



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN *REST-API* SISTEM INFORMASI
AKADEMIK MENGGUNAKAN GOLANG:
STUDI KASUS DI RA. AMANAH SCHOOL**

TUGAS AKHIR

DADEN DHARMAWAN

0110220267

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024



**STT TERPADU
NURUL FIKRI**

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN *REST-API* SISTEM INFORMASI
AKADEMIK MENGGUNAKAN GOLANG:
STUDI KASUS DI RA. AMANAH SCHOOL**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

STT - NF
DADEN DHARMAWAN
0110220267

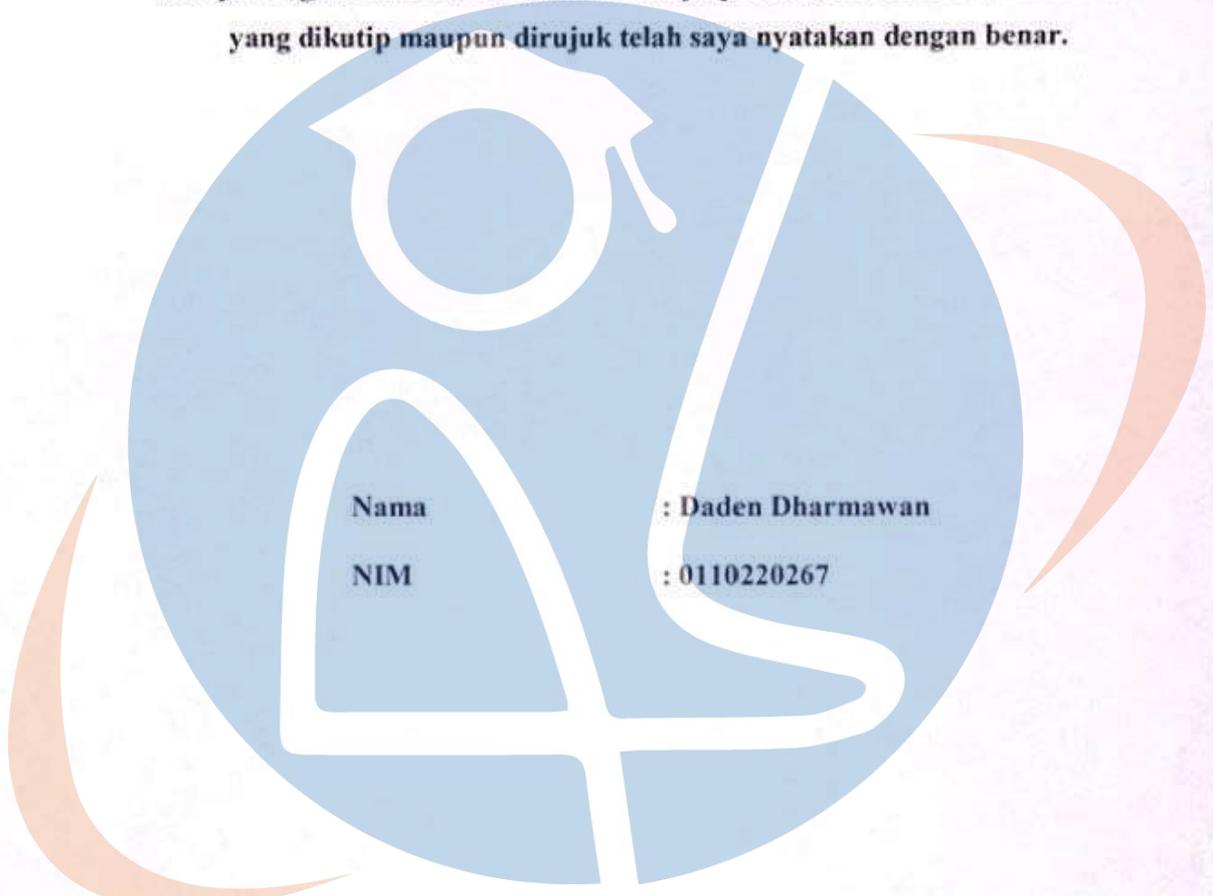
PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.



Nama : Daden Dharmawan
NIM : 0110220267

Depok, 18 April 2024

STT - NF 

Daden Dharmawan

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Daden Dharmawan

NIM : 0110220267

Program Studi : Teknik Informatika

Judul Skripsi : Rancang Bangun *REST-API* Sistem Informasi Akademik

Menggunakan Golang: Studi Kasus di RA. Amanah School

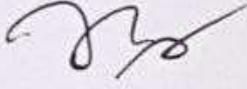
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing

Penguji


Nasrul, S.Pd.I., S.Kom., M.Kom.


Henry Saptono, S.Si., M.Kom.

Ditetapkan di : Depok

Tanggal : 29 Juli 2024

KATA PENGANTAR

Dengan penuh rasa syukur, penulis panjatkan puji dan terima kasih kehadirat Allah SWT atas limpahan berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi/tugas akhir ini dengan lancar. Penyelesaian skripsi/tugas akhir ini merupakan salah satu langkah penting dalam meraih gelar Sarjana Komputer pada Program Studi Teknik Informatika di Sekolah Tinggi Teknologi Terpadu Nurul Fikri..

Tanpa campur tangan Allah SWT dan bantuan dari berbagai pihak, mustahil bagi penulis untuk menyelesaikan karya ilmiah ini. Oleh karena itu, dengan penuh rasa hormat dan penghargaan, penulis ingin mengucapkan terima kasih kepada:

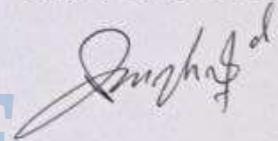
1. Allah Subhana Wa Ta'ala, atas berkat dan rahmat-Nya dalam memberikan kemudahan dan kelancaran dalam menyelesaikan tugas akhir ini.
2. Orang tua dan seluruh anggota keluarga, yang telah memberikan dukungan moril dan materiil selama penulis menjalani masa perkuliahan dan dalam penyelesaian tugas akhir ini.
3. Bapak Dr. Lukman Rosyidi, S.T., M.T., M.M., selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri, atas bimbingan dan arahnya yang berharga selama penulis menimba ilmu di Sekolah Tinggi ini.
4. Ibu Tifanny Nabarian, S.Kom., M.T.I., selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri, atas bimbingan dan arahnya selama penulis menjalani perkuliahan di Program Studi ini.
5. Bapak Zaki Imaduddin, S.T., M.Kom., selaku Dosen Pembimbing Akademik, yang telah membimbing penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Nasrul, S.Pd.I., S.Kom., M.Kom., selaku Dosen Pembimbing Tugas Akhir, yang telah dengan sabar membimbing dan mengarahkan penulis dalam penyusunan skripsi/tugas akhir ini.

7. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri, yang telah dengan ikhlas membimbing dan memberikan ilmu pengetahuan kepada penulis selama menjalani perkuliahan.
8. Muhammad Jihad Robbani dan Zulfania Shofa, sahabat tercinta yang selalu memberikan dukungan moral kepada penulis selama penyusunan laporan tugas akhir ini.
9. Shiro selaku hewan peliharaan tersayang yang selalu menemani penulis dalam penyusunan tugas akhir.

Penulis juga ingin mengucapkan terima kasih kepada semua pihak yang telah membantu penulis dalam menyelesaikan skripsi/tugas akhir ini, baik secara langsung maupun tidak langsung. Penulis menyadari bahwa skripsi/tugas akhir ini masih jauh dari sempurna, oleh karena itu penulis menerima kritik dan saran yang membangun demi perbaikan di masa depan.

Penulis berharap agar segala kebaikan yang telah diberikan oleh semua pihak yang membantu akan mendapat balasan yang baik dari Allah SWT. Semoga skripsi ini memberikan manfaat yang besar dalam pengembangan ilmu..

Depok, 18 April 2024



STT - NF

Daden Dharmawan

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Daden Dharmawan
NIM : 0110220267
Program Studi : Teknik Informatika
Jenis karya : Skripsi / Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul :

RANCANG BANGUN *REST-API* SISTEM INFORMASI AKADEMIK
MENGUNAKAN GOLANG: STUDI KASUS DI RA. AMANAH SCHOOL

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 13 Juli 2024

STT - NF

Yang Menyatakan



Daden Dharmawan

ABSTRAK

Nama : Daden Dharmawan
NIM : 0110220267
Program Studi : Teknik Informatika
Judul : Rancang Bangun *REST-API* Sistem Informasi Akademik Menggunakan Golang: Studi Kasus di RA. Amanah School.

RA. Amanah School berkomitmen untuk menyediakan pendidikan berkualitas bagi para muridnya. Akan tetapi, RA. Amanah School memiliki keterbatasan aksesibilitas informasi dan pengelolaan data yang tidak efisien. Dimana penyampaian informasi akademik, seperti presensi, nilai, dan tagihan SPP masih dilakukan secara manual, baik melalui pengumuman guru di sekolah maupun melalui WhatsApp. Rancang Bangun *REST-API* Sistem Informasi Akademik Menggunakan Golang: Studi Kasus di RA. Amanah School dirancang dan dibangun untuk memungkinkan orang tua siswa untuk memantau informasi akademik anaknya kapanpun dan dimanapun. Sistem ini diharapkan dapat meningkatkan efektivitas dan efisiensi penyampaian informasi akademik, meningkatkan transparansi dan akuntabilitas pengelolaan data sekolah, serta meningkatkan kualitas pelayanan pendidikan RA. Amanah School secara keseluruhan. Penelitian ini menggunakan bahasa pemrograman Go dengan metode *Extreme Programming* serta menggunakan *Black Box Testing* untuk menguji fungsionalitas dari sistem yang dibangun. Dari hasil pengujian yang dilakukan pada 26 skenario didapatkan skor sebesar 100% yang berarti hasil dari pengujian sesuai dengan apa yang diharapkan. Harapannya dengan adanya hasil penelitian ini dapat meningkatkan efektivitas dan efisiensi penyampaian informasi akademik, meningkatkan transparansi dan akuntabilitas pengelolaan data sekolah, serta meningkatkan kualitas pelayanan pendidikan RA. Amanah School secara keseluruhan.

Kata kunci : *Black Box Testing*, *Extreme Programming*, *Golang*, *REST-API*, Sistem Informasi Akademik



STT - NF

ABSTRACT

Name : Daden Dharmawan
NIM : 0110220267
Study Program : Informatics Engineering
Title : Design and Development of REST-API for Academic Information System Using Golang: A Case Study of RA Amanah School.

RA. Amanah School is committed to providing quality education for its students. However, RA. Amanah School has limitations in information accessibility and inefficient data management. Where the delivery of academic information, such as attendance, grades, and tuition bills, is still done manually, both through teacher announcements at school and through WhatsApp. Design and Development of REST-API Academic Information System Using Golang: A Case Study at RA. Amanah School is designed and built to allow parents of students to monitor their child's academic information anytime, anywhere. This system is expected to improve the effectiveness and efficiency of delivering academic information, increase transparency and accountability in school data management, and improve the overall quality of education services at RA. Amanah School. This study uses the Go programming language with the Extreme Programming method and uses Black Box Testing to test the functionality of the system being built. From the results of the tests carried out on 26 scenarios, a score of 100% was obtained, which means that the test results are in accordance with what is expected. It is hoped that with the results of this study, the effectiveness and efficiency of delivering academic information can be improved, transparency and accountability in school data management can be increased, and the overall quality of education services at RA. Amanah School can be improved.

Key words : Academic Information System, Black Box Testing, Extreme Programming, Golang, REST-API

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	Error! Bookmark not defined.
HALAMAN PENGESAHAN.....	Error! Bookmark not defined.
KATA PENGANTAR	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	v
ABSTRAK	vi
<i>ABSTRACT</i>	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
1.5 Batasan Masalah.....	3
1.6 Sistematia Penulisan.....	3
BAB II KAJIAN LITERATUR	5
2.1 Sistem Informasi Akademik (SIA).....	5
2.2 <i>Web Service</i>	5
2.3 <i>Representational State Transfer (REST)</i>	5
2.4 <i>Application Programming Interface (API)</i>	6
2.5 <i>Hypertext Transfer Protocol (HTTP)</i>	6
2.6 <i>HTTP Method</i>	6

