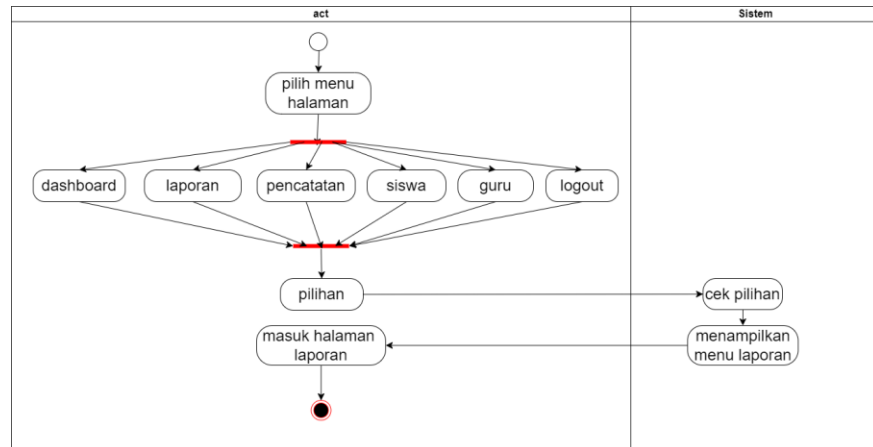
Proses ini diawali ketika berada di menu halaman, pengguna memilih menu pencatatan. Sistem akan mengecek pilihan dan menampilkan menu pencatatan. Di menu ini, dapat mencatat pemasukan semesteran, pemasukan lainnya dan jenis transaksi.



Gambar 4. 6 *Activity Diagram Pencatatan*

6). Activity Diagram Laporan

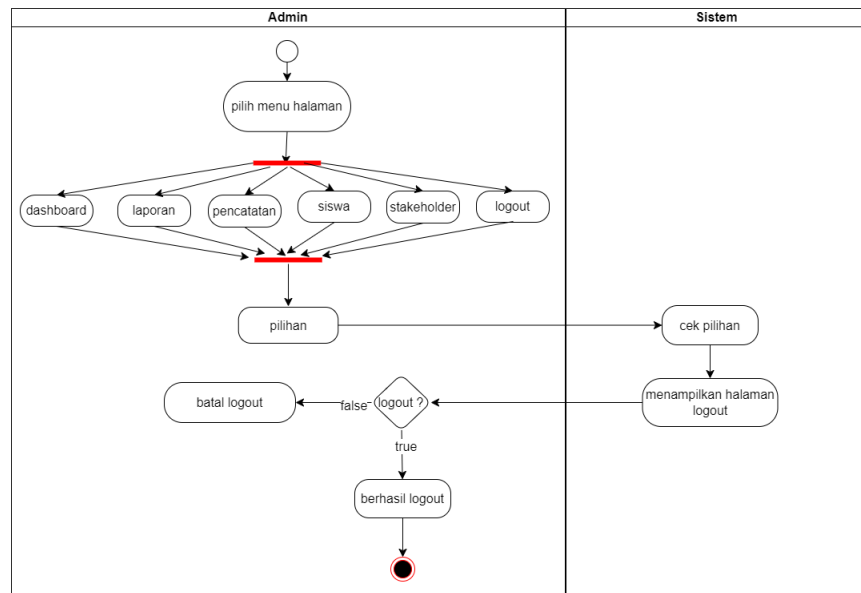
Proses ini diawali ketika berada di menu halaman, pengguna memilih menu laporan, kemudian sistem mengecek pilihan dan menampilkan menu laporan. Di menu ini, *Admin* bisa *mendownload* data, melihat secara detail terkait dengan laporan keuangan dan menambahkan pengeluaran.



Gambar 4. 7 Activity Diagram Laporan

7). Activity Diagram Logout

Proses ini diawali ketika pengguna berada di menu halaman, kemudian memilih menu *logout* sistem akan mengecek pilihan dan menampilkan pesan untuk meyakinkan apakah *Admin* ingin *logout*, jika benar maka berhasil *logout* jika tidak maka *logout* dibatalkan.



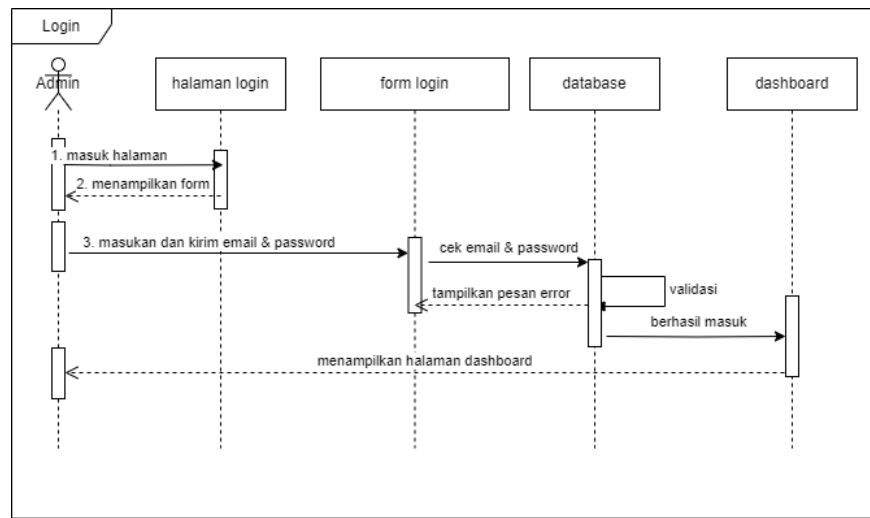
Gambar 4. 8 Activity Diagram Logout

c. *Sequence Diagram*

Berikut ini adalah *Sequence Diagram* pada sistem informasi yang akan dibangun.

1). *Sequence Diagram Login*

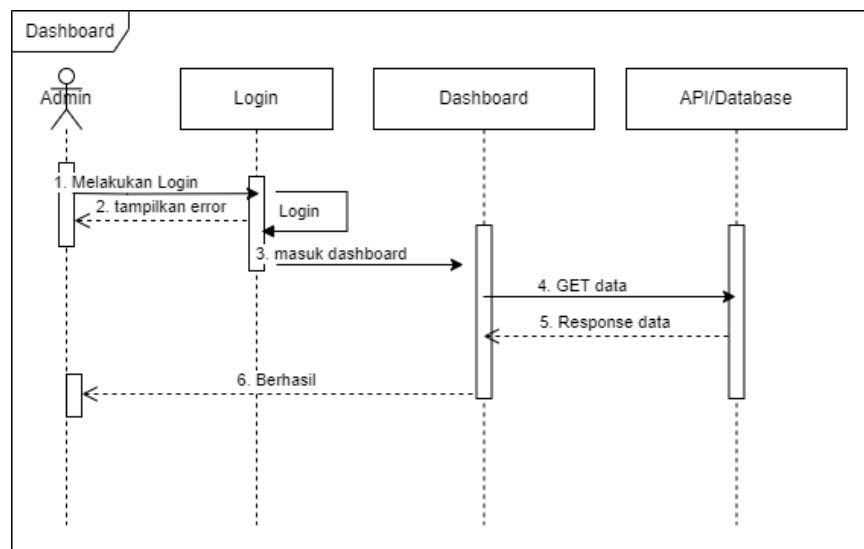
Di bawah ini merupakan proses interaksi antara *actor* dan beberapa objek ketika melakukan *login*. Objek ini antara lain halaman *login*, *form login*, *database* dan *dashboard*.



Gambar 4. 9 *Sequence Diagram Login*

2). *Sequence Diagram Dashboard*

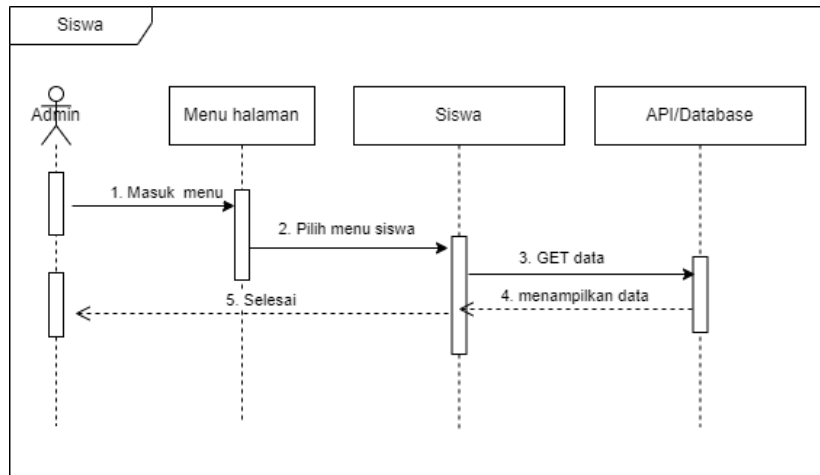
Gambar di bawah ini, merupakan proses masuk ke *dashboard* yang melibatkan interaksi antara aktor dan beberapa objek seperti *login*, *dashboard* dan *API/database*.



Gambar 4. 10 *Sequence Diagram Dashboard*

3). *Sequence Diagram Siswa*

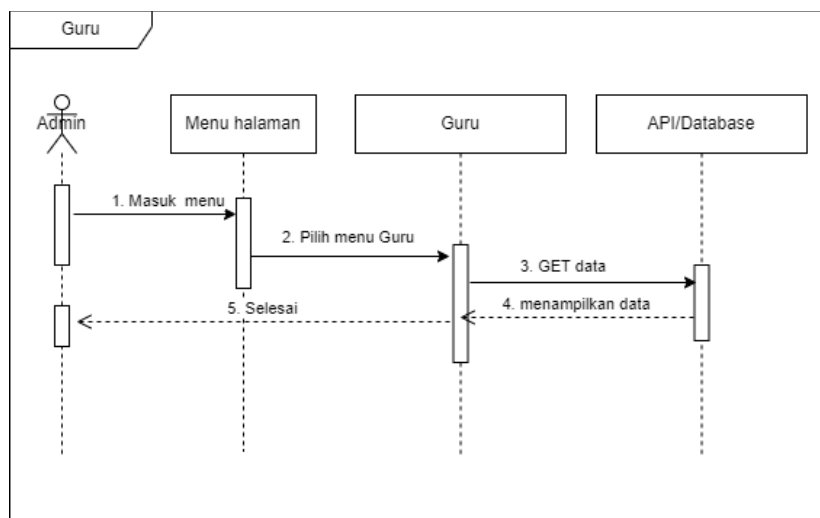
Gambar di bawah ini, merupakan proses masuk ke halaman siswa yang melibatkan interaksi antara aktor dan objek yaitu menu halaman, siswa dan *API/database*.



Gambar 4. 11 *Sequence Diagram Siswa*

4). *Sequence Diagram Guru*

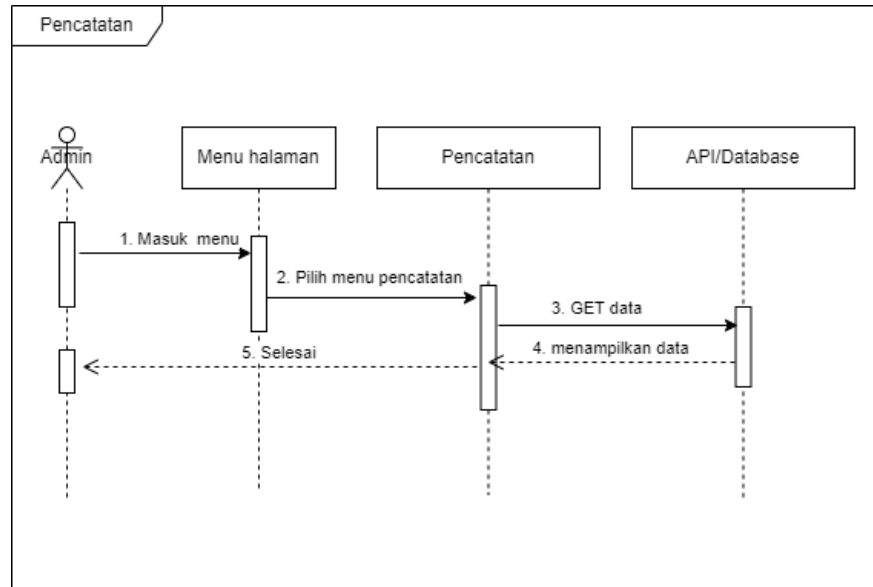
Gambar di bawah ini, merupakan proses masuk halaman dan mengelola data guru yang melibatkan interaksi antara aktor dan objek yaitu menu halaman, guru dan *API/database*.



Gambar 4. 12 *Sequence Diagram Guru*

5). *Sequence Diagram Pencatatan*

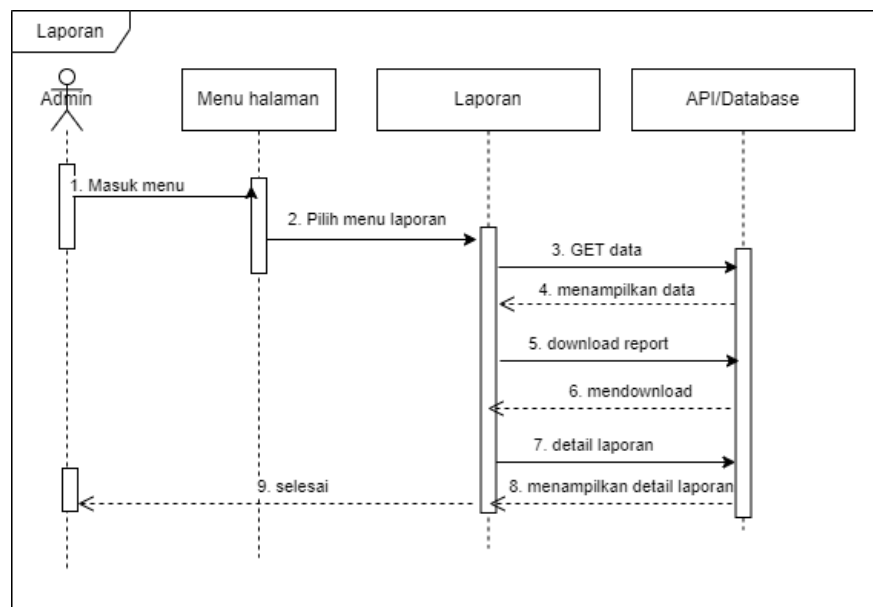
Gambar di bawah ini, merupakan proses masuk halaman pencatatan yang melibatkan interaksi antara aktor dan objek yaitu menu halaman, pencatatan dan *API/database*.



Gambar 4. 13 *Sequence Diagram* Pencatatan

6). *Sequence Diagram* Laporan

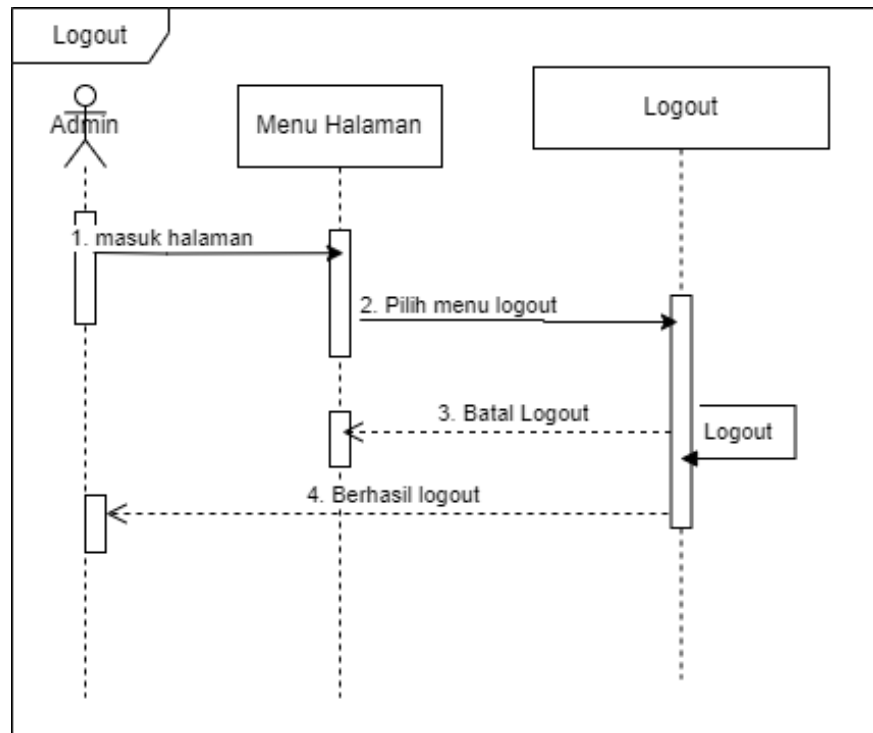
Gambar di bawah ini, merupakan proses masuk halaman laporan yang melibatkan interaksi antara aktor dan objek yaitu menu halaman, laporan dan *API/database*.



Gambar 4. 14 *Sequence Diagram* Laporan

7). *Sequence Diagram* Logout

Gambar di bawah ini, merupakan proses untuk melakukan *logout* yang melibatkan interaksi antara aktor dan objek yaitu menu halaman dan *logout*.



Gambar 4. 15 *Sequence Diagram Logout*

2. Desain Basis Data

Berikut desain basis data dari sistem yang akan dibangun di SMA Plus Nurul Iman Leles.

a. Desain Tabel

Adapun tabel yang terdapat pada sistem informasi ini, sebagai berikut.

1). Tabel *User*

Tabel *user* dipakai untuk menyimpan informasi pengguna seperti *email* dan *password*, ketika selesai melakukan registrasi.

Tabel 4. 5 *User*

No	Atribut	Tipe	Panjang/Ket
1.	id	int	pkey
2.	email	varchar	35
3.	password	varchar	15
4.	confirm_password	varchar	15
5.	created_at	date	-
6.	updated_at	date	-

2). Tabel Siswa

Tabel siswa dipakai untuk menyimpan data siswa saat proses *create* data dan ketika proses pembayaran siswa. Pada tabel ini *Atribut* nsn digunakan sebagai kunci utamanya. Selain itu, terdapat *Atribut* yang memiliki kunci asing seperti kelas_id, jurusan_id, agama_id dan gender_id, sebagai penanda adanya relasi tabel.

Tabel 4. 6 Siswa

No	Atribut	Tipe	Panjang/Ket
1.	nsn	varchar	11 (pkey)
2.	nama	varchar	35
3.	kelas_id	varchar	10 (fkey)
4.	jurusan_id	varchar	10 (fkey)
5.	agama_id	int	fkey
6.	tempat_lahir	varchar	15
7.	tanggal_lahir	date	-
8.	gender_id	int	fkey
9.	nama_ayah	varchar	35
10.	nama_ibu	varchar	35
11.	nomor_telepon	varchar	13
12.	angkatan	varchar	5
13.	email	varchar	35
14.	alamat	varchar	50
15.	gambar	varchar	255
16.	created_at	date	-
17.	updated_at	date	-

3). Tabel Guru

Tabel guru dipakai untuk menyimpan data guru saat proses *create* data dan ketika proses pembayaran kas. Pada tabel ini *Atribut* nip digunakan sebagai kunci utamanya. Selain itu, terdapat *Atribut* yang memiliki kunci asing seperti agama_id, jabatan_id, dan gender_id sebagai penanda adanya relasi tabel.

Tabel 4. 7 Guru

No	Atribut	Tipe	Panjang/Ket
1.	nip	varchar	17 (pkey)
2.	nama	varchar	35
3.	agama_id	int	fkey
4.	jabatan_id	int	fkey
5.	gender_id	Int	fkey
6.	tempat_lahir	varchar	15
7.	tanggal_lahir	date	-
8.	nomor_telepon	varchar	13
9.	email	varchar	35
10.	alamat	varchar	50
11.	gambar	varchar	255
12.	created_at	date	-
13.	updated_at	date	-

4). Tabel Kelas

Tabel ini dipakai untuk menyimpan data kelas saat proses *create* data. Tabel ini memiliki kode_kelas sebagai kunci utama untuk direlasikan ke tabel siswa.