2.7	Postman	7
2.8	Golang	7
2.9	MySQL.....	8
2.10	<i>Unified Modelling Language (UML)</i>	8
2.11	<i>Extreme Programming (XP)</i>	9
2.12	<i>Testing</i>	10
2.12.1	<i>Black Box Testing</i>	11
2.13	Penelitian Terkait	11
2.14	Posisi Penelitian	13
BAB III METODOLOGI PENELITIAN.....		17
3.1	Tahapan Penelitian	17
3.1.1	Studi Pendahuluan.....	18
3.1.2	Analisa Perancangan.....	18
3.1.3	Implementasi.....	18
3.1.4	Pengujian.....	18
3.1.5	Kesimpulan	18
3.2	Rancangan Penelitian	19
3.2.1	Jenis Penelitian.....	19
3.2.2	Metode Analisis Data.....	19
3.2.3	Metode Pengumpulan Data	20
3.2.4	Metode Pengujian.....	21
3.2.5	Metode Implementasi dan Evaluasi	21
3.2.6	Lingkungan Pengembangan	21
3.2.6.1	Lokasi Penelitian	22
3.2.6.2	Alat dan Bahan	22

BAB IV IMPLEMENTASI DAN EVALUASI.....	23
4.1 Analisis dan Perancangan Sistem.....	23
4.1.1 Analisis Sistem Berjalan	23
4.1.2 <i>User Story</i>	24
4.1.3 Identifikasi Aktor	26
4.1.4 <i>Product Backlog</i>	27
4.1.5 <i>Use Case Diagram</i>	28
4.1.4.1 <i>Use Case Diagram</i> Mengelola Data Akademik.....	30
4.1.4.2 <i>Use Case Diagram</i> Mengelola Data <i>User</i>	31
4.1.6 <i>Class Diagram</i>	32
4.1.7 <i>Activity Diagram</i>	34
4.1.8 Rancangan Pengujian.....	43
4.2 Implementasi Bahasa Pemrograman Go	55
4.2.1 <i>Middleware</i>	55
4.2.2 <i>Error Handler</i>	55
4.3 Implementasi <i>API</i> Sistem Informasi Akademik RA Amanah School	56
4.3.1 Antarmuka Kelola Data Guru	56
4.3.2 Antarmuka Kelola Data Siswa.....	59
4.3.3 Antarmuka Kelola Data Presensi	63
4.3.4 Antarmuka Kelola Data Penilaian.....	66
4.3.5 Antarmuka Kelola Data Tagihan SPP.....	68
4.3.6 Antarmuka Kelola Data Dengan Akun <i>Unauthorized</i>	70
4.4 Pengujian Sistem.....	74
4.5 Evaluasi Hasil Pengujian.....	87
BAB V KESIMPULAN DAN SARAN.....	88

5.1 Kesimpulan	88
5.2 Saran.....	88
DAFTAR PUSTAKA	90
LAMPIRAN.....	92



STT - NF

DAFTAR GAMBAR

Gambar 2. 1 Alur Metode Extreme Programming.....	10
Gambar 3. 1 Tahapan Penelitian	17
Gambar 4. 1 Alur Informasi Akademik RA. Amanah School	23
Gambar 4. 2 Use Case Diagram.....	29
Gambar 4. 3 Use Case Diagram Mengelola Data Akademik.....	30
Gambar 4. 4 Use Case Diagram Mengelola Data User.....	31
Gambar 4. 5 Class Diagram	33
Gambar 4. 6 Activity Diagram Mengelola Data Guru	34
Gambar 4. 7 Activity Diagram Mengelola Data Siswa.....	36
Gambar 4. 8 Activity Diagram Mengelola Data Presensi.....	37
Gambar 4. 9 Activity Diagram Mengelola Data Nilai	38
Gambar 4. 10 Activity Diagram Mengelola Data Tagihan SPP	39
Gambar 4. 11 Activity Diagram Wali Murid Melihat Data Presensi.....	40
Gambar 4. 12 Activity Diagram Wali Murid Melihat Data Nilai	41
Gambar 4. 13 Activity Diagram Wali Murid Melihat Data Tagihan SPP	42
Gambar 4. 14 Middleware.....	55
Gambar 4. 15 Error Handler.....	56
Gambar 4. 16 Input Data Guru Dengan Benar.....	57
Gambar 4. 17 Akun User Teacher.....	57
Gambar 4. 18 Input Data Guru Dengan Salah	58
Gambar 4. 19 Mengubah Data Guru	58
Gambar 4. 20 Menampilkan Seluruh Data Guru	59
Gambar 4. 21 Input Data Siswa Dengan Benar	60
Gambar 4. 22 Akun User Student	60
Gambar 4. 23 Input Data Siswa Dengan Salah	61
Gambar 4. 24 Input Data Siswa Lebih Dari Satu Menggunakan File CSV	61
Gambar 4. 25 Data Siswa.....	62
Gambar 4. 26 Mengubah Data Siswa.....	62
Gambar 4. 27 Menampilkan Seluruh Data Siswa Berdasarkan Kelompok Kelas	63

Gambar 4. 28 Input Data Presensi Dengan Benar.....	64
Gambar 4. 29 Input Data Kehadiran Dengan Salah.....	64
Gambar 4. 30 Daftar Presensi	65
Gambar 4. 31 Mengubah Data Presensi.....	65
Gambar 4. 32 Menampilkan Data Presensi Berdasarkan Id dan Periode	65
Gambar 4. 33 Input Data Nilai Dengan Benar	66
Gambar 4. 34 Input Data Presensi Dengan Salah	67
Gambar 4. 35 Data Nilai	67
Gambar 4. 36 Mengubah Data Nilai	67
Gambar 4. 37 Menampilkan Data Penilaian Berdasarkan Id dan Periode	68
Gambar 4. 38 Input Data Pembayaran Tagihan SPP Dengan Benar	68
Gambar 4. 39 Input Data Pembayaran Tagihan SPP Dengan Salah	69
Gambar 4. 40 Mengubah Data Pembayaran	69
Gambar 4. 41 Menampilkan Data Tagihan SPP Berdasarkan Id dan Periode	70
Gambar 4. 42 Login Dengan Akun Guru.....	71
Gambar 4. 43 Input Data Guru Dengan Akun Guru	71
Gambar 4. 44 Login Dengan Akun Siswa	72
Gambar 4. 45 Input Data Guru Dengan Akun Siswa.....	72
Gambar 4. 46 Input Data Penilaian Dengan Akun Siswa	73
Gambar 4. 47 Input Data Kehadiran Dengan Akun Siswa	73
Gambar 4. 48 Input Data Tagihan SPP Dengan Akun Siswa	74
Gambar 4. 49 Input Data Siswa Dengan Akun Siswa	74

STT - NF

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	11
Tabel 2. 2 Posisi Penelitian	13
Tabel 4. 1 User Story	24
Tabel 4. 2 Identifikasi Aktor	26
Tabel 4. 3 Product Backlog	27
Tabel 4. 4 Rancangan Pengujian - Kelola Data Guru	43
Tabel 4. 5 Rancangan Pengujian - Kelola Data Siswa	45
Tabel 4. 6 Rancangan Pengujian - Kelola Data Kehadiran	47
Tabel 4. 7 Rancangan Pengujian - Kelola Data Penilaian	49
Tabel 4. 8 Rancangan Pengujian - Kelola Data Tagihan SPP	51
Tabel 4. 9 Rancangan Pengujian - Kelola Data Dengan Akun Unauthorized	53
Tabel 4. 10 Black Box Testing - Kelola Data Guru	75
Tabel 4. 11 Black Box Testing - Kelola Data Siswa	77
Tabel 4. 12 Black Box Testing - Kelola Data Kehadiran	79
Tabel 4. 13 Black Box Testing - Kelola Data Penilaian	81
Tabel 4. 14 Black Box Testing - Kelola Data Tagihan SPP	83
Tabel 4. 15 Black Box Testing - Kelola Data Dengan Akun Unauthorized	85

STT - NF

BAB I PENDAHULUAN

1.1 Latar belakang

Perkembangan teknologi yang sangat cepat memungkinkan kita untuk melakukan banyak hal yang mudah dalam kehidupan modern. Salah satu contohnya adalah kemajuan teknologi dalam bidang pendidikan[1]. Teknologi informasi merupakan contoh nyata dari kemajuan teknologi yang telah dimanfaatkan secara luas dalam mengelola berbagai macam data, seperti pengumpulan, penyimpanan, pengiriman, dan manipulasi data untuk menghasilkan informasi yang bermanfaat bagi pengguna. Dengan demikian, teknologi informasi telah menjadi tulang punggung dalam berbagai bidang, mulai dari bisnis hingga pendidikan, dengan memberikan kemampuan untuk membuat keputusan yang lebih baik berdasarkan informasi yang akurat dan relevan. Adapun salah satu implementasi dari teknologi informasi dalam dunia pendidikan adalah sebagai alat penyedia informasi pendidikan, atau lebih dikenal dengan Sistem Informasi Akademik (SIA).

Raudhatul Athfal (RA) Amanah School adalah lembaga pendidikan yang bertekad untuk menyediakan pendidikan unggul bagi murid-muridnya. Namun, pada saat ini, penyampaian informasi akademik yang meliputi presensi siswa, nilai, dan juga tagihan sumbangan pembinaan pendidikan (SPP) masih dilakukan secara manual, baik melalui pengumuman langsung dari guru di sekolah maupun melalui WhatsApp. Metode penyampaian informasi yang masih manual tersebut menyebabkan adanya keterbatasan dalam aksesibilitas informasi dan pengelolaan data yang efisien. Dampaknya, proses administrasi sekolah dapat terhambat, mengakibatkan potensi terjadinya kesalahan, keterlambatan, dan ketidakjelasan informasi yang dapat mempengaruhi kualitas pelayanan pendidikan yang diberikan oleh sekolah secara keseluruhan.

Oleh karena itu, berdasarkan masalah yang telah diuraikan di atas, penelitian ini, yang berjudul "Rancang Bangun *REST-API* Sistem Informasi Akademik Menggunakan Golang: Studi Kasus di RA. Amanah School" dilakukan dengan harapan dapat menyediakan solusi untuk permasalahan tersebut. Dengan adanya

sistem informasi akademik, proses penyampaian informasi akademik oleh guru diharapkan menjadi lebih efisien, sehingga informasi akademik dapat diakses dengan mudah di mana pun dan kapan pun. Dengan demikian, diharapkan bahwa penelitian ini akan memberikan kontribusi positif dalam meningkatkan efisiensi dan efektivitas pengelolaan informasi akademik di RA. Amanah School.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, dapat dirumuskan beberapa pokok permasalahan, yaitu:

1. Bagaimanakah rancangan *REST-API* Sistem Informasi Akademik RA. Amanah School agar dapat berjalan dengan baik menggunakan Golang?
2. Apakah *REST-API* yang dibangun sudah berjalan dengan baik?

1.3 Tujuan Penelitian

Setelah memahami latar belakang dan rumusan masalah, diharapkan penelitian ini dapat meningkatkan pemahaman tentang penyediaan informasi yang dapat diakses dengan mudah di mana saja dan kapan saja. Adapun tujuan dan manfaat penelitian ini sebagai berikut:

1. Merancang dan membangun *REST-API* sistem informasi akademik untuk RA. Amanah School dengan baik, dengan menggunakan Golang.
2. Menguji *REST-API* apakah sudah berjalan dengan baik.

1.4 Manfaat Penelitian

Berdasarkan tujuan penelitian yang telah dijelaskan, diharapkan penelitian ini dapat memberikan hasil yang bermanfaat sebagai berikut:

1. Memungkinkan sistem lain untuk mengakses informasi akademik kapan saja dan di mana saja secara lebih mudah dan cepat.
2. Meningkatkan transparansi informasi antara sekolah, guru, dan orang tua.
3. Memberikan pengalaman yang lebih baik dalam mengakses informasi akademik, meningkatkan kepuasan dan loyalitas terhadap sistem informasi akademik yang digunakan.

1.5 Batasan Masalah

Batasan masalah ini bertujuan untuk memandu penelitian agar tetap terfokus pada pengembangan *REST-API* untuk Sistem Informasi Akademik RA. Amanah School. Batasan masalah tersebut mencakup:

1. Fitur yang dikembangkan tidak mencakup pemulihan kata sandi.
2. Pengujian sistem hanya dilakukan dengan Postman.
3. Sistem pembayaran masih dengan cara pembayaran konvensional, tidak menggunakan *payment gateway*.

1.6 Sistematia Penulisan

Dalam penelitian ini, metode penulisan berikut digunakan untuk mempermudah pemahaman tugas akhir secara keseluruhan:

BAB I PENDAHULUAN

Merupakan bab pembuka yang memberikan penjelasan tentang latar belakang penelitian, rumusan masalah, tujuan dan manfaat penelitian untuk menjawab rumusan masalah, batasan masalah, serta sistematika penulisan yang merupakan bagian untuk menjelaskan struktur kerangka dalam penyajian laporan penelitian ini.