Tabel 4. 8 Kelas

No	Atribut	Tipe	Panjang/Ket
1.	kode_kelas	varchar	10 (pkey)
2.	kelas	varchar	5
3.	created_at	date	-
4.	updated_at	date	-

5). Tabel Jurusan

Tabel jurusan dipakai untuk menyimpan data jurusan saat proses *create* data. Tabel ini memiliki kode_jurusan sebagai kunci utama untuk direlasikan ke tabel siswa.

Tabel 4. 9 Jurusan

No	Atribut	Tipe	Panjang/Ket
1.	kode_jurusan	varchar	10 (pkey)
2.	jurusan	varchar	25
3.	created_at	date	-
4.	updated_at	date	-

6). Tabel Jabatan

Tabel jabatan dipakai untuk menyimpan data jabatan saat proses *create* data. Tabel ini memiliki id_jabatan sebagai kunci utama (*primary key*) untuk direlasikan ke tabel guru.

Tabel 4. 10 Jabatan

No	Atribut	Tipe	Panjang/Ket
1.	id_jabatan	int	pkey
2.	nama	varchar	30

7). Tabel Agama

Tabel ini dipakai untuk menyimpan data agama saat proses *create* data. Tabel ini memiliki id_agama sebagai kunci utama (*primary key*) untuk direlasikan ke tabel siswa dan guru.

Tabel 4. 11 Agama

No	Atribut	Tipe	Panjang/Ket
1.	id_agama	int	pkey
2.	nama	varchar	18

8). Tabel Gender

Tabel *gender* adalah tabel yang digunakan untuk menyimpan data *gender*, saat proses *create* data melalui *database* secara langsung. Tabel ini memiliki id_gender sebagai kunci utama (*primary key*) untuk direlasikan ke tabel siswa dan guru.

Tabel 4. 12 Gender

No	Atribut	Tipe	Panjang/Ket
1.	id_gender	int	pkey
2.	nama	varchar	10

9). Tabel Transaksi

Tabel ini dipakai untuk menyimpan data transaksi saat proses *create* data. Tabel ini memiliki id sebagai kunci utama (*primary key*) untuk direlasikan ke tabel pembayaran.

Tabel 4. 13 Transaksi

No	Atribut	Tipe	Panjang/Ket
1.	id	int	-
2.	nama	varchar	35
3.	jumlah	int	-
4.	kategori	varchar	1
5.	created_at	date	-
6.	updated_at	date	-

10). Tabel Pembayaran

Tabel pembayaran dipakai untuk menyimpan data pembayaran saat proses *create* data. Tabel ini memiliki id sebagai kunci utama dan beberapa kunci tamu yang berelasi ke tabel siswa dan transaksi.

Tabel 4. 14 Pembayaran

No	Atribut	Tipe	Panjang/Ket
1.	id	int	pkey
2.	siswa_id	varchar	11 (fkey)
3.	transaksi_id	int	fkey
4.	bulan	varchar	9
5.	semester	varchar	7
6.	tanggal	date	-

No	Atribut	Tipe	Panjang/Ket
7.	jumlah	int	-
8.	status	varchar	12
9.	created_at	date	-
10.	updated_at	date	-

11). Tabel *Histori* Pembayaran

Tabel histori ini dipakai pengguna untuk menyimpan data pembayaran saat proses *create* data, dimana data dalam tabel ini akan dijadikan sebagai histori pembayaran siswa berdasarkan NISN tertentu.

Tabel 4. 15 Histori Pembayaran

No	Atribut	Tipe	Panjang/Ket
1.	id	int	-
2.	siswa	varchar	35
3.	nisn	varchar	11
4.	nama_transaksi	varchar	35
5.	tanggal	varchar	11
6.	jumlah	int	-
7.	status	varchar	12
8.	created_at	date	-
9.	updated_at	date	-

12). Tabel Pemasukan

Tabel ini dipakai pengguna untuk menyimpan data pemasukan saat proses *create* data dan datanya akan digunakan sebagai laporan pemasukan.

Tabel 4. 16 Pemasukan

No	Atribut	Tipe	Panjang/Ket
1.	id	int	pkey

No	Atribut	Tipe	Panjang/Ket
2.	nama	varchar	35
3.	jumlah	int	-
4.	tanggal	date	-
5.	created_at	date	-
6.	updated_at	date	-

13). Tabel Pengeluaran

Tabel pengeluaran digunakan untuk menyimpan data pengeluaran saat proses *create* data dan datanya akan digunakan sebagai laporan pengeluaran.

Tabel 4. 17 Pengeluaran

No	Atribut	Tipe	Panjang/Ket
1.	id	int	pkey
2.	nama	varchar	35
3.	jumlah	int	-
4.	tanggal	date	-
5.	created_at	date	-
6.	updated_at	date	-

14). Tabel *Session*

Tabel *session* digunakan untuk menyimpan data *session* saat proses *create* data. *Session* digunakan untuk menyimpan informasi seperti token dan *email* pengguna disertai waktu kadaluarsanya.

Tabel 4. 18 *Session*

No	Atribut	Tipe	Panjang/Ket
1.	id	int	pkey
2.	token	varchar	255
3.	email	varchar	35
4.	expiry	date	-

15). Tabel Kas Guru

Tabel kas guru dipakai untuk menyimpan data kas guru ketika proses *create* data. Data pada tabel ini akan menyimpan setiap pembayaran kas yang dilakukan oleh guru dengan NIP tertentu. Tabel ini memiliki id sebagai kunci utama (*primary key*) dan guru_id sebagai kunci tamu (*primary key*) yang berelasi ke tabel guru.

Tabel 4. 19 Kas Guru

No	Atribut	Tipe	Panjang/Ket
1.	id	int	pkey
2.	guru_id	varchar	17 (fkey)
3.	jumlah	int	-
4.	saldo	int	-
5.	tanggal	date	-
6.	created_at	date	-
7.	updated_at	date	-

16). Tabel Pengambilan Kas

Tabel pengambilan kas dipakai untuk menyimpan data ketika guru melakukan pengambilan kas. Data yaang ada pada tabel ini akan menyimpan setiap histori pengambilan kas yang dilakukan oleh guru dengan NIP tertentu. Tabel ini memiliki id sebagai kunci utama (*primary key*) dan guru_id (*foreign key*) sebagai kunci tamu yang berelasi ke tabel guru.

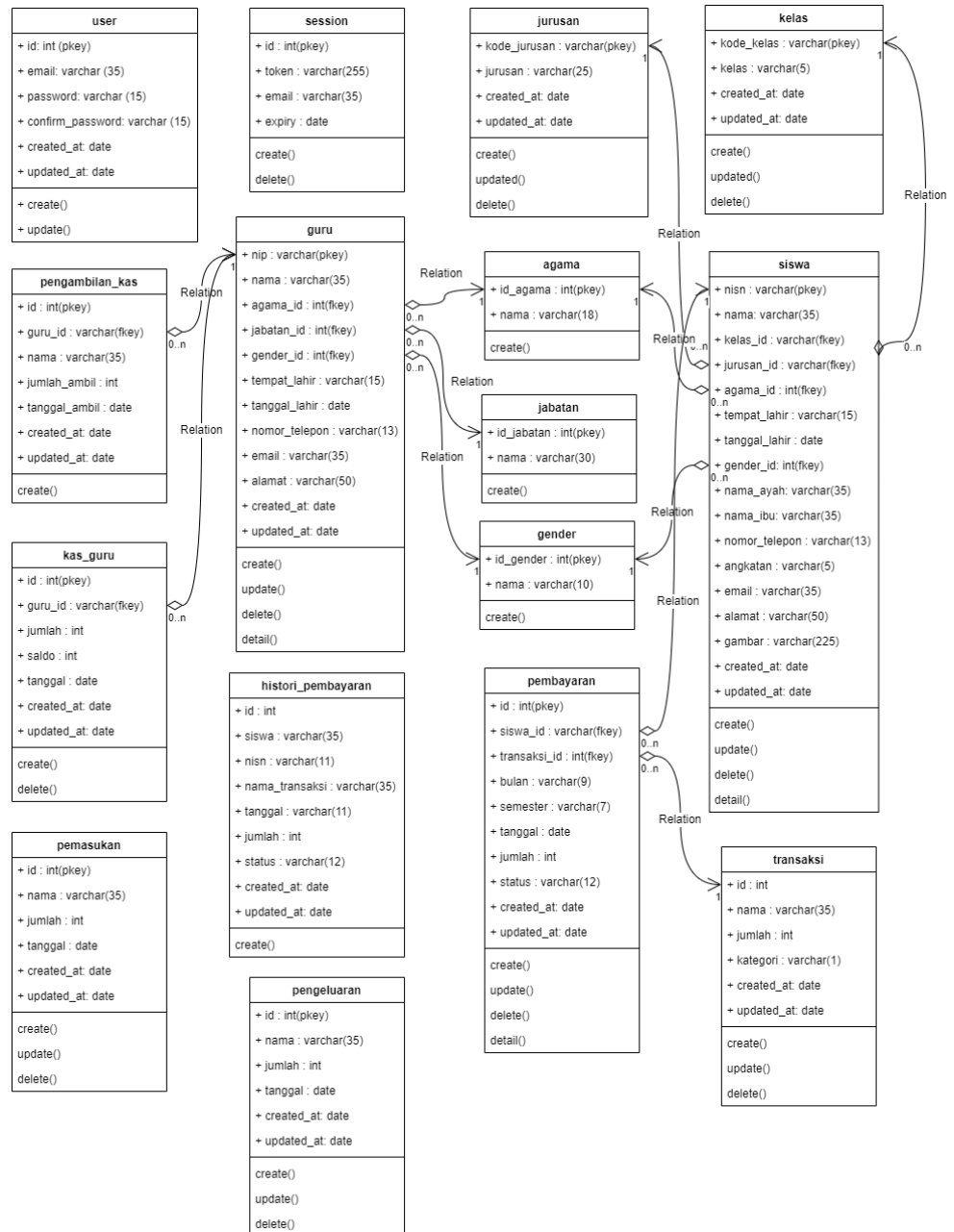
Tabel 4. 20 Pengambilan Kas

No	Atribut	Tipe	Panjang/Ket
1.	id	int	pkey
2.	guru_id	varchar	17 (fkey)
3.	nama	varchar	35
4.	jumlah_ambil	int	-
5.	tanggal_ambil	date	-
6.	created_at	date	-

No	Atribut	Tipe	Panjang/Ket
7.	updated_at	date	-

b. *Class Diagram*

Di bawah ini merupakan diagram kelas dari sistem informasi administrasi keuangan SMA Plus Nurul Iman Leles.



Gambar 4. 16 Class Diagram

Class Diagram di atas terdiri dari 16 *class*, masing-masing dengan atribut yang berbeda. Setiap *class* memiliki atribut-atribut yang diatur

sesuai dengan *class* yang telah disusun. Beberapa *class* memiliki hubungan atau relasi dengan *class* lain. Relasi ini menunjukkan bagaimana setiap data dalam satu *class* berhubungan dengan data lain dalam *class* lainnya, memungkinkan integrasi dan keterkaitan data yang lebih baik di seluruh diagram.

Berikut ini penjelasan dari *class* yang memiliki relasi dengan *class* yang lain.

1). *Class* siswa dan *class* kelas

Kedua *class* ini memiliki relasi dengan kardinalitas *one-to-many*, yang berarti setiap *class* siswa hanya boleh memiliki satu *class* kelas, namun setiap *class* kelas boleh memiliki banyak *class* siswa.

2). *Class* siswa dan *class* jurusan

Kedua entitas ini memiliki relasi dengan kardinalitas *one-to-many*, yang berarti setiap *class* siswa hanya dapat memiliki satu *class* jurusan, namun setiap jurusan boleh memiliki banyak *class* siswa.

3). *Class* siswa dan entitas agama

Kedua *class* ini memiliki relasi dengan kardinalitas *one-to-many*, yang berarti setiap *class* siswa hanya dapat memiliki satu *class* agama, namun setiap agama boleh memiliki banyak *class* siswa.

4). *Class* siswa dan *class* gender

Kedua *class* ini memiliki relasi dengan kardinalitas *one-to-many*, yang berarti setiap *class* siswa hanya dapat memiliki satu *class* gender, namun setiap *class* gender boleh memiliki banyak *class* siswa.

5). *Class* pembayaran dan *class* siswa

Kedua *class* ini memiliki relasi dengan kardinalitas *one-to-many*, yang berarti setiap *class* pembayaran akan memiliki satu *class* siswa, namun setiap siswa dapat terkait dengan banyak *class* pembayaran.

6). *Class* pembayaran dan *class* transaksi

Kedua *class* ini memiliki relasi dengan kardinalitas *one-to-many*, ini karena setiap *class* pembayaran akan memiliki satu *class* transaksi,

tetapi satu *class* transaksi dapat terkait dengan banyak *class* pembayaran.

7). *Class* guru dan *class* agama

Kedua *class* ini memiliki relasi dengan kardinalitas *one-to-many*, yang berarti setiap *class* guru hanya dapat memiliki satu *class* agama, namun setiap *class* agama boleh memiliki banyak *class* guru.

8). *Class* guru dan *class* gender

Kedua *class* ini memiliki relasi dengan kardinalitas *one-to-many*, yang berarti setiap *class* guru hanya dapat memiliki satu *gender*, namun setiap *gender* boleh memiliki banyak *class* guru.

9). *Class* guru dan *class* jabatan

Kedua *class* ini memiliki relasi dengan kardinalitas *one-to-many*, yang berarti setiap *class* guru hanya dapat memiliki satu jabatan, namun setiap jabatan boleh memiliki banyak *class* guru.

10). *Class* kas guru dan *class* guru

Kedua *class* ini memiliki relasi dengan kardinalitas *one-to-many*, yang berarti setiap *class* kas guru hanya dapat memiliki satu *class* guru, namun setiap *class* guru boleh memiliki banyak *class* kas guru.

11). *Class* pengambilan kas dan *class* guru

Kedua *class* ini memiliki relasi dengan kardinalitas *one-to-many*, artinya setiap *class* pengambilan kas hanya boleh memiliki satu *class* guru, namun setiap entitas guru boleh memiliki banyak *class* pengambilan kas.

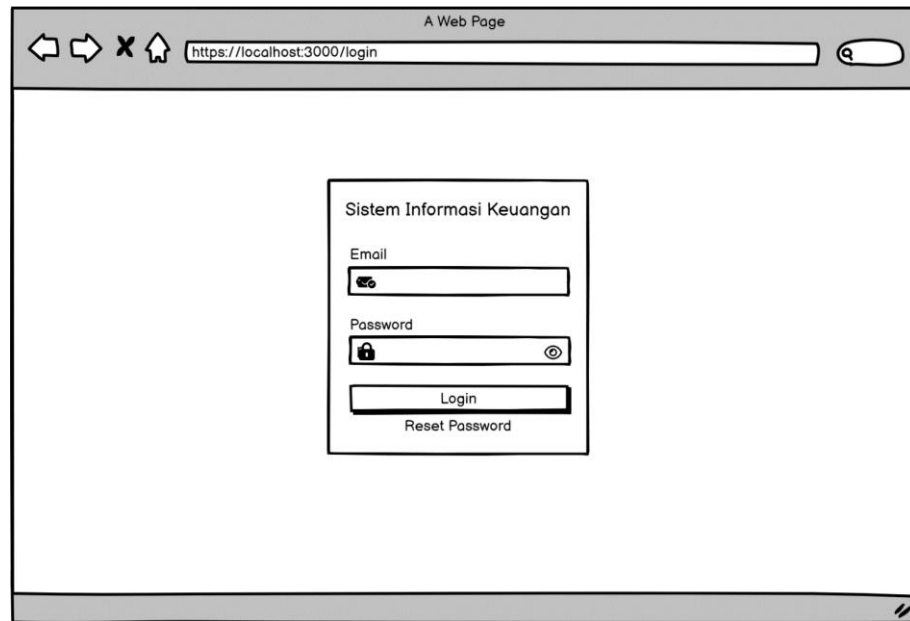
Secara keseluruhan, *class diagram* ini memberikan gambaran yang jelas dan terstruktur tentang bagaimana *class-class* dalam sistem saling berhubungan dan bagaimana data mengalir di antara mereka. Ini penting untuk memastikan *desain database* yang efisien dan efektif.

3. Desain Antar Muka

Desain antarmuka yang dibuat untuk sistem administrasi keuangan SMA Plus Nurul Iman Leles berupa *wireframe* dengan kombinasi warna dasar hitam dan putih. Adapun desain antarmukanya bisa dilihat pada gambar berikut.

a. Halaman *Login*

Halaman ini dipakai pengguna untuk masuk ke sistem (halaman utama/*Dashboard*), dimana pengguna harus memasukan email dan *password* kemudian sistem akan memvalidasinya.

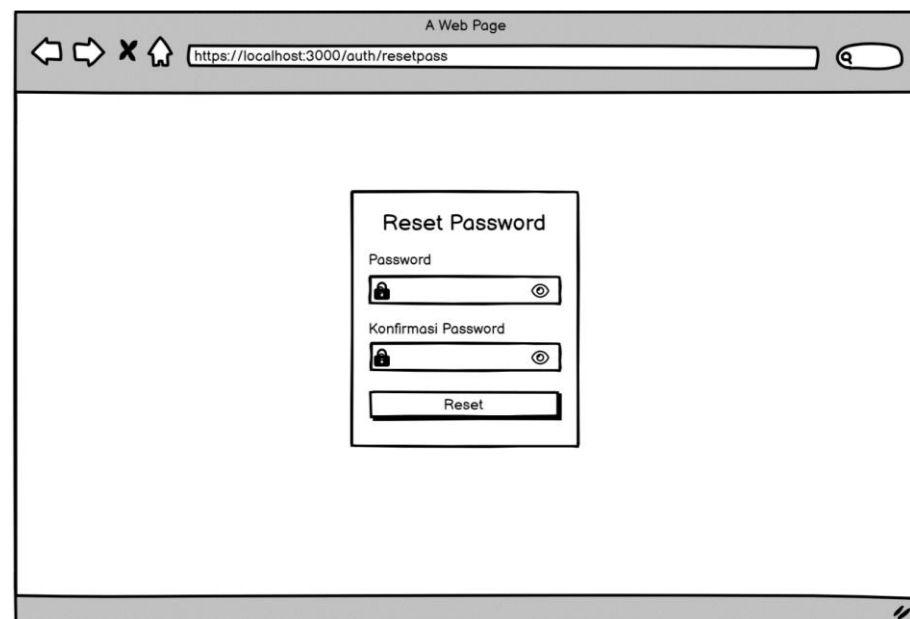


The screenshot shows a web browser window titled "A Web Page" with the address bar displaying "https://localhost:3000/login". The main content area contains a login form titled "Sistem Informasi Keuangan". The form has two input fields: "Email" with an envelope icon and "Password" with a lock icon and a toggle eye icon. Below the password field are two buttons: "Login" and "Reset Password".

Gambar 4. 17 Desain Tampilan Halaman *Login*

b. Halaman *Reset Password*

Halaman ini dipakai pengguna untuk mereset *password*. Admin harus mengisi *form password* dan konfirmasi *password* yang baru.