BAB II KAJIAN LITERATUR

Bab ini menguraikan konsep dan teori yang berkaitan dengan penelitian, mengevaluasi temuan dari penelitian terkait, dan mengidentifikasi perbedaan penelitian ini dengan studi sebelumnya yang telah dilakukan oleh para ahli.

BAB III METODOLOGI PENELITIAN

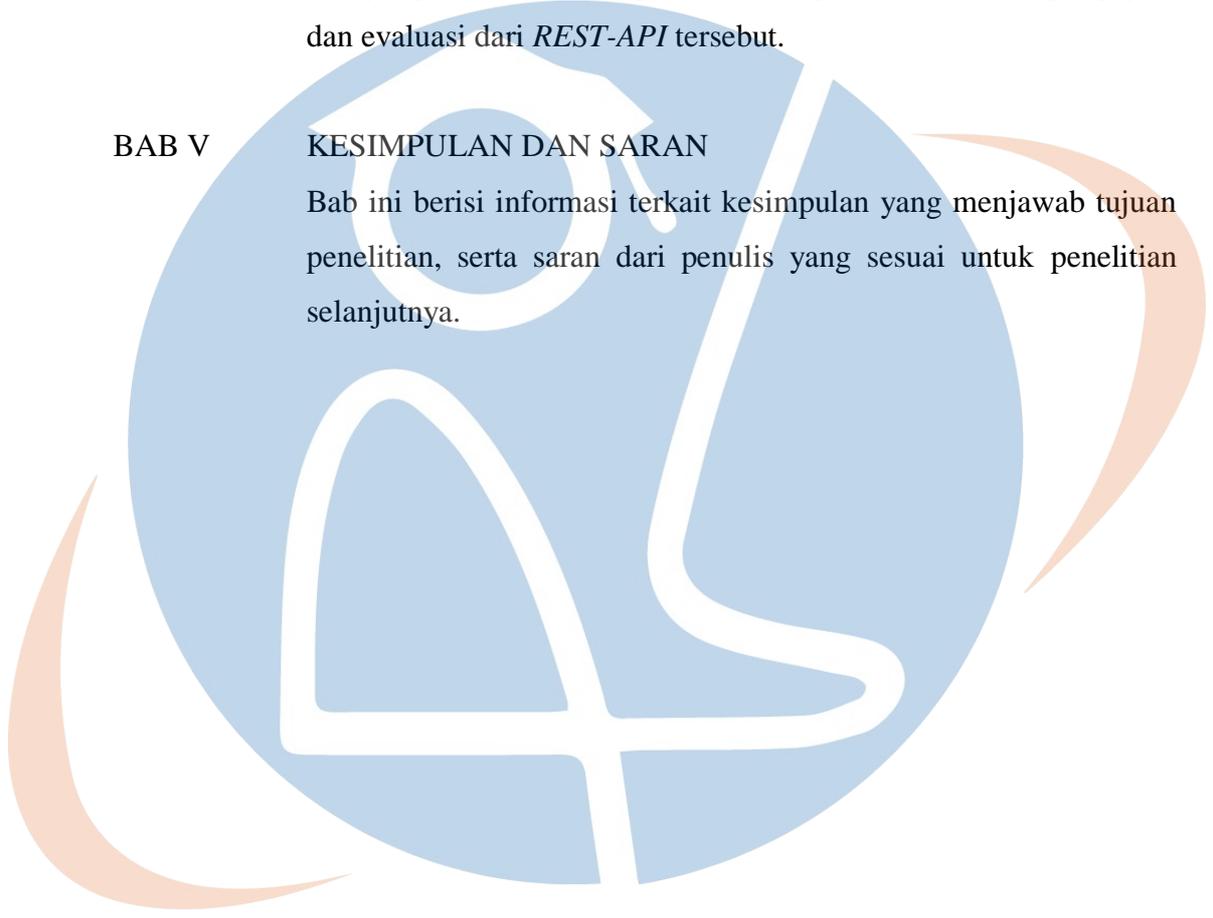
Bab ini menguraikan tahapan penelitian, mulai dari alur penelitian dan rancangan penelitian. Sebagai langkah akhir, rancangan pengujian akan menggunakan metode *black box*.

BAB IV IMPLEMENTASI DAN EVALUASI

Bab ini akan membahas perancangan *REST-API* untuk sistem informasi akademik menggunakan Golang: studi kasus di RA. Amanah School. Bab ini mencakup hasil dari perancangan *REST-API* yang dilakukan oleh penulis, serta pembahasan hasil pengujian dan evaluasi dari *REST-API* tersebut.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi informasi terkait kesimpulan yang menjawab tujuan penelitian, serta saran dari penulis yang sesuai untuk penelitian selanjutnya.



STT - NF

BAB II KAJIAN LITERATUR

2.1 Sistem Informasi Akademik (SIA)

Sistem Informasi Akademik adalah sistem informasi yang dirancang untuk memudahkan pengelolaan dan presentasi data akademik. Pengguna sistem ini meliputi seluruh elemen sekolah, seperti kepala sekolah, guru, staf, siswa, dan wali siswa. Sistem Informasi Akademik terdiri dari komponen-komponen data siswa, data guru, dan lainnya yang saling terhubung untuk membentuk sebuah sistem[2].

2.2 Web Service

Web service adalah serangkaian protokol dan standar terbuka yang memfasilitasi pertukaran informasi antara aplikasi atau sistem yang berbeda. Ini menyediakan sarana untuk koneksi antar aplikasi guna berkomunikasi, berbagi data, dan layanan lainnya. Dengan *web service*, interaksi antara perangkat, sistem, atau aplikasi yang menggunakan bahasa yang berbeda menjadi mungkin, sehingga memungkinkan pertukaran informasi dan menjalankan fungsionalitas tertentu[3]. *Web service* bisa diinterpretasikan sebagai cara untuk bertukar data tanpa memperhatikan di mana basis data disimpan, bahasa pemrograman yang digunakan dalam aplikasi yang mengakses data, atau platform di mana data tersebut dikonsumsi. *Web service* mendukung interoperabilitas, menjadikannya sebagai jembatan penghubung antara berbagai sistem yang berbeda[4].

2.3 Representational State Transfer (REST)

REST adalah kumpulan prinsip arsitektural yang memungkinkan pengiriman data melalui antarmuka yang terstandarisasi seperti *HTTP*. *REST* berfungsi seperti aplikasi website biasa. Dengan menggunakan protokol *HTTP*, *client* dapat mengirimkan permintaan kepada *server*, yang kemudian akan memberikan tanggapan kepada klien. Roy Fielding, *co-founder* proyek *Apache HTTP Server*, mengembangkan *REST*. Dalam arsitekturnya sendiri, *REST server* menyediakan sumber daya atau data, dan klien *REST* dapat mengakses dan menampilkan sumber

daya tersebut untuk penggunaan berikutnya. Sumber daya diidentifikasi oleh *Universal Resource Identifiers (URIs)* atau global ID, dan dapat diwakili dalam format teks, *JSON*, atau *XML*[5].

2.4 *Application Programming Interface (API)*

API adalah antarmuka yang memungkinkan seseorang mengakses aplikasi atau layanan dalam program. *API* memungkinkan pengembang untuk menggunakan fungsi yang sudah ada dalam aplikasi lain tanpa perlu membuat aplikasi baru. *API* digunakan untuk memanggil fungsi melalui *Hyper Text Transfer Protocol (HTTP)* dan menerima tanggapan berupa *Extensible Markup Language (XML)* atau *JavaScript Object Notation (JSON)*. *API* digunakan sebagian besar untuk berbagi data antara berbagai aplikasi. *API* juga mempercepat proses pengembangan aplikasi dengan memberi fungsi yang berbeda, sehingga pengembang tidak perlu membuat fitur yang sama berulang kali. *API* yang beroperasi di tingkat sistem operasi membantu aplikasi berkomunikasi dengan lapisan dasar dan satu sama lain dengan mengikuti berbagai protokol dan spesifikasi yang telah disesuaikan[5].

2.5 *Hypertext Transfer Protocol (HTTP)*

HTTP adalah protokol jaringan lapisan aplikasi yang digunakan untuk mengirimkan sistem informasi secara bersamaan dan menggunakan hipermedia. Banyak digunakan untuk mengumpulkan sumber daya yang saling terhubung dengan tautan, atau dokumen hiperteks, yang diciptakan oleh fisikawan Inggris Tim Berners-Lee pada tahun 1990 dan membentuk *World Wide Web*. Protokol *HTTP* saat ini memiliki dua versi utama: *HTTP/1.0*, yang memerlukan koneksi yang berbeda untuk setiap dokumen, dan *HTTP/1.1*, yang memerlukan koneksi yang sama untuk setiap dokumen[6].

2.6 *HTTP Method*

HTTP method digunakan untuk menandai aksi yang diminta oleh klien *API* pada suatu sumber daya. Setiap metode *HTTP* memiliki hubungan dengan operasi

khusus, seperti membuat (*create*), membaca (*read*), memperbarui (*update*), atau menghapus (*delete*) sumber daya. Menyertakan metode *HTTP* dalam setiap permintaan ke *REST-API* sangatlah penting[7]. Adapun *method* yang sering digunakan, antara lain:

1. *GET*: Metode ini digunakan untuk meminta data dari sumber daya tertentu. Biasanya, data yang diminta dapat disertakan dalam *URL*. Metode ini bersifat idempoten, yang berarti melakukan permintaan yang sama beberapa kali tidak akan mengubah hasil.
2. *POST*: Metode ini digunakan untuk mengirim data ke *server* untuk membuat sumber daya baru. Data yang dikirimkan dengan metode ini dikirim dalam *body request HTTP*. *POST* tidak bersifat idempoten, karena setiap permintaan biasanya dapat menghasilkan perubahan di *server*.
3. *PUT*: Metode ini digunakan untuk memperbarui sumber daya atau membuat sumber daya baru jika tidak ada yang ada. Seperti *POST*, data yang dikirimkan dengan metode ini dikirim dalam badan permintaan *HTTP*. *PUT* bersifat idempoten, artinya melakukan permintaan yang sama beberapa kali akan menghasilkan hasil yang sama.
4. *DELETE*: Metode ini digunakan untuk menghapus sumber daya yang ditentukan.

2.7 Postman

Postman adalah sebuah aplikasi yang berperan sebagai *REST client* untuk menguji *REST-API*. Biasanya, *developer* pembuat *API* menggunakan Postman sebagai alat untuk menguji *API* yang telah dibuat. Postman menjadi sebuah alat yang penting dalam proses pengembangan *API*, karena telah dilengkapi dengan berbagai fitur yang sangat membantu dalam proses tersebut[8].

2.8 Golang

Bahasa pemrograman Go, juga dikenal sebagai Golang, dibuat oleh Google bersama dengan Ken Thompson, Robert Griesemer, dan Rob Pike pada tahun 2009. Tujuan dari pengembangan bahasa ini adalah untuk menghasilkan bahasa yang

lebih cepat, andal, skalable, dan sederhana. Golang adalah evolusi dari bahasa pemrograman C modern, dan dapat diketik secara statis dan dikompilasi mesin. Bahasa Go dapat digunakan untuk banyak aplikasi, situs web, dan perangkat lunak lainnya.[9].

2.9 MySQL

MySQL adalah salah satu jenis sistem manajemen basis data relasional atau dikenal sebagai *Relational Database Management System (RDBMS)* dalam bahasa Inggris yang bersifat *open-source* dan menggunakan bahasa SQL. Sistem ini beroperasi dengan model *client-server*. Jika *Database Management System (DBMS)* merujuk pada sistem manajemen basis data secara umum, *RDBMS* adalah perangkat lunak yang mengelola basis data berdasarkan model relasional. Data warehousing, yang melibatkan pengumpulan data dari berbagai sumber, *e-commerce*, dan aplikasi *logging*, menggunakan *MySQL*. Perusahaan asal Swedia MySQL AB mengembangkan *MySQL* pertama kali pada tahun 1994. Setelah membeli MySQL AB pada tahun 2008, Sun Microsystems, perusahaan teknologi Amerika Serikat, mengambil alih hak kepemilikan *MySQL* sepenuhnya. Pada tahun 2010, Oracle, salah satu perusahaan teknologi terbesar di Amerika Serikat, mengakuisisi Sun Microsystems, sehingga *MySQL* sepenuhnya dimiliki oleh Oracle.[10]

2.10 Unified Modelling Language (UML)

UML adalah model yang digunakan dalam pengembangan perangkat lunak berbasis objek. *UML* juga menetapkan standar untuk membuat *blueprint* sistem, termasuk konsep proses bisnis, penulisan kelas dalam bahasa pemrograman tertentu, skema *database*, dan komponen yang diperlukan dalam perangkat lunak[11]. Contoh model *UML* yang sering digunakan sebagai berikut:

1. Use Case Diagram

Use Case Diagram menggambarkan interaksi antara aktor-aktor eksternal dan sistem untuk mencapai tujuan tertentu. Meskipun diagram ini menggambarkan

urutan kegiatan, fokusnya adalah pada apa yang dilakukan oleh aktor dan sistem, bukan bagaimana kegiatan tersebut dilakukan.