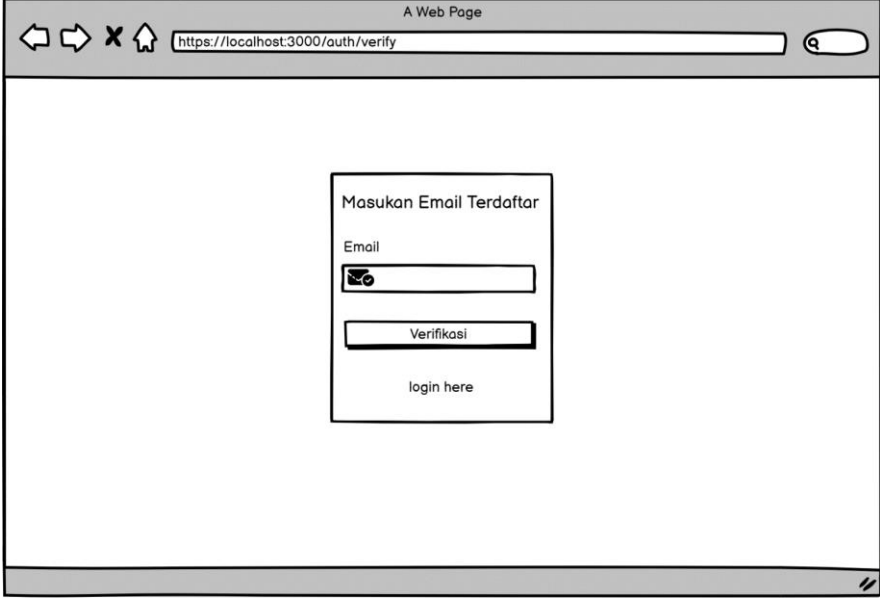


The screenshot shows a web browser window titled "A Web Page" with the address bar displaying "https://localhost:3000/auth/resetpass". The main content area contains a reset password form titled "Reset Password". The form has two input fields: "Password" and "Konfirmasi Password", both with lock icons and toggle eye icons. Below these fields is a "Reset" button.

Gambar 4. 18 Desain Tampilan Halaman *Reset Password*

c. Halaman *Verifikasi Email*

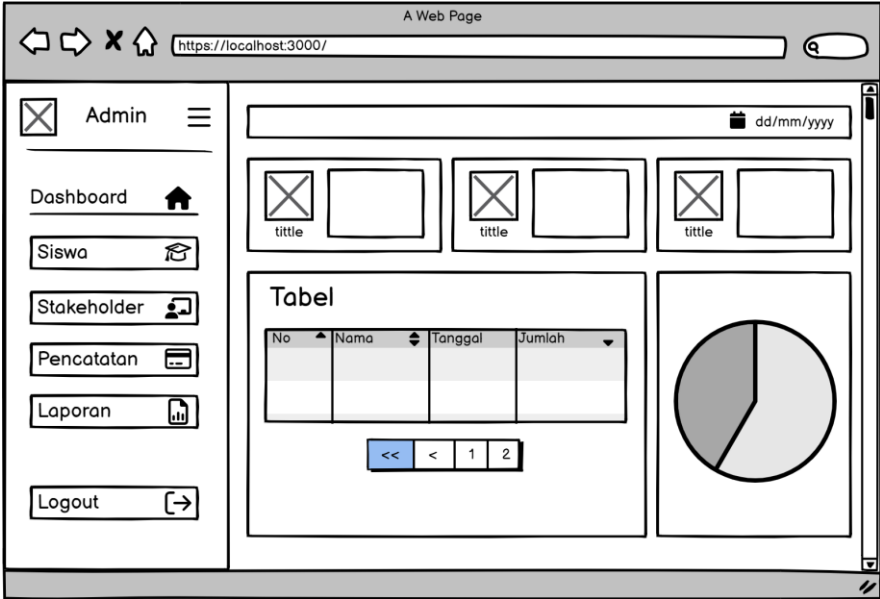
Halaman ini digunakan untuk memverifikasi email apabila ingin mereset *password*. Jika email tersedia, maka sistem akan mengirimkan link verifikasi ke email tersebut kemudian *Admin* bisa mengklik dan akan diarahkan ke halaman berikutnya.



Gambar 4. 19 Desain Tampilan Halaman Verifikasi *Email*

d. *Dashboard*

Halaman ini digunakan untuk mereset *password*. Admin harus mengisi *form password* baru dan konfirmasi *password*.

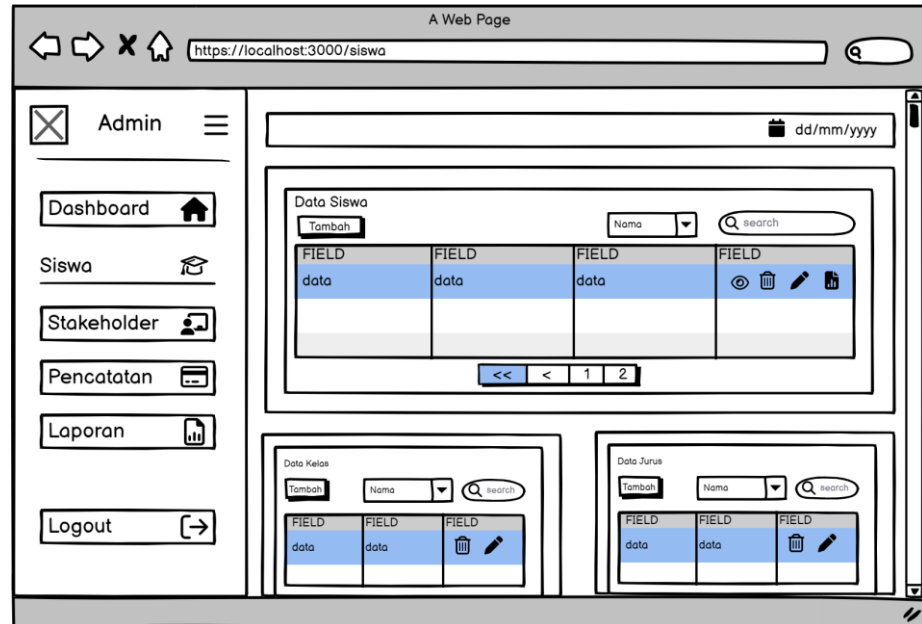


No	Nama	Tanggal	Jumlah

Gambar 4. 20 Desain Tampilan *Dashboard*

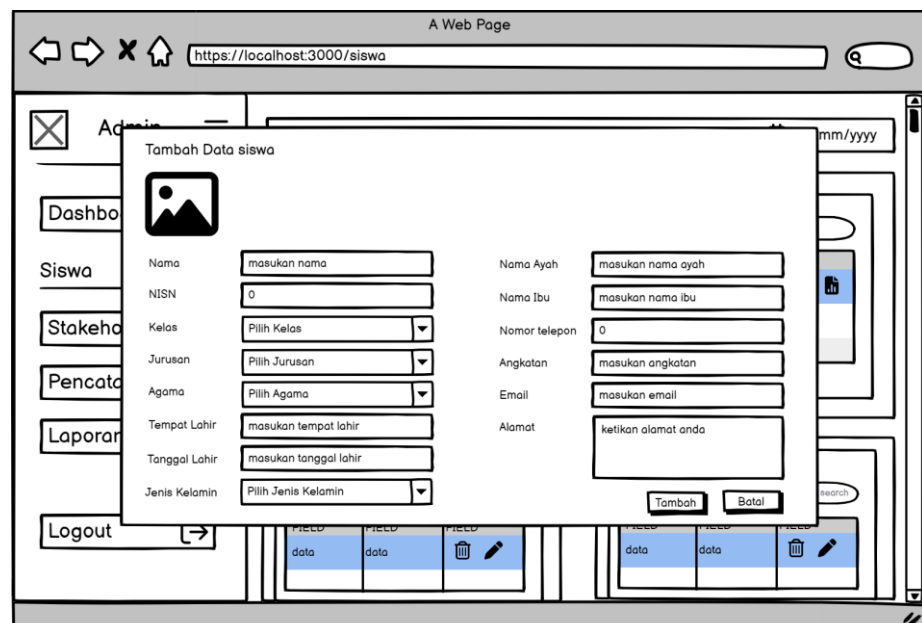
e. Halaman Siswa

Halaman ini menampilkan data berbentuk tabel data siswa, data kelas, dan data jurusan disertai dengan fitur *create*, *read*, *update*, *delete*, *view history* pembayaran, *view report* pembayaran, dan *download*.



Gambar 4. 21 Desain Tampilan Halaman siswa

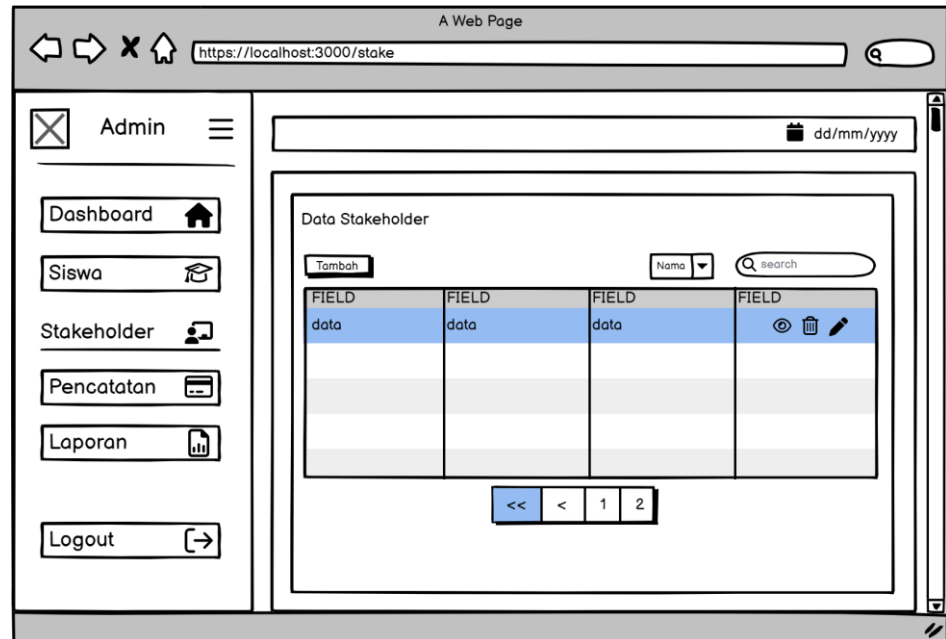
Di bawah ini merupakan tampilan form tambah data siswa. Form yang ada pada fitur ini dilengkapi dengan pengisian atribut-atribut yang dibutuhkan untuk data siswa, seperti nama, nim, dan lain-lain.



Gambar 4. 22 Desain Tampilan Form Tambah Siswa

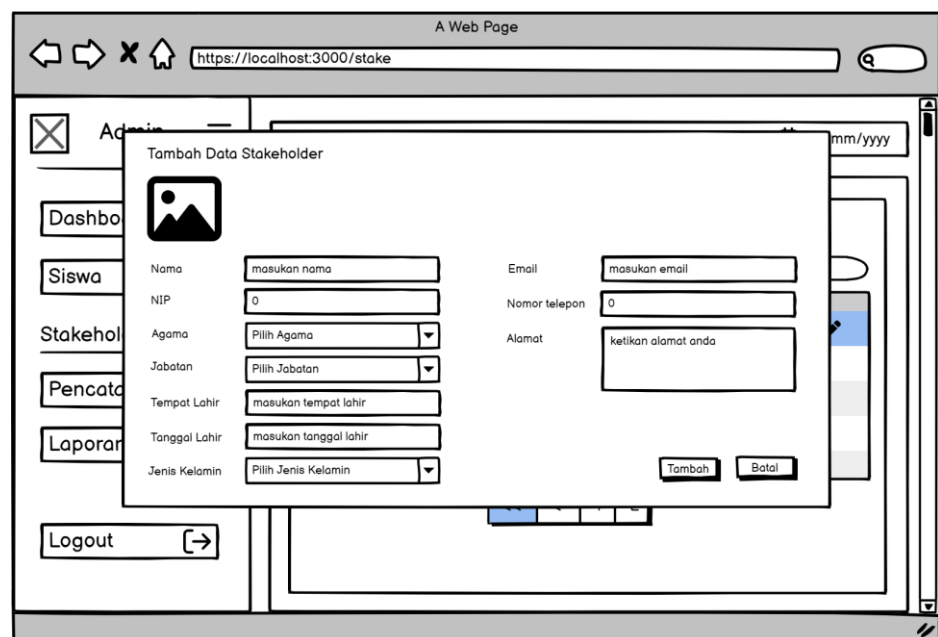
f. Halaman Guru

Halaman ini menampilkan data berbentuk tabel data guru disertai dengan fitur *create*, *read*, *update*, *delete*, dan *download*.



Gambar 4. 23 Desain Tampilan Halaman Guru

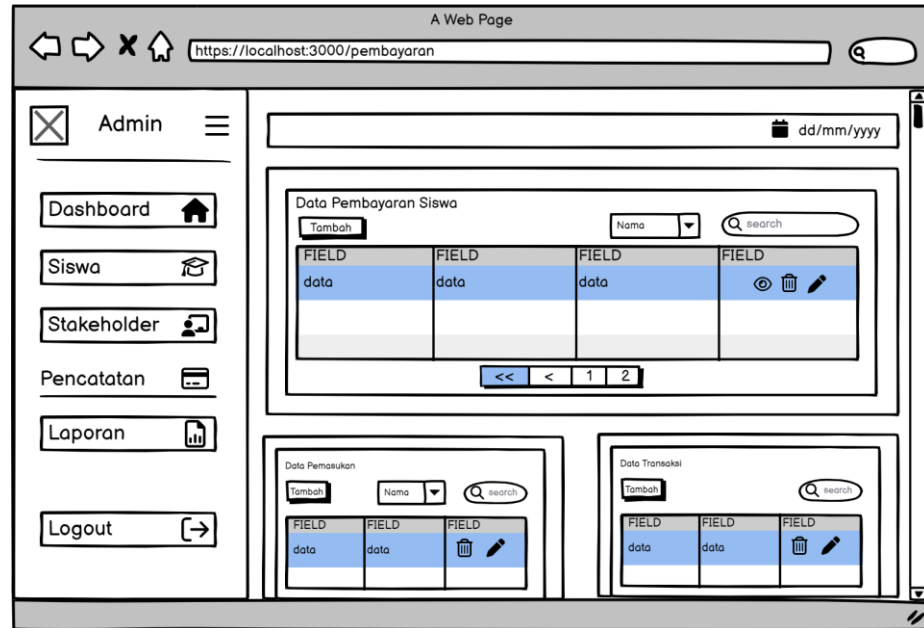
Di bawah ini merupakan tampilan form tambah data guru. Form yang ada pada fitur ini dilengkapi dengan pengisian atribut-atribut yang dibutuhkan untuk data guru, seperti nama, nim, dan lain-lain.



Gambar 4. 24 Desain Tampilan Form Tambah Guru

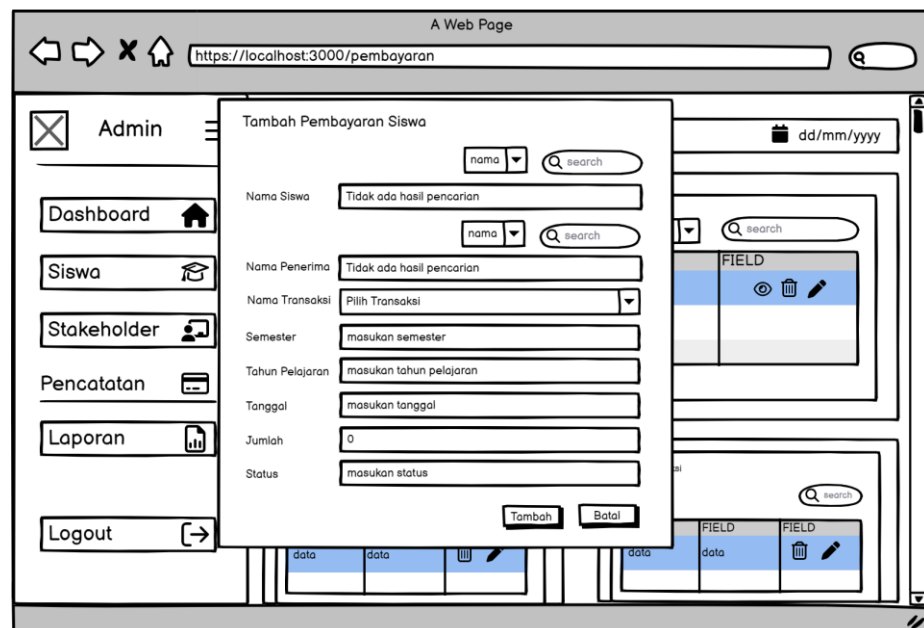
g. Halaman Pencatatan

Halaman ini menampilkan data berbentuk tabel data pembayaran siswa, data pemasukan dan data jenis transaksi disertai dengan fitur *create*, *read*, *update* dan *delete*.



Gambar 4. 25 Desain Tampilan Halaman Pencatatan

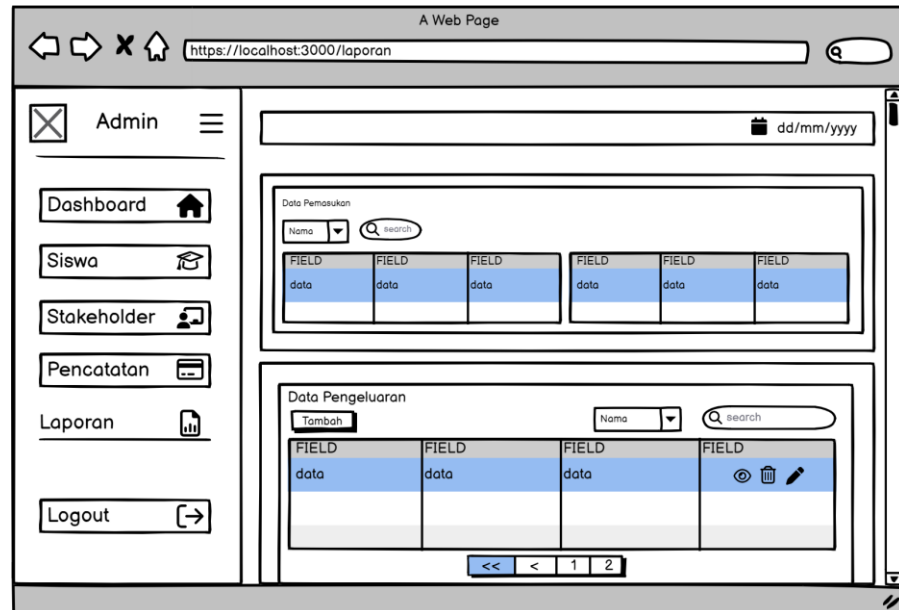
Di bawah ini merupakan tampilan form tambah data pembayaran siswa. Form yang ada pada fitur ini dilengkapi dengan pengisian atribut-atribut yang dibutuhkan untuk data pembayaran siswa.



Gambar 4. 26 Desain Tampilan Form Tambah Pembayaran Siswa

h. Halaman Laporan

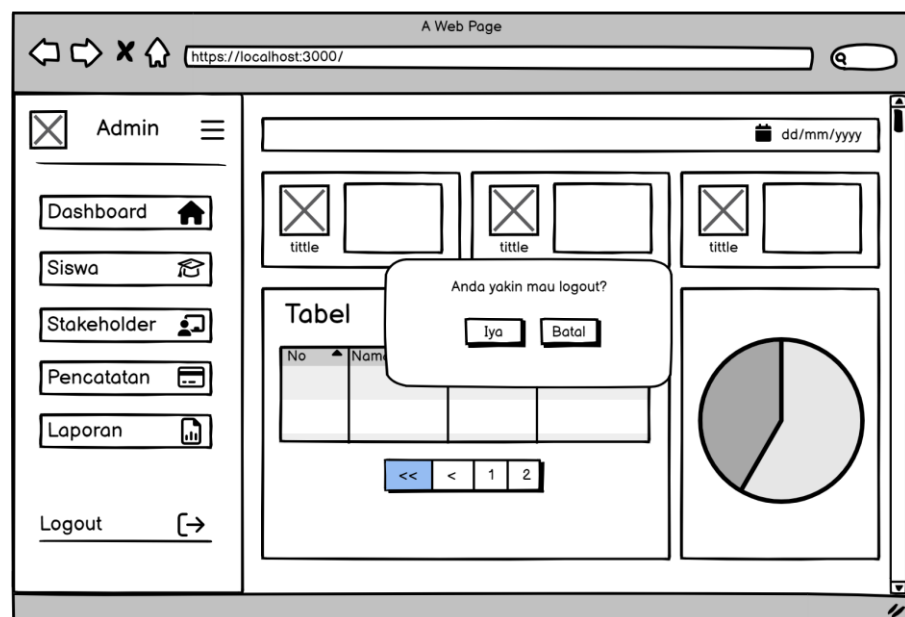
Halaman ini menampilkan data berbentuk tabel pemasukan dan pengeluaran disertai dengan fitur *create*, *read*, *update* dan *delete*.