2. *Activity Diagram*

Activity Diagram menggambarkan urutan interaksi antara objek dalam suatu skenario. Diagram ini digunakan untuk menggambarkan bagaimana objek-objek berinteraksi dalam suatu proses atau aktivitas, mulai dari permulaan hingga akhir. Selain untuk menggambarkan aktivitas dalam suatu operasi, diagram urutan juga dapat digunakan untuk menggambarkan interaksi dalam *use case* atau interaksi lainnya dalam sistem.

3. *Sequence Diagram*

Sequence Diagram mengilustrasikan hubungan interaksi antara objek-objek di dalam sistem serta objek-objek di sekitarnya melalui pesan-pesan yang ditransmisikan sepanjang waktu. Diagram ini memanfaatkan dimensi vertikal untuk merepresentasikan waktu dan dimensi horizontal untuk menunjukkan objek-objek yang terlibat dalam interaksi tersebut.

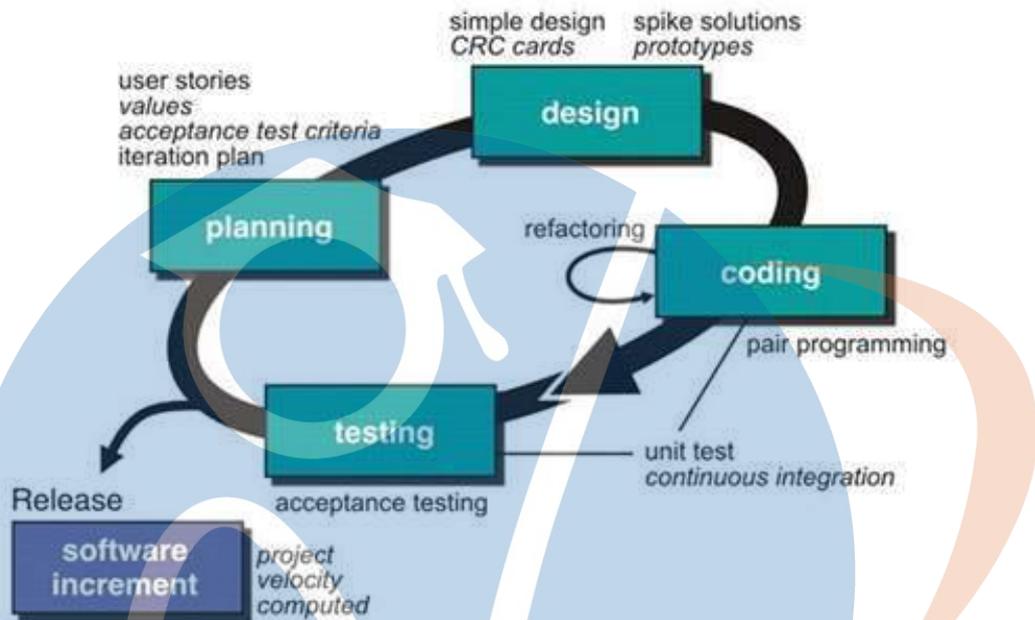
4. *Class Diagram*

Class Diagram merepresentasikan entitas yang dikelola oleh sistem. Setiap *class* dapat terhubung satu sama lain melalui berbagai jenis hubungan, seperti *associated* (terkait satu sama lain), *dependent* (satu *class* menggunakan yang lain), *specialized* (satu *class* adalah turunan dari yang lain), atau *package* (mereka dikelompokkan bersama sebagai satu unit). Sebuah sistem biasanya terdiri dari beberapa *class diagram* yang saling terkait[12].

2.11 *Extreme Programming (XP)*

XP adalah salah satu pendekatan fleksibel yang sering digunakan dalam pembuatan perangkat lunak yang sederhana. Awalnya diperkenalkan oleh Ken Beck, Ron Jeffries, dan Ward Cunningham, *XP* adalah bagian dari kerangka metodologi *Agile* yang mengutamakan kemampuan untuk menanggapi perubahan dengan cepat. Dalam *XP*, pengembangan perangkat lunak dilakukan dalam iterasi pendek, memungkinkan tim pengembang untuk secara efektif menyesuaikan diri dengan perubahan dalam kebutuhan atau keadaan proyek. Metode ini berakar pada

prinsip-prinsip yang sama dengan metodologi *Agile* lainnya, yang mengutamakan kolaborasi, adaptasi, dan responsif terhadap perubahan[13].



Gambar 2. 1 Alur Metode Extreme Programming[14]

Tahapan *Extreme Programming (XP)* terdiri dari perencanaan, desain, pengkodean, dan pengujian yang dilakukan dalam iterasi yang singkat, seperti yang digambarkan pada Gambar 2.1. Tim *XP* berkonsentrasi pada pembuatan perangkat lunak berkualitas tinggi yang sesuai dengan kebutuhan pelanggan dan responsif terhadap perubahan.

2.12 Testing

Testing adalah proses di mana suatu program dianalisis melalui berbagai tes yang dilakukan secara intensif dengan menjalankan program tersebut. Setelah menjalankan program, penguji akan memonitor dan mengevaluasi kinerja program tersebut, untuk menentukan apakah program tersebut berfungsi dengan baik atau masih memiliki kesalahan. Hasil pemantauan selama proses *testing* akan digunakan untuk evaluasi. Selanjutnya, berdasarkan hasil evaluasi, program akan diperbaiki dan diperbarui agar dapat berjalan dengan baik dan semakin baik di masa mendatang, sehingga dapat digunakan dengan lancar oleh pengguna[15].