Gambar 4. 27 Desain Tampilan Halaman Laporan

i. Logout

Tampilan ini adalah tampilan saat pengguna akan melakukan *logout*. *User* bisa mengklik tombol (iya) untuk mengkonfirmasi *logout* atau mengklik tombol (batal) untuk membatalkan *logout*.



Gambar 4. 28 Desain Tampilan Logout

4.5.2. Desain Kode Program

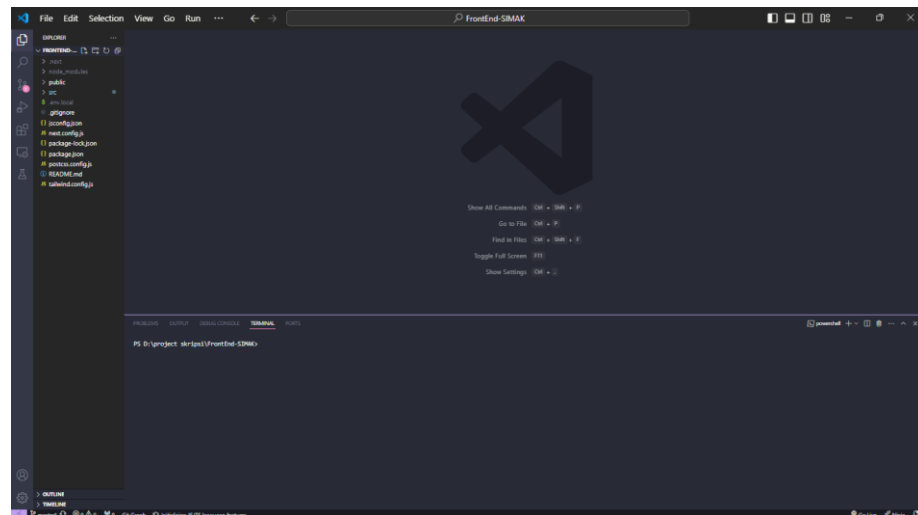
Desain program dari sistem informasi administrasi keuangan SMA Plus Nurul Iman Leles yaitu sebagai berikut.

1. *Front End*

Kode program ini dirancang untuk membuat tampilan atau *interface* dari aplikasi.

a. Struktur Folder

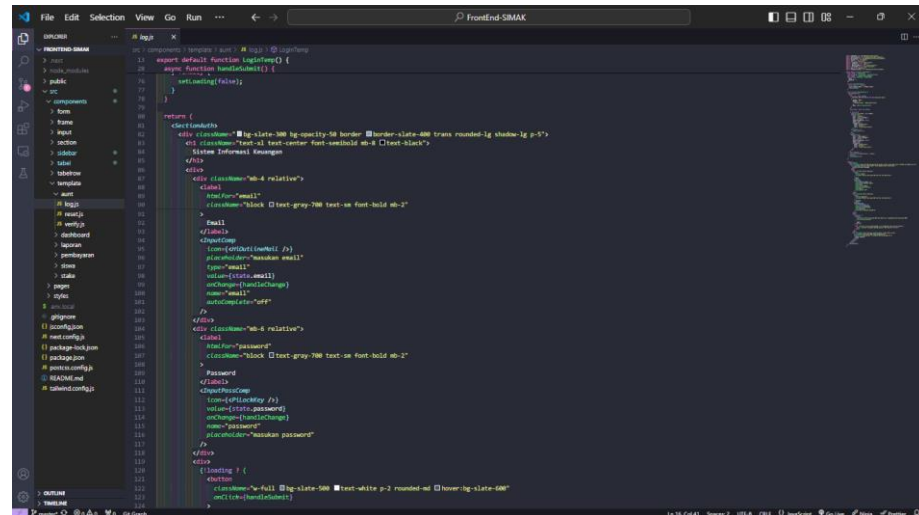
Struktur folder digunakan untuk menyusun penulisan kode program supaya lebih tertata rapi dan mudah difahami. Adapun struktur folder pada aplikasi dibuat sesuai dengan standar *framework NextJS*, yang dimana terdapat folder *next* sebagai hasil proses rendering, *node_modules* sebagai folder informasi dari package yang digunakan, *public* untuk menyimpan berbagai *asset*, *source* (src) untuk menyimpan sebagian besar kode program sumber proyek termasuk halaman, tata letak atau komponen lainnya.



Gambar 4. 29 Struktur Folder *Front End*

b. Kode Program *Login*

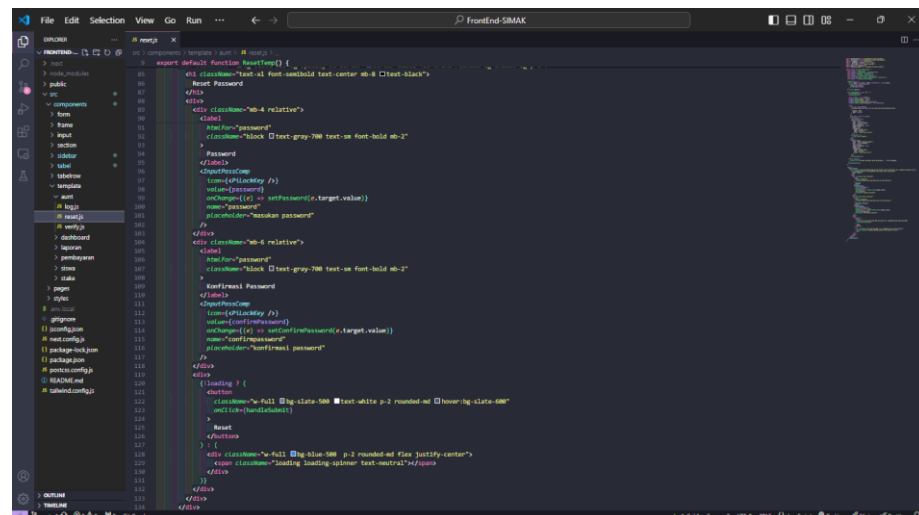
Kode program ini digunakan untuk membuat tampilan *login* di sisi *client*, dimana terdapat *component* seperti *form input email*, *form input password*, dan *button* serta fungsi untuk mengirim *request API* ke *server*, melalui alamat URL yang telah dimasukan ke dalam permintaan di dalam kode program.



Gambar 4. 30 Kode Program *Front End Login*

c. Kode Program *Reset Password*

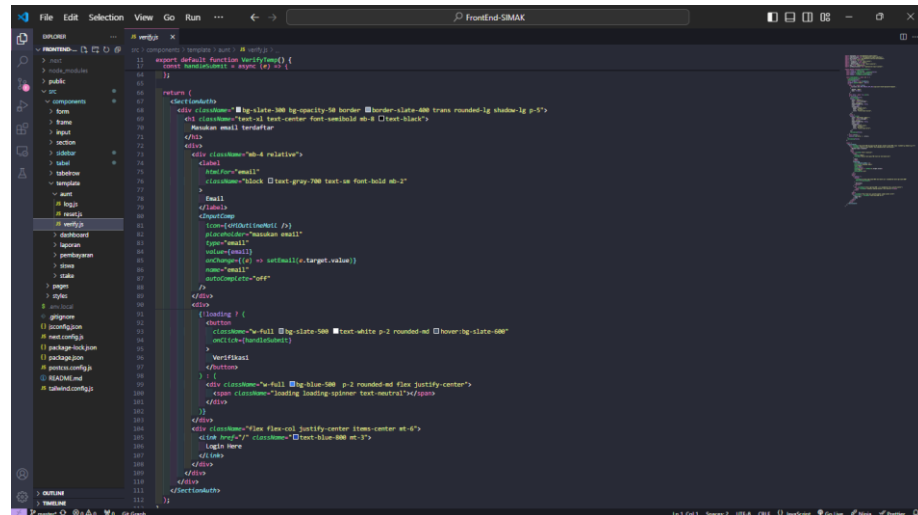
Kode program ini digunakan untuk membuat tampilan *Reset Password* di sisi *client*, dimana terdapat *component* seperti *form input password*, *form input konfirmasi password* dan *button* serta fungsi untuk mengirim *request API* ke *server*, melalui alamat URL yang telah dimasukan ke dalam permintaan di dalam kode program.



Gambar 4. 31 Kode Program *Front End Reset Password*

d. Kode Program *Verifikasi Password*

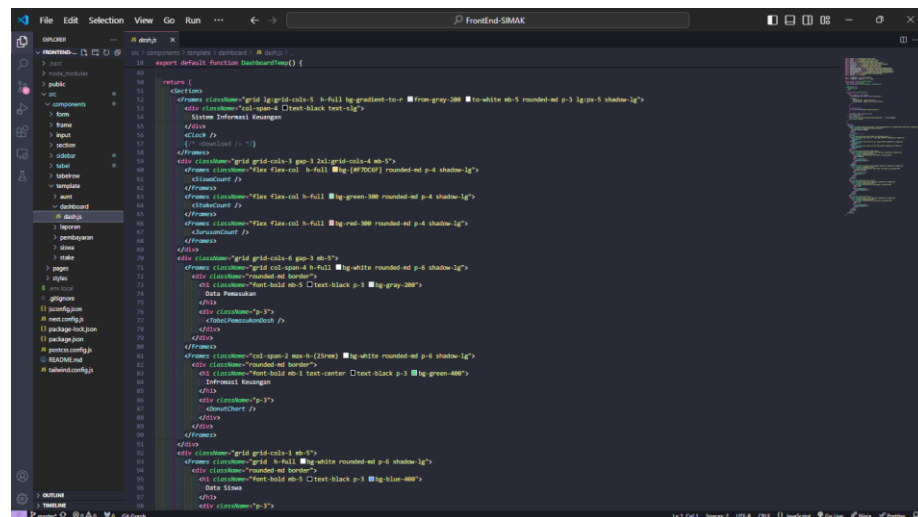
Kode program ini digunakan untuk membuat tampilan verifikasi di sisi *client*, dimana terdapat *component* seperti *form input* dan *button* serta fungsi untuk mengirim *request API* ke *server*, melalui alamat URL yang telah dimasukan ke dalam permintaan di dalam kode program.



Gambar 4. 32 Kode Program *Front End Verifikasi Password*

e. *Dashboard*

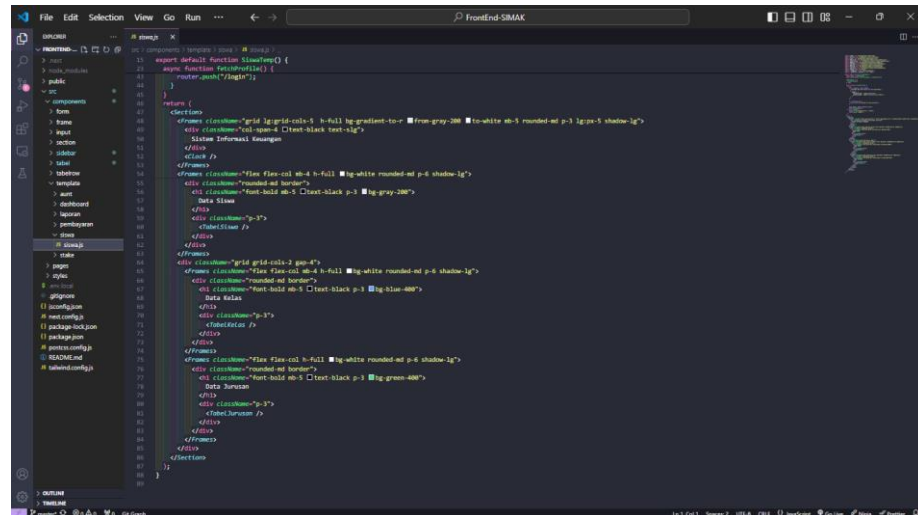
Kode program ini digunakan untuk membuat tampilan *dashboard*, dimana terdapat *component* seperti tabel dan diagram *chart* yang diambil dari *library chart JS*, serta fungsi untuk mengirim *request* API ke *server*, melalui alamat URL yang telah dimasukan ke dalam permintaan di dalam kode program.



Gambar 4. 33 Kode Program *Front End Dashboard*

f. *Kode Program Siswa*

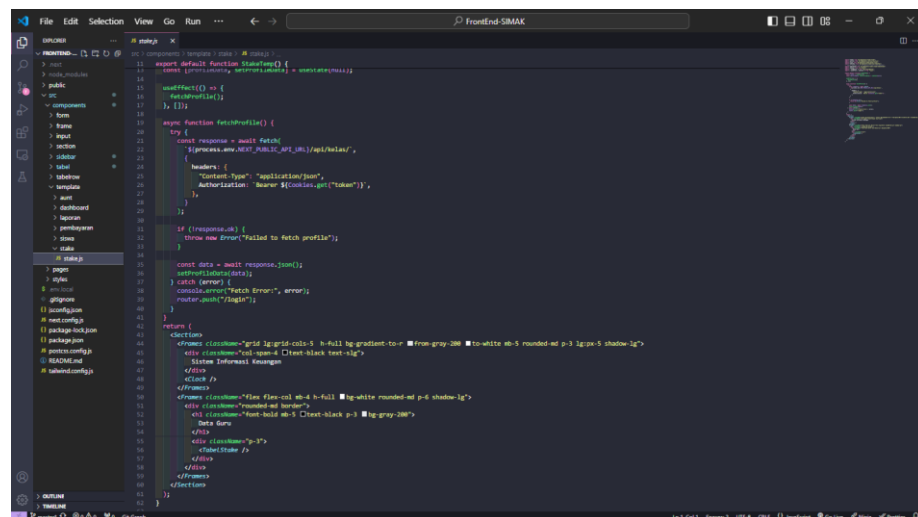
Kode program ini digunakan untuk membuat tampilan halaman siswa dan fitur-fitur yang ada di dalamnya. Program ini memiliki fungsi untuk mengirim *request* API ke *server*, melalui alamat URL yang telah dimasukan ke dalam permintaan di dalam kode program.



Gambar 4. 34 Kode Program *Front End* Siswa

g. Kode Program Guru

Kode program ini digunakan untuk membuat tampilan halaman guru dan fitur-fitur yang ada di dalamnya. Program ini memiliki fungsi untuk mengirim *request* API ke *server*, melalui alamat URL yang telah dimasukan ke dalam permintaan di dalam kode program.



Gambar 4. 35 Kode Program *Front End* Guru

h. Kode Program Laporan

Kode program ini digunakan untuk membuat tampilan halaman laporan dan fitur-fitur yang ada di dalamnya seperti cetak laporan, kwitansi dan sebagainya. Program ini memiliki fungsi untuk mengirim *request* API ke *server*, melalui alamat URL yang telah dimasukan ke dalam permintaan di dalam kode program.

i. Kode Program Pencatatan

[illegible]

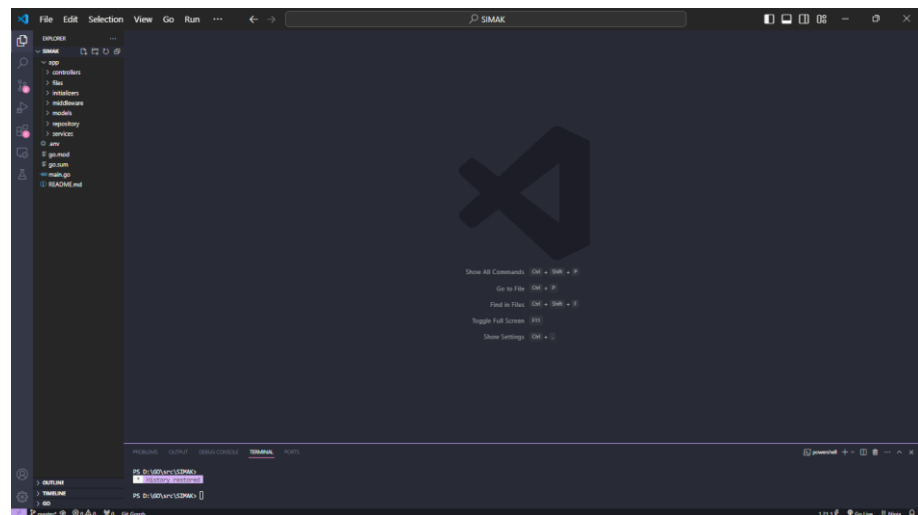
j. Kode Program *Logout*

Kode program ini digunakan untuk membuat tampilan ketika mau melakukan *logout*. Program ini memiliki fungsi untuk mengirim *request* API ke *server*, melalui alamat URL yang telah dimasukan ke dalam permintaan di dalam kode program.

2. Back End

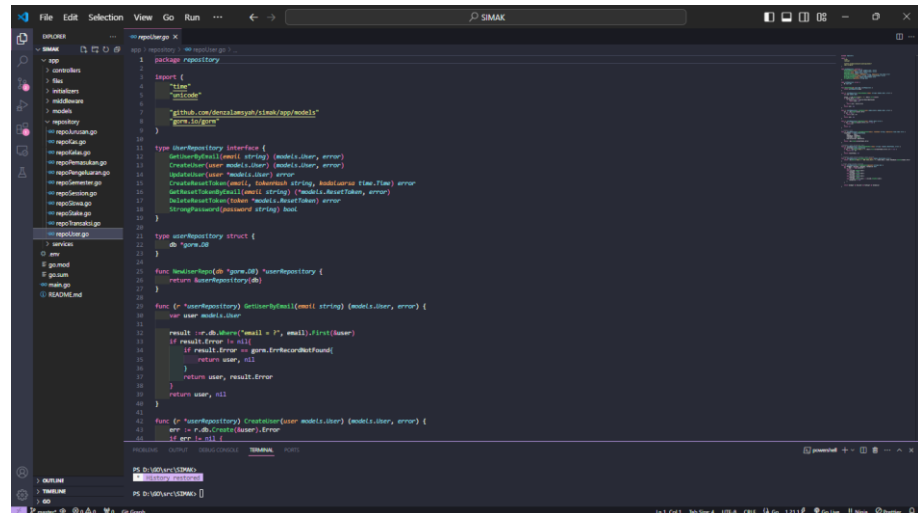
a. Struktur Folder

Seperti halnya pada *front end*, struktur folder di *back end* juga digunakan untuk merapikan dan mempermudah dalam penulisan kode program.



b. Kode Program *Repository*

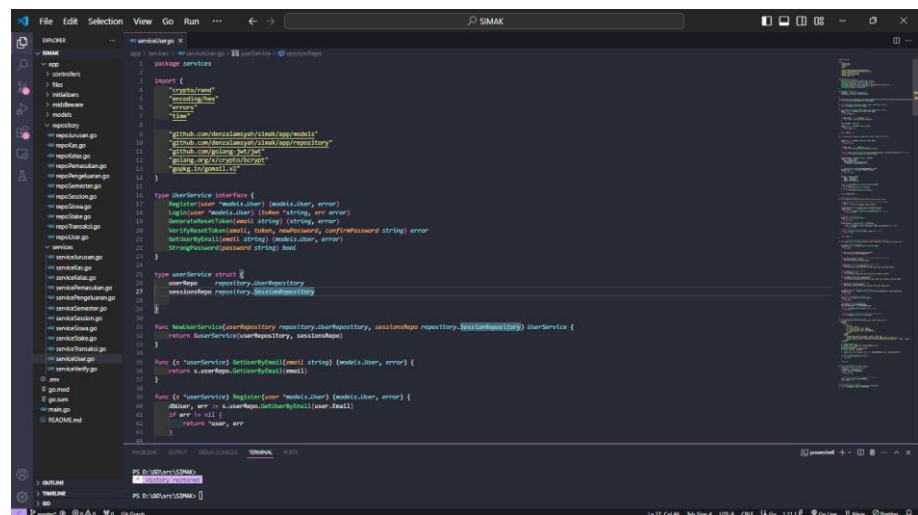
Kode program pada *repository*, mengurus akses data dan interaksi dengan *database*. Ini adalah tempat di mana logika terkait pengambilan, penyimpanan, pembaruan, dan penghapusan data ditempatkan.



Gambar 4. 40 Kode Program *Back End Repository*

c. Kode Program *Service*

Kode program pada *services*, mengandung logika bisnis utama aplikasi. *Services* berinteraksi dengan *repository* untuk mengambil dan menyimpan data, serta melakukan operasi atau manipulasi data yang lebih kompleks.



Gambar 4. 41 Kode Program *Back End Service*

d. Kode Program *Controller*

Kode program *controller* digunakan untuk mengurus logika presentasi dan interaksi dengan pengguna. Program ini merupakan tempat di mana *request* HTTP diterima dan direspon. *Controller* akan memanggil *service* untuk memproses data dan kemudian mengembalikan hasilnya kepada pengguna.

e. Kode Program *Models*

[illegible]

f. Kode Program *Middleware*

Bagian ini digunakan untuk memproses *request* sebelum mencapai fungsi *handler* utama atau untuk memproses *response* sebelum dikirim kembali ke *client*. *Middleware* bertindak sebagai lapisan penghubung antara aplikasi atau layanan untuk memproses data atau permintaan sebelum mencapai tujuan akhir.

g. Kode Program *Main*

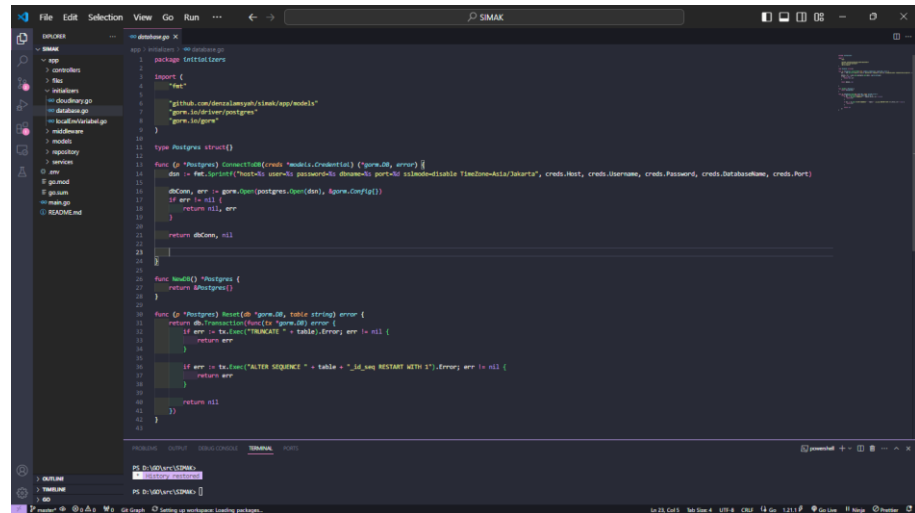
```

1 // main.go
2
3 package main
4
5 import (
6     "log"
7     "net/http"
8     "os"
9     "strings"
10
11     "github.com/gin-gonic/gin"
12     "gorm.io/gorm"
13 )
14
15 // GinRoutes
16 func GinRoutes(db *gorm.DB) {
17     router := gin.Default()
18
19     // User
20     router.GET("/users", func(c *gin.Context) {
21         // TODO: Implement user listing logic
22     })
23
24     // User Registration
25     router.POST("/register", func(c *gin.Context) {
26         // TODO: Implement user registration logic
27     })
28
29     // User Login
30     router.POST("/login", func(c *gin.Context) {
31         // TODO: Implement user login logic
32     })
33
34     // User Logout
35     router.POST("/logout", func(c *gin.Context) {
36         // TODO: Implement user logout logic
37     })
38
39     // User Profile
40     router.GET("/profile", func(c *gin.Context) {
41         // TODO: Implement user profile logic
42     })
43
44     // User Password Reset
45     router.POST("/reset-password", func(c *gin.Context) {
46         // TODO: Implement user password reset logic
47     })
48
49     // User Password Change
50     router.POST("/change-password", func(c *gin.Context) {
51         // TODO: Implement user password change logic
52     })
53
54     // User Account Deletion
55     router.DELETE("/delete", func(c *gin.Context) {
56         // TODO: Implement user account deletion logic
57     })
58
59     // User Account Activation
60     router.PUT("/activate", func(c *gin.Context) {
61         // TODO: Implement user account activation logic
62     })
63
64     // User Account Deactivation
65     router.PUT("/deactivate", func(c *gin.Context) {
66         // TODO: Implement user account deactivation logic
67     })
68
69     // User Account Verification
70     router.GET("/verify", func(c *gin.Context) {
71         // TODO: Implement user account verification logic
72     })
73
74     // User Account Confirmation
75     router.POST("/confirm", func(c *gin.Context) {
76         // TODO: Implement user account confirmation logic
77     })
78
79     // User Account Cancellation
80     router.DELETE("/cancel", func(c *gin.Context) {
81         // TODO: Implement user account cancellation logic
82     })
83
84     // User Account Recovery
85     router.GET("/recover", func(c *gin.Context) {
86         // TODO: Implement user account recovery logic
87     })
88
89     // User Account Search
90     router.GET("/search", func(c *gin.Context) {
91         // TODO: Implement user account search logic
92     })
93
94     // User Account Request
95     router.POST("/request", func(c *gin.Context) {
96         // TODO: Implement user account request logic
97     })
98
99     // User Account Download
100    router.GET("/download", func(c *gin.Context) {
101        // TODO: Implement user account download logic
102    })
103
104    state := gin.Default()
105}
106
107 // Main
108 func main() {
109     db, err := gorm.Open(dialect, config.DB)
110     if err != nil {
111         log.Fatalf("Gorm error: %v", err)
112     }
113
114     GinRoutes(db)
115
116     log.Println("Server is running on port 8080")
117     http.ListenAndServe(":8080", nil)
118 }

```

h. Kode Program *Initializers*

Initializers digunakan untuk mengatur setiap konfigurasi yang dipakai di dalam proyek aplikasi *golang*. Dalam hal ini, penulis mengatur konfigurasi seperti ke *database postgresQL*, *cloudinary* untuk *upload* gambar secara *online* dan *gmail* untuk membuat verifikasi melalui *email*.



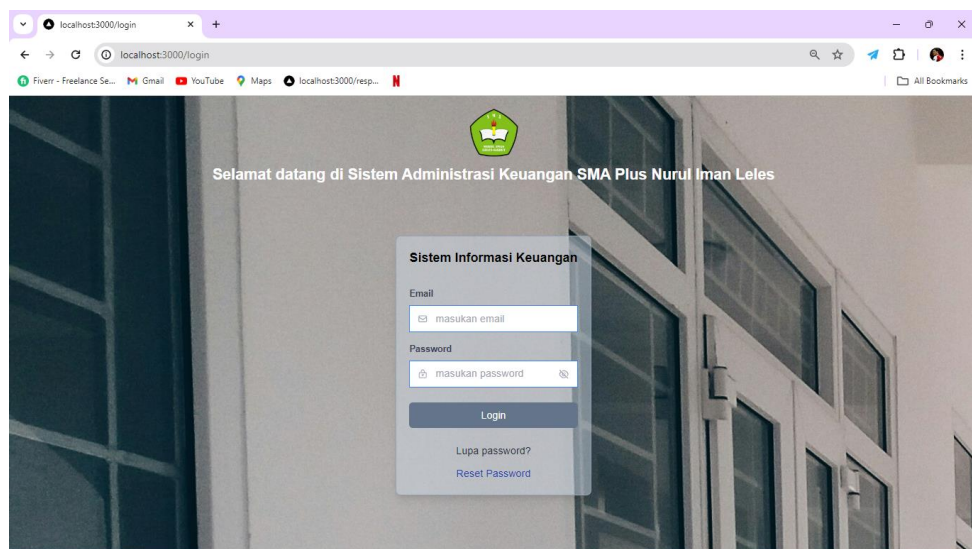
Gambar 4. 46 Kode Program *Back End Initializers*

4.5.3. Hasil Perancangan

Hasil perancangan dari sistem informasi administrasi keuangan SMA Plus Nurul Iman Leles mencakup berbagai fitur untuk meningkatkan efisiensi dan akurasi pengelolaan keuangan. Dengan antarmuka yang ramah pengguna, sistem ini diharapkan mendukung transparansi, akuntabilitas, dan standar pengelolaan keuangan di sekolah. Di bawah ini, merupakan contoh hasil perancangan sistem informasi administrasi keuangan SMA Plus Nurul Iman Leles.

1. Halaman *Login*

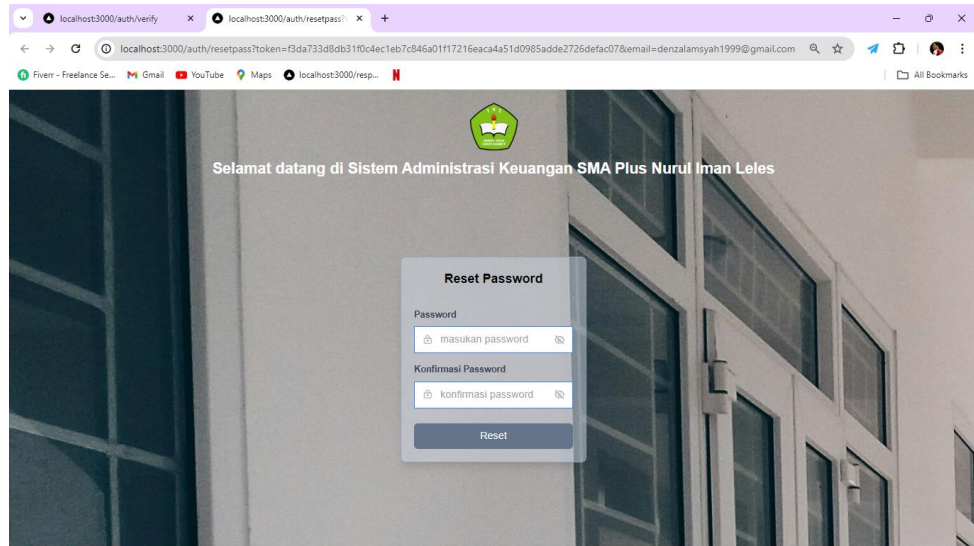
Desain halaman *login* ini dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan mudah digunakan. Adapun tampilannya bisa dilihat pada gambar 4.47 berikut.



Gambar 4. 47 Tampilan Halaman *Login*

2. Halaman *Reset Password*

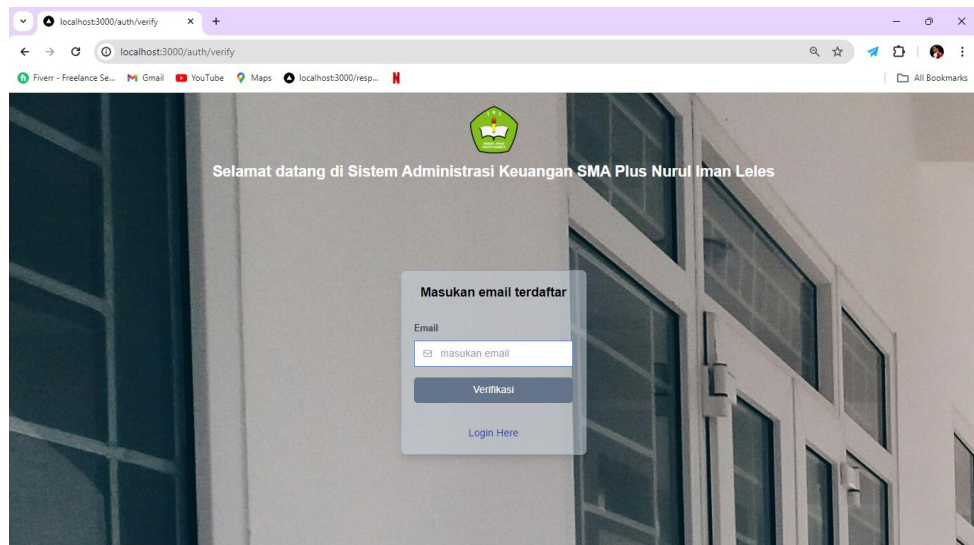
Desain halaman *reset password* ini dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan mudah digunakan. Adapun tampilannya bisa dilihat pada gambar 4.48 berikut.



Gambar 4. 48 Tampilan Halaman *Reset Password*

3. Halaman Verifikasi

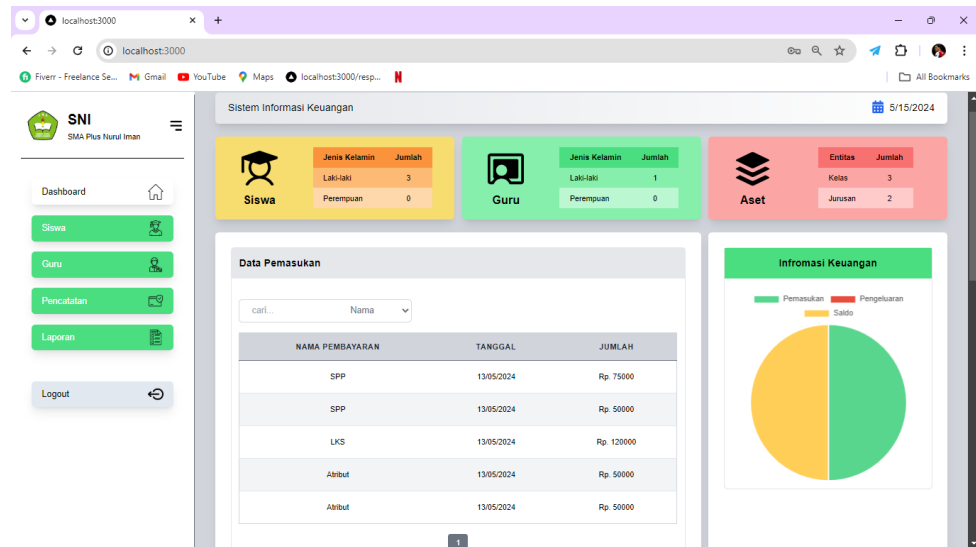
Desain halaman verifikasi ini dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan mudah digunakan. Adapun tampilannya bisa dilihat pada gambar 4.49 berikut.



Gambar 4. 49 Tampilan Halaman Verifikasi

4. Dashboard

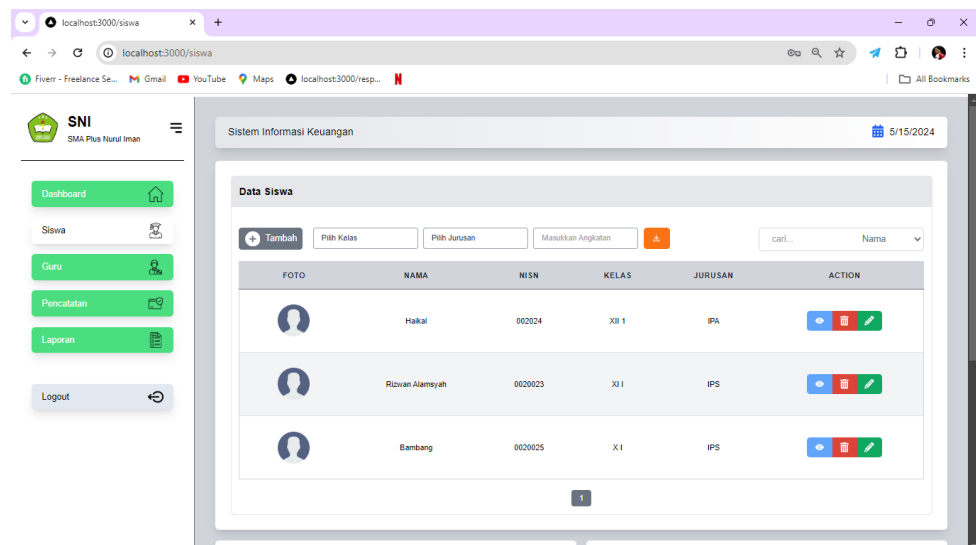
Desain halaman *dashboard* ini dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan mudah digunakan. Adapun tampilannya bisa dilihat pada gambar 4.50 berikut.



Gambar 4. 50 Tampilan *Dashboard*

5. Halaman Siswa

Desain halaman siswa ini dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan mudah digunakan. Adapun tampilannya bisa dilihat pada gambar 4.51 berikut.



Gambar 4. 51 Tampilan Halaman Siswa

Desain *form* tambah siswa ini dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan mudah digunakan. Adapun tampilannya bisa dilihat pada 4.52 berikut.

Gambar 4. 52 Tampilan *Form* Tambah Siswa

6. Halaman Guru

Desain halaman guru ini dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan mudah digunakan. Adapun tampilannya bisa dilihat pada gambar 4.53 berikut.

Gambar 4. 53 Tampilan Halaman Guru

Desain *form* tambah guru ini sudah dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan

mudah digunakan. Adapun tampilannya bisa dilihat pada gambar 4.54 berikut.

Gambar 4. 54 Tampilan Form Tambah Guru

7. Halaman Pencatatan

Desain halaman pencatatan ini sudah dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan mudah digunakan. Adapun tampilannya bisa dilihat pada gambar 4.55 berikut.

Gambar 4. 55 Tampilan Halaman Pencatatan

Desain *form* tambah pencatatan ini dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan

mudah digunakan, dirancang dengan desain minimalis namun fitur yang cukup optimal. Adapun tampilannya bisa dilihat pada gambar 4.56 berikut.

The screenshot shows a web browser at localhost:3000/pembayaran. A modal titled 'Tambah Pembayaran Siswa' is open. It contains the following fields: 'Nama Siswa' (a dropdown menu), 'Nama Pembayaran' (a text input), 'Semester' (a dropdown menu), 'Bulan' (a dropdown menu), 'Tanggal Bayar' (a date picker), and 'Jumlah' (a text input with '0' entered). At the bottom of the modal are two buttons: 'Tambah' (blue) and 'Batal' (white). The background shows a sidebar with 'Dashboard', 'Siswa', 'Guru', 'Pencatatan', 'Laporan', and 'Logout' options, and a table with columns 'NAMA PEMBAYARAN', 'NAMA SISWA', 'TANGGAL', and 'JUMLAH'.

Gambar 4. 56 Tampilan *Form* Tambah Pembayaran

8. Halaman Laporan

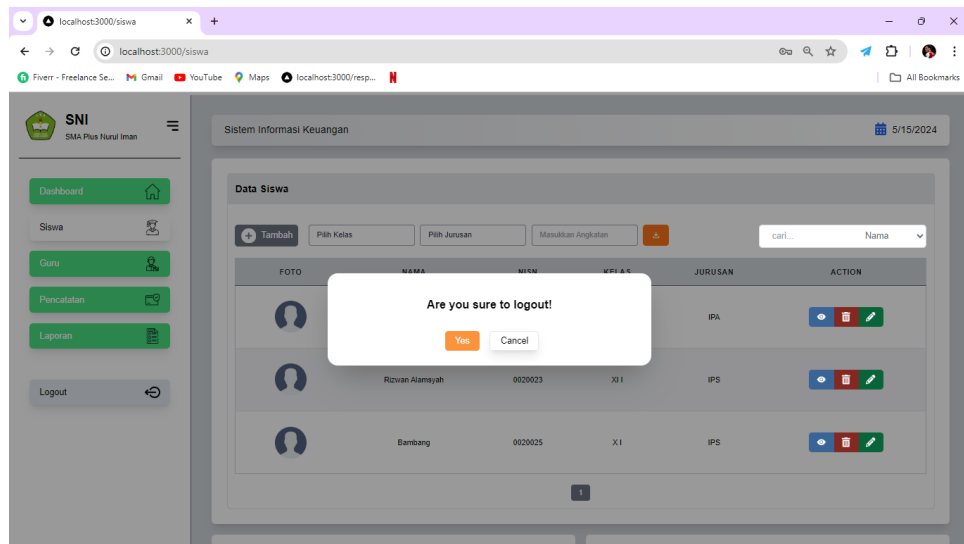
Desain halaman ini sudah dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan mudah digunakan. Adapun tampilannya bisa dilihat pada gambar 4.57 berikut.

The screenshot shows a web browser at localhost:3000/laporan. The page title is 'Keuangan'. It features a sidebar with 'Dashboard', 'Siswa', 'Guru', 'Pencatatan', 'Laporan', and 'Logout' options. The main content area contains two tables. The first table, 'Pemasukan', has columns 'NAMA PEMBAYARAN', 'TANGGAL', and 'JUMLAH'. It lists five entries: SPP (Rp. 75000), SPP (Rp. 50000), LKS (Rp. 120000), Abribud (Rp. 50000), and Abribud (Rp. 50000). Below this table is a summary row 'JUMLAH PEMASUKAN' with a value of 'Rp. 0'. The second table, 'Pengeluaran', also has columns 'NAMA PEMBAYARAN', 'TANGGAL', and 'JUMLAH'. It shows 'Data not found' and a summary row 'JUMLAH PENGELUARAN' with a value of 'Rp. 0'. At the bottom, there is a summary section with 'TOTAL PEMASUKAN', 'TOTAL PENGELUARAN', and 'SALDO'.

Gambar 4. 57 Tampilan Laporan

9. Logout

Desain *logout* ini dibuat sesuai perancangan yang bertujuan untuk memberikan pengalaman pengguna yang nyaman dan mudah digunakan. Adapun tampilannya bisa dilihat pada gambar 4.58 berikut.



Gambar 4. 58 Tampilan Halaman *Logout*

4.6. Pengujian Sistem

Blackbox testing digunakan dalam pengujian sistem informasi ini. Pengujian ini menitikberatkan pada kebutuhan perangkat lunak. *Blackbox testing* diharapkan mampu mengidentifikasi dan memastikan bahwa hasil dari data *input* dan *output* sesuai dengan yang diinginkan. Adapun hasil pengujiannya, bisa dilihat pada tabel berikut.

1. Pengujian Fitur *Login*

Pada fitur *login* terdapat enam kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 21 Pengujian Fitur *Login*

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi <i>email</i> atau <i>password</i>	Tidak bisa masuk sistem, dan muncul pesan kesalahan	Sesuai
2.	Admin mengisi <i>email</i> saja atau <i>password</i> saja	Tidak bisa masuk sistem, dan muncul pesan kesalahan	Sesuai
3.	Admin salah mengisi <i>email</i>	Tidak bisa masuk sistem, dan muncul pesan kesalahan	Sesuai

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
4.	Admin salah mengisi <i>password</i>	Tidak bisa masuk sistem, dan muncul pesan kesalahan	Sesuai
5.	Admin mengisi <i>email</i> dan <i>password</i> yang benar	Masuk ke sistem, dan muncul pesan berhasil	Sesuai
6.	Admin mengklik <i>reset password</i>	Masuk ke halaman verifikasi	Sesuai

2. Pengujian Fitur Verifikasi *Email*

Pada fitur verifikasi *email* terdapat lima kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 22 Pengujian Fitur Verifikasi *Email*

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi <i>email</i>	<i>Link</i> verifikasi tidak dikirim ke <i>email</i> , dan muncul pesan kesalahan	Sesuai
2.	Admin mengisi <i>email</i> yang salah	<i>Link</i> verifikasi tidak dikirim ke <i>email</i> , dan muncul pesan kesalahan	Sesuai
3.	Admin mengisi <i>email</i> yang benar	<i>Link</i> verifikasi dikirim ke <i>email</i> , dan muncul pesan	Sesuai
4.	Admin mengklik <i>link</i> kadaluwarsa	Masuk ke halaman <i>reset password</i> tapi tidak bisa mengganti <i>password</i>	Sesuai
5.	Admin mengklik <i>link</i> aktif	Masuk ke halaman <i>reset password</i> dan bisa mengganti <i>password</i>	Sesuai

3. Pengujian Fitur *Reset Password*

Pada fitur *reset password* terdapat lima kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 23 Pengujian Fitur *Reset Password*

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi <i>password</i> baru dan <i>confirm password</i>	Muncul pesan kesalahan	Sesuai
2.	Admin mengisi <i>password</i> baru dan <i>confirm password</i> kurang dari 8 karakter	Muncul pesan kesalahan	Sesuai
3.	Admin mengisi <i>password</i> baru dan <i>confirm password</i> tidak melibatkan huruf besar, kecil, angka dan karakter tertentu	Muncul pesan kesalahan	Sesuai
4.	Admin mengisi <i>confirm password</i> yang tidak sama dengan <i>password</i>	Muncul pesan kesalahan	Sesuai
5.	Admin mengisi <i>password</i> baru dan <i>confirm password</i> yang benar	Berhasil mengganti <i>password</i> , otomatis masuk ke halaman <i>login</i> dan muncul pesan	Sesuai

4. Pengujian Fitur Tambah Data Siswa

Pada fitur ini terdapat enam kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 24 Pengujian Fitur Tambah Data Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi form	Menampilkan <i>form required</i>	Sesuai
2.	Admin tidak mengisi jenis kelamin atau agama atau kelas atau jurusan	Gagal menambahkan data dan menampilkan pesan kesalahan	Sesuai
3.	Admin mengisi NISN yang sama dengan yang sudah ada	Gagal menambahkan data dan menampilkan pesan kesalahan	Sesuai
4.	Admin tidak mengisi gambar	Otomatis diisi dengan <i>default</i> gambar	Sesuai
5.	Admin mengisi form dengan benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
6.	Admin mengklik tombol batal	Membatalkan proses tambah data	Sesuai

5. Pengujian Fitur Ubah Data Siswa

Pada fitur ini, terdapat lima kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 25 Pengujian Fitur Ubah Data Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengubah inputan	Berhasil mengubah data dan menampilkan pesan	Sesuai

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
2.	Admin mengisi NISN yang sama dengan yang sudah ada	Gagal mengubah data dan menampilkan pesan kesalahan	Sesuai
3.	Admin melakukan perubahan dengan benar	Berhasil mengubah data dan menampilkan pesan	Sesuai
4.	Admin mengubah gambar	Gambar akan diganti dengan yang baru	Sesuai
5.	Admin mengklik tombol batal	Membatalkan proses ubah data	Sesuai

6. Pengujian Fitur Hapus Data Siswa

Pada fitur ini, terdapat tiga kasus percobaan, dan masing-masing *kasus* percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 26 Pengujian Fitur Hapus Data Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menghapus data yang memiliki relasi	Gagal menghapus data dan menampilkan pesan kesalahan	Sesuai
2.	Admin menghapus data yang benar	Berhasil menghapus data dan muncul pesan kesalahan	Sesuai
3.	Admin mengklik tombol batal	Membatalkan proses hapus data	Sesuai

7. Pengujian Fitur Detail Data Siswa

Pada fitur detail data siswa terdapat dua kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 27 Pengujian Fitur Detail Data Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mengklik tombol detail	Menampilkan data siswa berdasarkan nisn	Sesuai
2.	Admin mengklik tombol cancel	Kembali ke halaman sebelumnya	Sesuai

8. Pengujian Fitur Pencarian Data Siswa

Pada fitur pencarian data siswa terdapat empat kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 28 Pengujian Fitur Pencarian Data Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mencari data berdasarkan nama	Menampilkan data siswa berdasarkan nama	Sesuai
2.	Admin mencari data berdasarkan nisn	Menampilkan data siswa berdasarkan nisn	Sesuai
3.	Admin mencari data berdasarkan kelas	Menampilkan data siswa berdasarkan kelas	Sesuai
4.	Admin mencari data berdasarkan jurusan	Menampilkan data siswa berdasarkan jurusan	Sesuai

9. Pengujian Fitur *Download* Data Siswa

Pada fitur *download* data siswa terdapat empat kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 29 Pengujian Fitur *Download* Data Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mendownload data berdasarkan kelas	Menghasilkan file excel berisi data siswa berdasarkan kelas	Sesuai
2.	Admin mendownload data berdasarkan jurusan	Menghasilkan file excel berisi data siswa berdasarkan jurusan	Sesuai
3.	Admin mendownload data berdasarkan angkatan	Menghasilkan file excel berisi data siswa berdasarkan angkatan	Sesuai
4.	Menghasilkan file excel berisi data siswa berdasarkan kelas, jurusan dan angkatan	Menghasilkan file excel berisi data siswa berdasarkan kelas, jurusan dan angkatan	Sesuai

10. Pengujian Fitur Tambah Data Kelas

Pada fitur tambah data kelas terdapat empat kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 30 Pengujian Fitur Tambah Data Kelas

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi form	Gagal menambah data dan menampilkan pesan kesalahan	Sesuai
2.	Admin memasukan kode kelas yang sama dengan yang sudah ada	Gagal menambah data dan menampilkan pesan kesalahan	Sesuai

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
3.	Admin mengisi form dengan benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
4.	Admin mengklik tombol batal	Membatalkan proses menambah data	Sesuai

11. Pengujian Fitur Ubah Data Kelas

Pada fitur ubah data kelas terdapat empat kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 31 Pengujian Fitur Ubah Data Kelas

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengubah inputan	Berhasil mengubah data dan menampilkan pesan	Sesuai
2.	Admin mengisi kode kelas yang sama dengan yang sudah ada	Gagal mengubah data dan menampilkan pesan kesalahan	Sesuai
3.	Admin melakukan perubahan dengan benar	Berhasil mengubah data dan menampilkan pesan	Sesuai
4.	Admin mengklik tombol batal	Membatalkan proses ubah data	Sesuai

12. Pengujian Fitur Hapus Data Kelas

Pada fitur hapus data kelas terdapat tiga kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 32 Pengujian Fitur Hapus Data Kelas

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menghapus data yang memiliki relasi	Gagal menghapus data dan menampilkan pesan kesalahan	Sesuai

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
2.	Admin menghapus data yang benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
3.	Admin mengklik tombol batal	Membatalkan proses hapus data	Sesuai

13. Pengujian Fitur Pencarian Data Kelas

Pada fitur pencarian data kelas terdapat satu kasus percobaan, dan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 33 Pengujian Fitur Pencarian Data Kelas

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mencari data berdasarkan nama	Menampilkan data kelas berdasarkan nama	Sesuai

14. Pengujian Fitur Tambah Data Jurusan

Pada fitur tambah data jurusan terdapat empat kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 34 Pengujian Fitur Tambah Data Jurusan

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi form	Gagal menambah data dan menampilkan pesan kesalahan	Sesuai
2.	Admin memasukan kode jurusan yang sama dengan yang sudah ada	Gagal menambah data dan menampilkan pesan kesalahan	Sesuai
3.	Admin mengisi form dengan benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
4.	Admin mengklik tombol batal	Membatalkan proses menambah data	Sesuai

15. Pengujian Fitur Ubah Data Jurusan

Pada fitur ubah data jurusan terdapat empat kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 35 Pengujian Fitur Ubah Data Jurusan

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengubah inputan	Berhasil mengubah data dan menampilkan pesan	Sesuai
2.	Admin mengisi kode jurusan yang sama dengan yang sudah ada	Gagal mengubah data dan menampilkan pesan kesalahan	Sesuai
3.	Admin melakukan perubahan dengan benar	Berhasil mengubah data dan menampilkan pesan	Sesuai
4.	Admin mengklik tombol batal	Membatalkan proses ubah data	Sesuai

16. Pengujian Fitur Hapus Data Jurusan

Pada fitur hapus data jurusan terdapat tiga kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 36 Pengujian Fitur Hapus Data Jurusan

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menghapus data yang memiliki relasi	Gagal menghapus data dan menampilkan pesan kesalahan	Sesuai
2.	Admin menghapus data yang benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
3.	Admin mengklik tombol batal	Membatalkan proses hapus data	Sesuai

17. Pengujian Fitur Pencarian Data Jurusan

Pada fitur pencarian data jurusan terdapat satu kasus percobaan, hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 37 Pengujian Fitur Pencarian Data Jurusan

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mencari data berdasarkan nama	Menampilkan data jurusan berdasarkan nama	Sesuai

18. Pengujian Fitur Tambah Data Guru

Pada fitur tambah data guru terdapat enam kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 38 Pengujian Fitur Tambah Data Guru

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi form	Menampilkan <i>form required</i>	Sesuai
2.	Admin tidak mengisi jenis kelamin atau agama atau jabatan	Gagal menambahkan data dan menampilkan pesan kesalahan	Sesuai
3.	Admin mengisi NIP yang sama dengan yang sudah ada	Gagal menambahkan data dan menampilkan pesan kesalahan	Sesuai
4.	Admin tidak mengisi gambar	Otomatis diisi dengan <i>default</i> gambar	Sesuai
5.	Admin mengisi form dengan benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
6.	Admin mengklik tombol batal	Membatalkan proses tambah data	Sesuai

19. Pengujian Fitur Ubah Data Guru

Pada fitur ubah data guru terdapat lima kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 39 Pengujian Fitur Ubah Data Guru

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengubah inputan	Berhasil mengubah data dan menampilkan pesan	Sesuai
2.	Admin mengisi NIP yang sama dengan yang sudah ada	Gagal mengubah data dan menampilkan pesan kesalahan	Sesuai
3.	Admin melakukan perubahan dengan benar	Berhasil mengubah data dan menampilkan pesan	Sesuai
4.	Admin mengubah gambar	Gambar akan diganti dengan yang baru	Sesuai
5.	Admin mengklik tombol batal	Membatalkan proses ubah data	Sesuai

20. Pengujian Fitur Hapus Data Guru

Pada fitur hapus data guru terdapat tiga kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 40 Pengujian Fitur Hapus Data Guru

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menghapus data yang memiliki relasi	Data gagal terhapus dan muncul pesan kesalahan	Sesuai
2.	Admin menghapus data yang benar	Data berhasil dihapus dan muncul pesan	Sesuai

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
3.	Admin mengklik tombol batal	Membatalkan proses hapus data	Sesuai

21. Pengujian Fitur Detail Data Guru

Pada fitur detail data guru terdapat dua kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 41 Pengujian Fitur Detail Data Guru

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mengklik tombol detail	Menampilkan data guru berdasarkan nip	Sesuai
2.	Admin mengklik tombol cancel	Kembali ke halaman sebelumnya	Sesuai

22. Pengujian Fitur Pencarian Data Guru

Pada fitur pencarian data guru terdapat tiga kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 42 Pengujian Fitur Pencarian Data Guru

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mencari data berdasarkan nama	Menampilkan data guru berdasarkan nama	Sesuai
2.	Admin mencari data berdasarkan nip	Menampilkan data guru berdasarkan nip	Sesuai
3.	Admin mencari data berdasarkan jabatan	Menampilkan data guru berdasarkan jabatan	Sesuai

23. Pengujian Fitur *Download* Data Guru

Pada fitur download data guru terdapat satu kasus percobaan, dan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 43 Pengujian Fitur *Download* Data Guru

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mengklik tombol <i>download</i>	Menghasilkan file excel berisi data guru	Sesuai

24. Pengujian Fitur Pembayaran Siswa

Pada fitur pembayaran siswa terdapat lima kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 44 Pengujian Fitur Pembayaran Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi form	Menampilkan <i>form required</i>	Sesuai
2.	Admin tidak mengisi nsn siswa atau nama transaksi	Gagal menambahkan data dan menampilkan pesan kesalahan	Sesuai
3.	Admin mengisi nsn siswa dan nama transaksi dan bulan dan semester yang sama dengan data yang sudah ada	Gagal menambahkan data dan menampilkan pesan kesalahan	Sesuai
4.	Admin mengisi form dengan benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
5.	Admin mengklik tombol batal	Membatalkan proses tambah data	Sesuai

25. Pengujian Fitur Ubah Pembayaran Siswa

Pada fitur ubah pembayaran siswa terdapat empat kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 45 Pengujian Fitur Ubah Pembayaran Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengubah inputan	Berhasil mengubah data dan menampilkan pesan	Sesuai
2.	Admin mengubah nisl siswa dan nama transaksi dan bulan dan semester yang sama dengan data yang sudah ada	Gagal mengubah data dan menampilkan pesan kesalahan	Sesuai
3.	Admin mengisi form dengan benar	Berhasil mengubah data dan menampilkan pesan	Sesuai
4.	Admin mengklik tombol batal	Membatalkan proses ubah data	Sesuai

26. Pengujian Fitur Hapus Pembayaran Siswa

Pada fitur hapus pembayaran siswa terdapat dua kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 46 Pengujian Fitur Hapus Pembayaran Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menghapus data yang benar	Berhasil menghapus data dan menampilkan pesan	Sesuai
2.	Admin mengklik tombol batal	Membatalkan proses hapus data	Sesuai

27. Pengujian Fitur Detail Pembayaran Siswa

Pada fitur detail pembayaran siswa terdapat dua kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 47 Pengujian Fitur Detail Pembayaran Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mengklik tombol detail	Menampilkan data pembayaran siswa berdasarkan id	Sesuai
2.	Admin mengklik tombol cancel	Kembali ke halaman sebelumnya	Sesuai

28. Pengujian Fitur Pencarian Pembayaran Siswa

Pada fitur pencarian pembayaran siswa terdapat dua kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 48 Pengujian Fitur Pencarian Pembayaran Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mencari data berdasarkan nama	Menampilkan data pembayaran siswa berdasarkan nama	Sesuai
2.	Admin mencari data berdasarkan tanggal	Menampilkan data pembayaran siswa berdasarkan tanggal	Sesuai

29. Pengujian Fitur *Download* Kwitansi Pembayaran Siswa

Pada fitur *download* kwitansi pembayaran siswa terdapat satu kasus percobaan, dan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 49 Pengujian Fitur *Download* Kwitansi Pembayaran Siswa

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mengklik tombol <i>download</i>	Menghasilkan file pdf berisi kwitansi pembayaran	Sesuai

30. Pengujian Fitur Tambah Transaksi

Pada fitur tambah transaksi terdapat empat kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 50 Pengujian Fitur Tambah Transaksi

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi form	Gagal menambah data dan menampilkan pesan kesalahan	Sesuai
2.	Admin memasukan nama dan tipe transaksi yang sama dengan yang sudah ada	Gagal menambah data dan menampilkan pesan kesalahan	Sesuai
3.	Admin mengisi form dengan benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
4.	Admin mengklik tombol batal	Membatalkan proses menambah data	Sesuai

31. Pengujian Fitur Ubah Transaksi

Pada fitur ubah transaksi terdapat empat kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 51 Pengujian Fitur Ubah Transaksi

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengubah inputan	Berhasil mengubah data dan menampilkan pesan	Sesuai
2.	Admin mengubah dengan nama dan tipe transaksi yang sama dengan yang sudah ada	Gagal mengubah data dan menampilkan pesan kesalahan	Sesuai
3.	Admin melakukan perubahan dengan benar	Berhasil mengubah data dan menampilkan pesan	Sesuai
4.	Admin mengklik tombol batal	Membatalkan proses ubah data	Sesuai

32. Pengujian Fitur Hapus Transaksi

Pada fitur hapus transaksi terdapat tiga kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 52 Pengujian Fitur Hapus Transaksi

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menghapus data yang benar	Berhasil menghapus data dan menampilkan pesan	Sesuai
2.	Admin mengklik tombol batal	Membatalkan proses hapus data	Sesuai
3.	Admin menghapus data yang memiliki relasi	Gagal menghapus data dan menampilkan pesan kesalahan	Sesuai

33. Pengujian Fitur Pencarian Transaksi

Pada fitur pencarian transaksi terdapat dua kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 53 Pengujian Fitur Pencarian Transaksi

No	Kasus Percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mencari data berdasarkan nama	Menampilkan data berdasarkan nama transaksi	Sesuai
2.	Admin mencari data berdasarkan kategori	Menampilkan data transaksi berdasarkan kategori	Sesuai

34. Pengujian Fitur Tambah Pemasukan

Pada fitur tambah pemasukan terdapat tiga kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 54 Pengujian Fitur Tambah Pemasukan

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi form	Gagal menambah data dan menampilkan pesan kesalahan	Sesuai
2.	Admin mengisi form dengan benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
3.	Admin mengklik tombol batal	Membatalkan proses menambah data	Sesuai

35. Pengujian Fitur Ubah Pemasukan

Pada fitur ubah pemasukan terdapat tiga kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 55 Pengujian Fitur Ubah Pemasukan

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengubah inputan	Data berhasil diubah dan muncul pesan	Sesuai
2.	Admin melakukan perubahan dengan benar	Data berhasil diubah dan muncul pesan	Sesuai
3.	Admin mengklik tombol batal	Membatalkan proses ubah data	Sesuai

36. Pengujian Fitur Hapus Pemasukan

Pada fitur hapus pemasukan terdapat dua kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 56 Pengujian Fitur Hapus Pemasukan

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menghapus data yang benar	Berhasil menghapus data dan menampilkan pesan	Sesuai
2.	Admin mengklik tombol batal	Membatalkan proses hapus data	Sesuai

37. Pengujian Fitur Pencarian Pemasukan

Pada fitur pencarian pemasukan terdapat dua kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 57 Pengujian Fitur Pencarian Pemasukan

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mencari data berdasarkan nama	Menampilkan data berdasarkan nama	Sesuai
2.	Admin mencari data berdasarkan tanggal	Menampilkan data transaksi berdasarkan tanggal	Sesuai

38. Pengujian Fitur *Download* Laporan Pemasukan

Pada fitur *download* laporan pemasukan terdapat satu kasus percobaan, dan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 58 Pengujian Fitur *Download* Laporan Pemasukan

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mengklik tombol <i>download</i>	Menghasilkan file pdf berisi laporan pemasukan	Sesuai

39. Pengujian Fitur Tambah Pengeluaran

Pada fitur tambah pengeluaran terdapat tiga kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 59 Pengujian Fitur Tambah Pengeluaran

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi form	Gagal menambah data dan menampilkan pesan kesalahan	Sesuai
2.	Admin mengisi form dengan benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
3.	Admin mengklik tombol batal	Membatalkan proses menambah data	Sesuai

40. Pengujian Fitur Ubah Pengeluaran

Pada fitur ubah pengeluaran terdapat tiga kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 60 Pengujian Fitur Ubah Pengeluaran

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengubah inputan	Data berhasil diubah dan menampilkan pesan	Sesuai
2.	Admin melakukan perubahan dengan benar	Data berhasil diubah dan menampilkan pesan	Sesuai
3.	Admin mengklik tombol batal	Membatalkan proses ubah data	Sesuai

41. Pengujian Fitur Hapus Pengeluaran

Pada fitur hapus pengeluaran terdapat dua kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 61 Pengujian Fitur Hapus Pengeluaran

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menghapus data yang benar	Berhasil menghapus data dan menampilkan pesan	Sesuai
2.	Admin mengklik tombol batal	Membatalkan proses hapus data	Sesuai

42. Pengujian Fitur Pencarian Pengeluaran

Pada fitur pencarian pengeluaran terdapat dua kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 62 Pengujian Fitur Pencarian Pengeluaran

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mencari data berdasarkan nama	Menampilkan data berdasarkan nama	Sesuai
2.	Admin mencari data berdasarkan tanggal	Menampilkan data transaksi berdasarkan tanggal	Sesuai

43. Pengujian Fitur *Download* Laporan Pengeluaran

Pada fitur download laporan pengeluaran terdapat satu kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 63 Pengujian Fitur *Download* Laporan Pengeluaran

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mengklik tombol <i>download</i>	Menghasilkan file pdf berisi laporan pemasukan	Sesuai

44. Pengujian Fitur Tambah Kas

Pada fitur tambah kas terdapat tiga kasus percobaan, dan masing-masing kasus percobaan, sejauh ini hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 64 Pengujian Fitur Tambah Kas

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengisi form	Gagal menambah data dan menampilkan pesan kesalahan	Sesuai
2.	Admin mengisi form dengan benar	Berhasil menambahkan data dan menampilkan pesan	Sesuai
3.	Admin mengklik tombol batal	Membatalkan proses menambah data	Sesuai

45. Pengujian Fitur Ubah Kas

Pada fitur ubah kas terdapat tiga kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 65 Pengujian Fitur Ubah Kas

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin tidak mengubah inputan	Berhasil mengubah data dan menampilkan pesan	Sesuai
2.	Admin melakukan perubahan dengan benar	Berhasil mengubah data dan menampilkan pesan	Sesuai
3.	Admin mengklik tombol batal	Membatalkan proses ubah data	Sesuai

46. Pengujian Fitur Hapus Kas

Pada fitur hapus kas terdapat dua kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 66 Pengujian Fitur Hapus Kas

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menghapus data yang benar	Berhasil menghapus data dan menampilkan pesan	Sesuai
2.	Admin mengklik tombol batal	Membatalkan proses hapus data	Sesuai

47. Pengujian Fitur Pencarian Kas

Pada fitur pencarian kas terdapat dua kasus percobaan, dan masing-masing kasus percobaan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 67 Pengujian Fitur Pencarian Kas

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin mencari data berdasarkan nama	Menampilkan data berdasarkan nama	Sesuai

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
2.	Admin mencari data berdasarkan nip	Menampilkan data transaksi berdasarkan nip	Sesuai

48. Pengujian Fitur *Download* Kwitansi Pembayaran Kas

Pada fitur *download* kwitansi pembayaran kas terdapat satu kasus percobaan, dan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 68 Pengujian Fitur *Download* Kwitansi Pembayaran Kas

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menekan tombol <i>download</i>	Menghasilkan file pdf berisi kwitansi pembayaran kas	Sesuai

49. Pengujian Fitur *Download* Kwitansi Pengambilan Kas

Pada fitur *download* kwitansi pengambilan kas, terdapat satu kasus percobaan dan hasilnya sudah sesuai dengan yang diharapkan.

Tabel 4. 69 Pengujian Fitur *Download* Kwitansi Pengambilan Kas

No	Kasus percobaan	Hasil yang Diinginkan	Hasil
1.	Admin menekan tombol <i>download</i>	Menghasilkan file pdf berisi kwitansi pengambilan kas	Sesuai

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Adapun kesimpulan dari penulisan skripsi yang berjudul Rancang Bangun Sistem Informasi Administrasi Keuangan SMA Plus Nurul Iman Leles Berbasis *Web* Menggunakan Metode *Scrum*, seperti berikut.

1. Sistem informasi administrasi keuangan yang dirancang untuk SMA Plus Nurul Iman Leles berhasil memenuhi kebutuhan dalam pengelolaan keuangan secara efektif dan efisien. Dengan menggunakan pendekatan yang sistematis, perancangan sistem ini telah melibatkan analisis dari kebutuhan pengguna, pemetaan proses bisnis, serta pengembangan model data yang komprehensif. Sistem ini dilengkapi berbagai fitur seperti pencatatan transaksi, manajemen laporan keuangan, dan pencatatan pembayaran siswa. Implementasi teknologi basis data yang terstruktur dan penggunaan antarmuka pengguna yang intuitif turut mendukung peningkatan efisiensi operasional dalam pengelolaan keuangan sekolah. Hasilnya, proses administrasi keuangan menjadi lebih transparan, mudah diaudit, dan meminimalisir kesalahan manual.
2. Pembangunan sistem informasi ini dilakukan sesuai dengan hasil perancangan yang telah disusun. Proses pengembangan melibatkan beberapa tahap penting, termasuk perancangan arsitektur sistem, pemrograman, pengujian, dan implementasi. Dalam tahap pembangunan, teknologi terkini seperti *framework web* dan basis data relasional digunakan untuk memastikan sistem berjalan dengan cepat dan andal. Pengujian yang dilakukan menunjukkan bahwa, sistem sudah berfungsi cukup baik sesuai dengan spesifikasi yang telah ditetapkan. Hasil percobaan sistem ini di lingkungan SMA Plus Nurul Iman Leles menunjukkan bahwa sistem dapat diintegrasikan dengan baik ke dalam proses bisnis yang ada, serta diterima dengan baik oleh pengguna. Dengan adanya sistem informasi ini, sekolah mampu mengelola keuangan dengan lebih efisien, mengurangi beban kerja staf administrasi, dan menyediakan

laporan keuangan yang akurat dan tepat waktu untuk pengambilan keputusan yang lebih baik.

5.2. Saran

Saran yang bisa diberikan penulis untuk pengembangan sistem informasi administrasi keuangan di SMA Plus Nurul Iman Leles yaitu:

1. Implementasikan sistem informasi yang sudah dibuat.
2. Meningkatkan fungsionalitas, dengan menambahkan fitur yang lebih lengkap.
3. Agar sistem dapat digunakan secara maksimal, pihak sekolah sebaiknya melakukan ujicoba atau pelatihan kepada staf administrasi mengenai cara penggunaan sistem ini.
4. Perlu ditingkatkan aspek keamanan data guna melindungi informasi dari akses yang tidak valid, dengan menggunakan enkripsi dan kontrol akses yang ketat.
5. Disarankan untuk melakukan pemeliharaan rutin pada sistem, untuk memastikan supaya sistem tetap berjalan dengan baik dan untuk mengatasi bug atau masalah lain yang mungkin terjadi.
6. Melakukan evaluasi secara berkala dan senantiasa mencoba menerima masukan dari pengguna untuk pengembangan lebih lanjut agar sistem dapat terus diperbarui sesuai kebutuhan sekolah.

DAFTAR PUSTAKA

- Aji Bayu Prasetyo, D., & Alfa Susetyo, Y. (2022). *Implementasi Information Schema Database Pada Postgre SQL Untuk Pembuatan Tabel Informasi Dengan Menggunakan Python Di PT XYZ* (Vol. 9, Issue 3). <http://jurnal.mdp.ac.id>
- Ananda, R. A., & Nisa, G. F. (2024). Analisis Dan Perancangan Layanan Streaming Film Berbasis Web Langganan Menggunakan Framework Nextjs. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(1). <https://doi.org/10.23960/jitet.v12i1.3967>.
- Aryanata Andipradana, & Kristoko Dwi Hartomo. (2021). Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum. *Jurnal Algoritma*, 19, 161–172.
- Baehaqi, A., Basit, M. S., Indrajit, R. E., & Kurniawan, R. D. (2023). Front End Learning Management System Development Using The Nextjs Framework. *Jurnal Teknik Informatika (Jutif)*, 4(4), 899–911. <https://doi.org/10.52436/1.jutif.2023.4.4.1273>
- Faqih, A. S., & Wahyudi, A. D. (2022). Rancang Bangun Sistem Informasi Penjualan Berbasis Web (Studi Kasus: Matchmaker). *Jurnal Teknologi Dan Sistem Informasi*, 3(2), 1–8. <http://jim.teknokrat.ac.id/index.php/JTSI>
- Firda Rosiana Tanjung, & Liptia Venica. (2023). Desain dan Implementasi Aplikasi Absensi Kepegawaian Berbasis Web Menggunakan Framework Laravel di Diskominfo Provinsi Jawa Barat. *Jurnal Media Infotama*, 19.
- Gladys Wahyu Khairunnisa, Issa Arwani, & Buce Trias Hanggara. (2022). Pengembangan Sistem Informasi Point of Sales berbasis Web menggunakan Framework Laravel (Studi Kasus: Meetup Station). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6, 1858–1864.
- Hajjul Ikram, & Muhammad Ardiansyah. (2024). Development of Online NAS Application User Guide Website using NextJS 13 and Rest API to Enhance User Understanding. *Jurnal Inovasi Teknologi Dan Rekayasa*, 9, 53–58.
- Hartono, B. (2020). *Cara Mudah Dan Cepat Belajar Pengembangan Sistem Informasi*. Yayasan Prima Agus Teknik Bekerja sama dengan Uninvertistas Sains & Teknologi Komputer.

- Hendri, H., Hasiholan Manurung, J. W., Ferian, R. A., Hanaatmoko, W. F., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Pengelolaan Masjid Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 3(2), 107. <https://doi.org/10.32493/jtsi.v3i2.4694>
- Ilham Taufiq, Rilfa Dini, Bobi Zinaidin Zidan, & Sayekti Harits suryawan. (2023). Sistem Informasi Terpadu Surat Perintah Perjalanan Dinas Dan Arsip Pada Bagian Pengadaan Barang Dan Jasa Sekretariat Daerah Kabupaten Berau. *Jurnal GEMBIRA (Pengabdian Kepada Masyarakat)*.
- Kristanto, A. A. Y. H. and J. E. S. (2020). Implementasi Golang dan New Simple Queue pada Sistem Sandbox Pihak Ketiga Berbasis Rest API. *Jurnal Rekayasa Sistem Dan Teknologi Informasi*, 745–750.
- Muhamad Rizky, & Yuni Sugiarti. (2021). Penggunaan Metode Scrum Dalam Pengembangan Perangkat Lunak: Literature Review. *Journal of Computer Science an Engineering (JCSE)*, 3, 41–48.
- Nawaf Naofal, Muhammad Ilman Aqilaa, & Rolly Maulana Awangga. (2023). *Pemrograman Golang Untuk Pemula* (R. M. Awangga, Ed.). PT. Penerbit Buku Pedia.
- Ni Kadek Dwi Sabrina, Dian Pramana, & Tubagus Mahendra Kusuma. (2023). Implementation of Golang and ReactJS in the COVID-19 Vaccination Reservation System. *ADI Journal on Recent Innovation (AJRI)*, 5, 1–12.
- Novitasari, Y. S., Adrian, Q. J., & Kurnia, W. (2021). Rancang Bangun Sistem Informasi Media Pembelajaran Berbasis Website (Studi Kasus: Bimbingan Belajar De Potlood). *Jurnal Teknologi Dan Sistem Informasi (JTISI)*, 2(3), 136–147. <http://jim.teknokrat.ac.id/index.php/JTISI>
- Oktaviani, E., Rodianto, Noviana, S., & Nawassyarif. (2020). Rancang Bangun Sistem Informasi Untuk Meningkatkan Tata Kelola Administrasi Surat Menyurat. *Jurnal Informatika, Teknologi Dan Sains*, 2(3), 203–207. <https://doi.org/10.51401/jinteks.v2i3.757>
- Pamungkas, F., & Setiaji, H. (n.d.). *Implementasi Clean Architecture Pada Pembuatan Api Menggunakan Golang*.

- Priadi, S., & Idris, M. (n.d.). *Implementasi Rest Dalam Membangun Web Service Menggunakan Golang (Studi Kasus: Modul Berita Aplikasi Satudikti)*. 8(2), 2023.
- Putra, F. D., Riyanto, J., & Zulfikar, A. F. (2020). Rancang Bangun Sistem Informasi Manajemen Aset pada Universitas Pamulang Berbasis WEB. *Journal of Engineering, Technology, and Applied Science*, 2(1), 32–50. <https://doi.org/10.36079/lamintang.jetas-0201.93>
- Raditya Aryaputra, & Kristoko Dwi Hartomo. (2023). Sistem Informasi Persediaan Suku Cadang Menggunakan Model Proses Scrum. *Jurnal Teknik Informatika Dan Sistem Informasi*, 10, 860–874.
- Raharjo, B. (2021). *Sistem Manajemen Database* (M. caroline Wibowo, Ed.). Yayasan Prima Agus Teknik.
- Salim, R., Arisandi, D., & Hendryli, J. (2022). Pembuatan Aplikasi Mostrans Transporter Berbasis Mobile Menggunakan React-Native JavaScript. *Jurnal Ilmu Komputer Dan Sistem Informasi*, 10(1). <https://doi.org/10.24912/jiksi.v10i1.17856>
- Seta, B., Darajat, P. P., Choirina, P., & Mubarak, F. A. (2021). Rancang Bangun Sistem Informasi Administrasi Keuangan Sekolah Berbasis Website Menggunakan Framework Laravel Di Smp Lukman Hakim Pakisaji. *G-Tech: Jurnal Teknologi Terapan*, 5(1), 376–381. <https://doi.org/10.33379/gtech.v5i1.1152>
- Setiadana, E. (2021). Pengembangan Sistem Penagihan Biaya Kuliah Dengan Fitur WhatsApp Menggunakan Metode Scrum Berbasis Website. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 8(3), 1252–1264. <https://doi.org/10.35957/jatisi.v8i3.1041>
- Shadiq, J., Safei, A., & Loly, R. W. R. (2021). Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing. *Information Management For Educators And Professionals: Journal of Information Management*, 5(2), 97. <https://doi.org/10.51211/imbi.v5i2.1561>
- Shalludin, ahmad, & Maulida, I. (2016). *Analisis Sistem Informasi*. Sekolah Tinggi Manajemen Informatika Dan Ilmu Komputer Stmik Indonesia Banjarmasin.

- Siti Sauda, & M Barokah. (2022). Penerapan Nodejs Dan Postgresql Sebagai *Backend* Pada Aplikasi Ecommerce Localla. *Infotech Journal*, 8.
- Solahudin, M. (2021). Rancang Bangun Sistem Informasi Akademik Sekolah (SIAS) Berbasis Website. *DoubleClick: Journal of Computer and Information Technology*, 4(2), 107. <https://doi.org/10.25273/doubleclick.v4i2.8315>
- Sri Puji Utami, Kyky Eviyanti, Wening Sari, & Sri Chusri Haryanti. (2022). Rancang *Bangun* Aplikasi Edukasi Tuberkulosis Menggunakan Metode Scrum. *Jurnal Inovtek Polbeng - Seri Informatika*, 7.
- Studi Sistem Informasi Universitas Trilogi Jakarta Jl TMP Kalibata No, P., Tiga Kec Pancoran, D., Kunci, K., Kost, R., & Informasi, S. (2021). Rancang Bangun Sistem Informasi Sewa Rumah Kost (E-Kost) Berbasis Website Chalidazia Nizar. *Jurnal Sistem Informasi Dan Sains Teknologi*, 3(1).
- Susanto, F. G. P., Fadlan, N. I. Y., & Haryani, P. (2023). Design of Web-Based Management Information System for Student Organizations in Kendal Regency Using Next.js Framework. *Compiler*, 12(1), 9. <https://doi.org/10.28989/compiler.v12i1.1616>
- Swala Putra, I. G. N., Satwika, I. P., & Eka Putra, I. G. J. (2020). Rancang Bangun Sistem Informasi Manajemen Administrasi Desa Berbasis Web Menggunakan Framework Laravel. *Jutisi: Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, 9(2), 163. <https://doi.org/10.35889/jutisi.v9i2.522>
- Utomo, A., Sutanto, Y., Tiningrum, E., & Susilowati, E. M. (2020). Pengujian Aplikasi *Transaksi* Perdagangan Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Bisnis Terapan*, 4(2), 133–140. <https://doi.org/10.24123/jbt.v4i2.2170>
- Wahyu Adi Prabowo, & Citra Wiguna. (2021). Sistem Informasi UMKM Bengkel Berbasis Web Menggunakan Metode Scrum. *Jurnal Media Informatika Budidarma*, 5, 149–156.
- Wahyudi. (2021). *Javascript Untuk Aplikasi Web* (S. Panji Pradana & H. Sukma, Eds.). Eureka Media Aksara.
- Wijaya, A., & Hendrastuty, N. (2022). Rancang Bangun Sistem Informasi Manajemen Kepegawaian (Simpeg) Berbasis Web (Studi Kasus: Pt Sembilan

Hakim Nusantara). *Jurnal Teknologi Dan Sistem Informasi (JTSI)*, 3(2), 9–17.
<http://jim.teknokrat.ac.id/index.php/JTSI>

Yefta Christian, & Hengky. (2023). Analysis Of Software Developer Perceptions Towards The Selection Of Javascript *Framework* In Batam City. *Jurnal Ilmiah Teknologi Informasi Dan Komunikasi (Jtik)*, 14, 190–200.

Yosef Jeffri Silvanus Nahak, & Hindriyanto Dwi Purnomo. (2023). Perancangan Sistem *Replikasi* Dan Sistem Backup Database Postgresql Menggunakan Repmgr Dan Barman. *Jurnal Ilmiah Komputer*, 19, 867–877.

DAFTAR LAMPIRAN

Lampiran 1 Jadwal Pengerjaan Sprint

No	Rincian Tugas		
1.	sprint 1	Sub-Tugas	Waktu (Tanggal)
	Pembuatan UML	Pembuatan <i>Use Case Diagram</i>	21 – 22 Desember 2023
		Pembuatan <i>Activity Diagram</i>	23 – 24 Desemberr 2023
		Pembuatan <i>Sequence Diagram</i>	25 – 26 Desemberr 2023
	Pembuatan skema basis data	Pembuatan <i>Entity Relationship Diagram</i>	26 – 31 Desemberr 2023
	Desain Aplikasi	Pembuatan tampilan Halaman <i>Login</i>	1 Januari 2024
		Pembuatan tampilan halaman <i>Verifikasi Email</i>	
		Pembuatan tampilan halaman <i>Reset Password</i>	2 Januari 2024
		Pembuatan tampilan halaman <i>Dashboard</i>	
		Pembuatan tampilan halaman Siswa	3 - 4 Januari 2024
		Pembuatan tampilan halaman <i>Stakeholders</i>	
		Pembuatan tampilan halaman Pencatatan	5 - 6 Januari 2024
		Pembuatan tampilan halaman Laporan Keuangan	
		Pembuatan tampilan <i>Logout</i>	
2.	sprint 2	Sub-Tugas	Waktu (Tanggal)
	<i>Login</i>	Pembuatan kode program (<i>Front-End</i>)	14 – 15 Januari 2024
		Pembuatan kode program (<i>Back-End</i>)	16 – 17 Januari 2024
		<i>Unit Testing</i>	18 Januari 2024
	<i>Verifikasi email</i>	Pembuatan kode program (<i>Front-End</i>)	19 – 20 Januari 2024
		Pembuatan kode program (<i>Back-End</i>)	21 – 22 Januari 2024
		<i>Unit Testing</i>	23 Januari 2024

No	Rincian Tugas		
	Reset Password	Pembuatan kode program (<i>Front-End</i>)	24 – 25 Januari 2024
		Pembuatan kode program (<i>Back-End</i>)	26 – 27 Januari 2024
		Unit Testing	28 Januari 2024
3.	sprint 3	Sub-Tugas	Waktu (Tanggal)
	Fitur kelola data siswa	Pembuatan kode program (<i>Front-End</i>)	1 – 2 Februari 2024
		Pembuatan kode program (<i>Back-End</i>)	3 – 6 Februari 2024
		Unit Testing	7 Februari 2024
	Fitur kelola data kelas	Pembuatan kode program (<i>Front-End</i>)	8 Februari 2024
		Pembuatan kode program (<i>Back-End</i>)	9 – 10 Februari 2024
		Unit Testing	11 Februari 2024
	Fitur kelola data jurusan	Pembuatan kode program (<i>Front-End</i>)	12 Februari 2024
		Pembuatan kode program (<i>Back-End</i>)	13– 14 Februari 2024
		Unit Testing	15 Februari 2024
4.	sprint 4	Sub-Tugas	Waktu (Tanggal)
	Fitur kelola data guru	Pembuatan kode program (<i>Front-End</i>)	16 - 17 Februari 2024
		Pembuatan kode program (<i>Back-End</i>)	18 - 21 Februari 2024
		Unit Testing	22 Februari 2024
5.	sprint 5	Sub-Tugas	Waktu (Tanggal)
	Fitur pencatatan pembayaran	Pembuatan kode program (<i>Front-End</i>)	1 - 2 Maret 2024
		Pembuatan kode program (<i>Back-End</i>)	3 – 7 Maret 2024
		Unit Testing	8 – 9 Maret 2024
	Fitur pemasukan & pengeluaran	Pembuatan kode program (<i>Front-End</i>)	10 - 11 Maret 2024
		Pembuatan kode program (<i>Back-End</i>)	12- 13 Maret 2024
		Unit Testing	14 - 15 Maret 2024

No	Rincian Tugas		
	Fitur jenis transaksi & kas gueu	Pembuatan kode program (<i>Front-End</i>)	16 Maret 2024
		Pembuatan kode program (<i>Back-End</i>)	17 - 19 Maret 2024
		<i>Unit Testing</i>	20 Maret 2024
6.	sprint 6	Sub-Tugas	Waktu (Tanggal)
	Fitur laporan keuangan	Pembuatan kode program (<i>Front-End</i>)	24 - 25 Maret 2024
		Pembuatan kode program (<i>Back-End</i>)	26 Maret – 1 April 2024
		<i>Unit Testing</i>	2 April 2024
	Fitur laporan siswa & guru	Pembuatan kode program (<i>Front-End</i>)	3 – 4 April 2024
		Pembuatan kode program (<i>Back-End</i>)	5 - 6 April 2024
		<i>Unit Testing</i>	7 April 2024
7.	sprint 7	Sub-Tugas	Waktu (Tanggal)
	<i>Dashboard</i>	Pembuatan kode program (<i>Front-End</i>)	16 - 19 April 2024
		Pembuatan kode program (<i>Back-End</i>)	20 - 23 April 2024
		<i>Unit Testing</i>	24 - 25 April 2024

Lampiran 2 Daftar Pertanyaan Wawancara

PERTANYAAN WAWANCARA

1. Bagaimana sejarah singkat SMA Plus Nurul Iman Leles?
2. Kapan sekolah ini didirikan dan oleh siapa?
3. Apa saja perubahan atau perkembangan signifikan yang telah terjadi sejak sekolah ini didirikan?
4. Apa visi dan misi SMA Plus Nurul Iman Leles?
5. Bagaimana visi dan misi ini diterapkan dalam kegiatan sehari-hari sekolah?
6. Bagaimana gambaran umum proses bisnis di SMA Plus Nurul Iman Leles?
7. Bagaimana sistem administrasi keuangan yang sedang berjalan saat ini?
8. Apa saja perangkat lunak atau alat yang digunakan dalam pengelolaan keuangan?
9. Apa saja kendala atau masalah yang sering dihadapi dalam sistem yang ada saat ini?
10. Teknologi apa saja yang saat ini digunakan dalam mendukung kegiatan operasional sekolah?
11. Apakah ada rencana untuk mengadopsi teknologi baru dalam waktu dekat?
12. Informasi keuangan apa saja yang dibutuhkan oleh manajemen sekolah?
13. Seberapa sering informasi keuangan perlu diperbarui atau diakses?
14. Siapa saja yang berhak mengakses informasi keuangan ini?
15. Data apa saja yang dikelola oleh bagian administrasi keuangan?
16. Bagaimana alur pengelolaan data dari mulai pengumpulan hingga penyimpanan?
17. Bagaimana keamanan data keuangan dijaga?
18. Apakah ada laporan keuangan khusus yang harus disiapkan secara berkala?
19. Bagaimana proses pelaporan keuangan dilakukan?
20. Adakah regulasi atau standar tertentu yang harus dipatuhi dalam pengelolaan keuangan di sekolah ini?
21. Bagaimana pengalaman pengguna (bendahara) dengan sistem saat ini?
22. Apa saja fitur yang sebenarnya diinginkan oleh pengguna?

Lampiran 3 Hasil Perancangan

1. Tampilan fitur kelas

Data Kelas

+ Tambah

cari...

Nama

▼

NO	KODE	KELAS	ACTION
1	B00XI1	XII	
2	A00X1	XI	

2. Tampilan fitur jurusan

Data Jurusan

+ Tambah

cari...

Nama

▼

NO	KODE	JURUSAN	ACTION
1	IPS001	IPS	
2	IPA001	IPA	

3. Tampilan fitur pembayaran siswa

Pembayaran Siswa X

+ Tambah

cari...

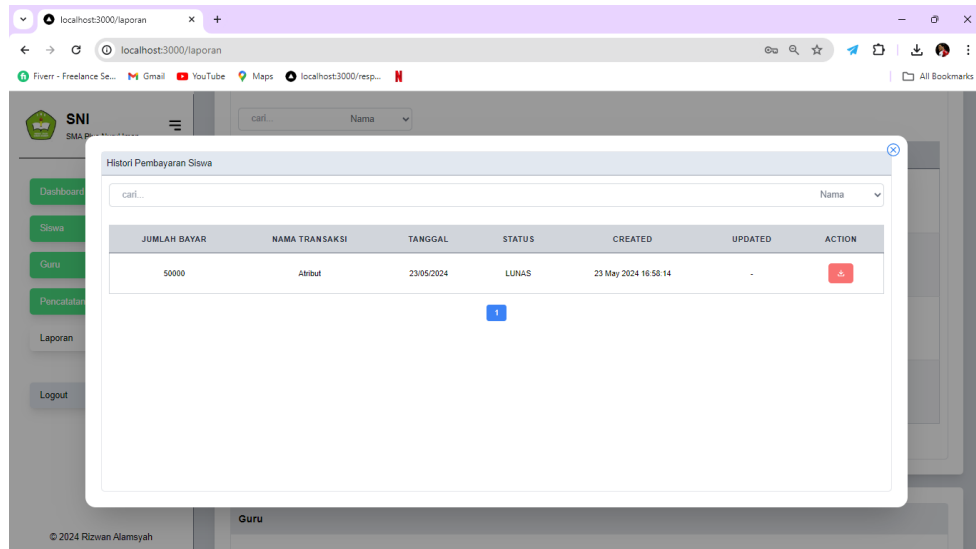
Siswa

▼

NAMA PEMBAYARAN	NAMA SISWA	TANGGAL	JUMLAH	ACTION
OSIS	Bambang	18/05/2024	Rp. 50000	
SPP	Bambang	19/05/2024	Rp. 75000	

1

4. Tampilan fitur histori pembayaran siswa



5. Tampilan kwitansi pembayaran siswa

**SMA Plus Nurul Iman Leles**

Kp. Galumpit Kidul RT 005/RW 004 Des. Cipancar Kec. Leles Garut Jawa Barat
Garut, 12 June 2024
No. Telp: 123456789

KWITANSI PEMBAYARAN

Nama : Haikal Alnawawi
NISN : 002024
Nama Pembayaran : SPP
Semester : -
Bulan : Januari

Jumlah Bayar : Rp. 80.000
Tanggal Pembayaran : 23/05/2024
Status : LUNAS
Waktu Update :

Mengetahui,

(Bendahara Sekolah)

6. Tampilan fitur pemasukan



7. Tampilan fitur transaksi

+

Tambah

cari...

Nama

▼

NAMA TRANSAKSI	JUMLAH BAYAR	KATEGORI	ACTION
LKS	Rp. 100000	A	🗑️ ✎️
LKS	Rp. 120000	B	🗑️ ✎️

8. Tampilan fitur pengeluaran

+

Tambah

cari...

Nama

▼

NAMA PENGELUARAN	TANGGAL	JUMLAH	ACTION
Pembelian Printer	10/05/2024	Rp. 200000	🗑️ ✎️
Pembelian ATK	12/06/2024	Rp. 20000	🗑️ ✎️
pembelian komputer	12/06/2024	Rp. 200000	🗑️ ✎️
pembelian kursi	12/06/2024	Rp. 150000	🗑️ ✎️

1

9. Tampilan fitur kas guru

+

Tambah

cari...

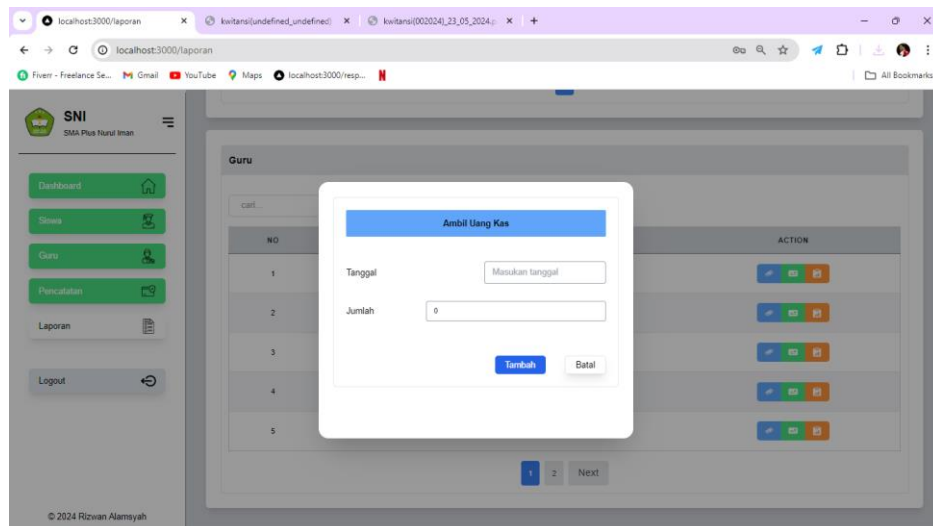
Nama

▼

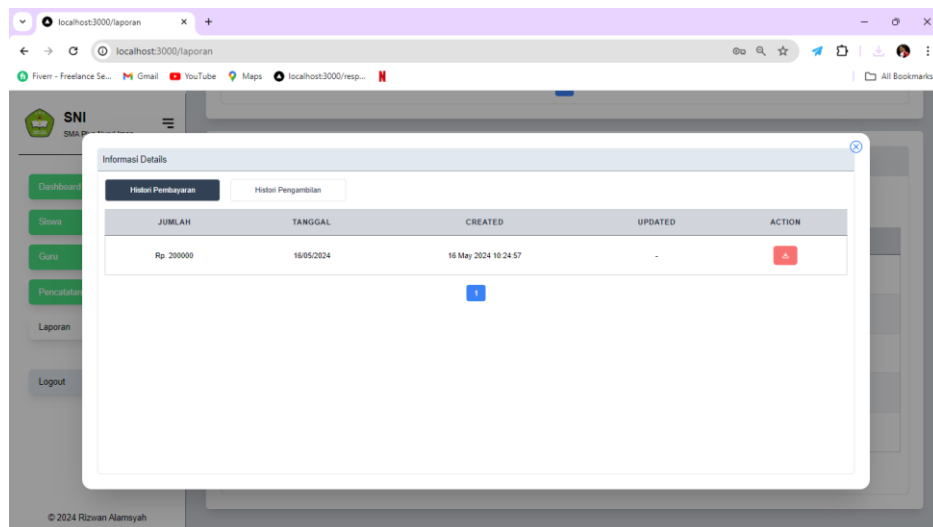
NIP	NAMA	TANGGAL	JUMLAH	ACTION
0021123	Cecep Somantri S Pd.	13/05/2024	Rp. 100000	🗑️ 📄
0021124	Zamzam Kamaludin S Pd.I.	16/05/2024	Rp. 200000	🗑️ 📄
0021123	Cecep Somantri S Pd.	22/05/2024	Rp. 50000	🗑️ 📄
0021123	Cecep Somantri S Pd.	29/05/2024	Rp. 30000	🗑️ 📄
0024	Arteta	30/05/2024	Rp. 100000	🗑️ 📄

1

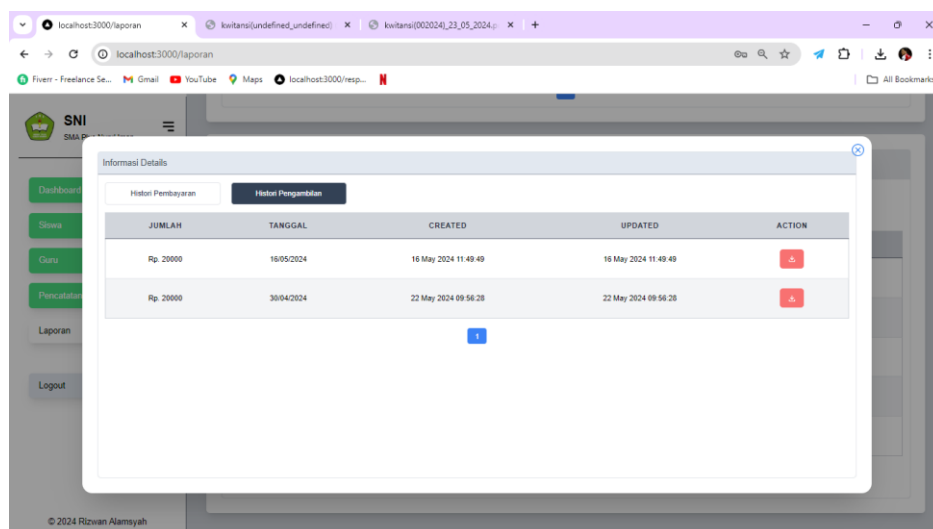
10. Tampilan fitur ambil kas guru



11. Tampilan fitur histori pembayaran kas



12. Tampilan fitur histori pengambilan kas



13. Tampilan kwitansi pembayaran kas

**SMA Plus Nurul Iman Leles**

Kp. Galumpit Kidul RT 005/RW 004 Des. Cipancar Kec. Leles Garut Jawa Barat
Garut, 12 June 2024
No. Telp: 123456789

KWITANSI PEMBAYARAN KAS

Nama: Cecep Somantri S.Pd.

Waktu Update :

NIP : 0021123

Mengetahui,

Jumlah Bayar: Rp. 100.000

Tanggal Pembayaran: 13/05/2024

(Bendahara Sekolah)

14. Tampilan kwitansi pengambilan kas

**SMA Plus Nurul Iman Leles**

Kp. Galumpit Kidul RT 005/RW 004 Des. Cipancar Kec. Leles Garut Jawa Barat
Garut, 12 June 2024
No. Telp: 123456789

KWITANSI PENGAMBILAN KAS

Nama : Zamzam Kamaludin S.Pd.i.

Waktu Update : 16 May 2024 11:49:49

NIP : 0021124

Mengetahui,

Jumlah Ambil : Rp. 20.000

Tanggal Pengambilan : 16/05/2024

(Bendahara Sekolah)

Lampiran 4 Surat Pernyataan Telah Melakukan Wawancara

SURAT PERNYATAAN TELAH MELAKUKAN WAWANCARA

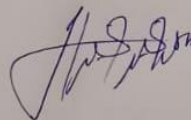
Yang bertanda tangan di bawah ini:

Nama : Sambas Hadiyansyah S.Pd.
NUPTK : 3234767668120003
Jabatan : Bendahara

Menyatakan bahwa benar mahasiswa dengan nama Rizwan Alamsyah NIM. 20576021 telah melakukan wawancara untuk penelitian dengan judul Rancang Bangun Sistem Informasi Administrasi Keuangan SMA Plus Nurul Iman Leles Berbasis *Web* Menggunakan Metode *Scrum* pada tanggal 12 Desember 2023.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya dan untuk digunakan sebagaimana mestinya.

Garut, 12 Desemberr 2024



Sambas Hadiyansyah, S.Pd.

Lampiran 5 Surat Pernyataan Telah Melakukan Pengujian Sistem**SURAT PERNYATAAN TELAH MELAKUKAN PENGUJIAN SISTEM**

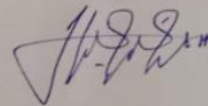
Yang bertanda tangan di bawah ini:

Nama : Sambas Hadiyansyah, S.Pd.
NUPTK : 3234767668120003
Jabatan : Bendahara

Menyatakan bahwa benar mahasiswa dengan nama Rizwan Alamsyah NIM. 20576021 telah melakukan pengujian Sistem Informasi Administrasi Keuangan SMA Plus Nurul Iman Leles pada tanggal 22 Mei 2024.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya dan untuk digunakan sebagaimana mestinya.

Garut, 12 Desemberr 2024



Sambas Hadiyansyah, S.Pd.

Lampiran 6 Dokumentasi Kegiatan Penelitian