2.12.1 Black Box Testing

Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada spesifikasi fungsionalnya. Dalam metode ini, penguji tidak perlu mengetahui detail kode program, melainkan hanya berfokus pada *input* dan *output* yang dihasilkan. Proses *black box testing* dilakukan dengan cara memberikan *input* pada berbagai formulir dalam program dan melihat *output* yang dihasilkan. Tujuannya adalah untuk memastikan bahwa program berjalan sesuai dengan spesifikasi fungsional yang telah ditentukan dan memenuhi kebutuhan pengguna. *Black Box Testing* merupakan metode pengujian yang penting untuk menjamin kualitas perangkat lunak dan meminimalisir *bug*. Dengan melakukan *Black Box Testing*, penguji dapat memastikan bahwa program yang mereka gunakan berfungsi dengan baik dan sesuai dengan kebutuhan pengguna[16].

2.13 Penelitian Terkait

Dalam penelitian ini, kajian literatur dilakukan untuk mengeksplorasi penelitian sebelumnya yang terkait dengan isu-isu yang menjadi fokus penelitian saat ini. Langkah ini bertujuan untuk mengevaluasi sejauh mana kebutuhan dalam menyelesaikan masalah yang sedang diselidiki, serta untuk memahami posisi penelitian yang telah dilakukan sebelumnya. Informasi tentang penelitian terkait dapat ditemukan dalam tabel yang disajikan di bawah ini:

Tabel 2. 1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
1	Galang Prang, Olivia Liando, Alfrina Mewengkang 2023	Sistem Informasi Pencapaian Indikator Kinerja Utama 6 di Universitas Negeri Manado	Sistem Informasi	<i>Civitas</i> akademik Universitas Negeri Manado dan masyarakat umum	Sitem indikator kinerja utama untuk mengelola dan memantau data kerjasama program pendidikan yang ada di Universitas Negeri Manado.

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
2	Sigit Priadi, Moh Idris 2023	Implementasi <i>REST</i> Dalam Membangun <i>Web Service</i> Menggunakan Golang (Studi Kasus: Modul Berita Aplikasi SATUDIKTI)	Perancangan <i>REST-API</i> untuk modul berita	Masyarakat umum	Sebuah layanan web yang menawarkan antarmuka pemrograman aplikasi (<i>API</i>) yang mengirimkan data berita sehingga aplikasi Satudikti dapat memanfaatkan <i>API</i> tersebut untuk menampilkan daftar berita.
3	Gedmi Sari Melati Rambe, Rita Novita Sari 2023	Rancang Bangun Pengenalan Sistem Informasi Sekolah Menengah Pertama (SMP) Bina Taruna Medan Berbasis Android Dengan Metode <i>Rest API</i> (<i>Application Programming Interface</i>)	Aplikasi sistem informasi akademik berbasis android	<i>Civitas</i> akademik SMP Bina Taruna Medan	Aplikasi Sistem Informasi Sekolah Menengah Pertama (SMP) Bina Taruna Medan.
4	Lulu Agustin 2023	Pengembangan <i>Microservice Core</i> Pada <i>Web Portal</i> HiColleagues Dengan <i>Framework</i> Echo Golang	Pengembangan <i>Microservice</i>	HiColleagues	Berhasil mengembangkan <i>microservice core</i> pada aplikasi berbasis web HiColleagues v2.0 dengan menggunakan <i>framework</i> Echo dan basis data PostgreSQL.

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
5	Riska Aryanti, Eka Fitriani, Dian Ardiansyah, Atang Saepudin 2021	Penerapan Metode <i>Rapid Application Development</i> Dalam Pengembangan Sistem Informasi Akademik Berbasis Web	Sistem Informasi Akademik	<i>Civitas</i> akademik SMA Panca Moral Cikampek	Aplikasi sistem informasi akademik berbasis web untuk memudahkan perolehan informasi yang akurat.

Dari Tabel 2.1 Penelitian Terkait, terlihat bahwa telah ada beberapa penelitian sebelumnya yang membahas mengenai perancangan *REST-API* untuk Sistem Informasi Akademik. Penelitian ini mengacu pada studi-studi sebelumnya yang relevan dengan topik yang sedang diteliti, sehingga dapat memberikan dukungan dan konteks untuk penelitian yang sedang dilakukan.

2.14 Posisi Penelitian

Tabel 2. 2 Posisi Penelitian

No	Nama dan Tahun	Penelitian	Rancang Bangun <i>REST-API</i>	Perancangan Aplikasi SIA	Metode <i>XP</i>	Golang
1	Galang Prang, Olivia Liando, Alfrina Mewengkang 2023	Sistem Informasi Pencapaian Indikator Kinerja Utama 6 di Universitas Negeri Manado	✓		✓	✓

No	Nama dan Tahun	Penelitian	Rancang Bangun <i>REST-API</i>	Perancangan Aplikasi SIA	Metode <i>XP</i>	Golang
2	Sigit Priadi, Moh Idris 2023	Implementasi <i>REST</i> Dalam Membangun <i>Web Service</i> Menggunakan Golang (Studi Kasus: Modul Berita Aplikasi SATUDIkti)	✓			✓
3	Gedmi Sari Melati Rambe, Rita Novita Sari 2023	Rancang Bangun Pengenalan Sistem Informasi Sekolah Menengah Pertama (SMP) Bina Taruna Medan Berbasis Android Dengan Metode <i>Rest API</i> (<i>Application Programming Interface</i>)	✓	✓		
4	Lulu Agustin 2023	Pengembangan <i>Microservice Core</i> Pada <i>Web Portal</i> HiColleagues Dengan <i>Framework</i> Echo Golang	✓			✓
5	Riska Aryanti, Eka Fitriani, Dian Ardiansyah, Atang Saepudin 2021	Penerapan Metode <i>Rapid Application Development</i> Dalam Pengembangan Sistem Informasi Akademik Berbasis Web		✓		

No	Nama dan Tahun	Penelitian	Rancang Bangun <i>REST-API</i>	Perancangan Aplikasi SIA	Metode <i>XP</i>	Golang
6	Daden Dharmawan 2024	Rancang Bangun <i>REST-API</i> Sistem Informasi Akademik Menggunakan Golang: Studi Kasus di RA. Amanah School	✓	✓	✓	✓

Pada Tabel 2.2 Posisi Penelitian, terlihat bahwa beberapa penelitian memiliki persamaan dalam penggunaan teknologi dan metode pengembangan sistem informasi. Berikut adalah penjelasan lebih rinci mengenai masing-masing penelitian:

1. Galang Prang, Olivia Liando, dan Alfrina Mewengkang mereka melakukan studi tentang penerapan sistem informasi untuk mencapai indikator kinerja utama 6 di Universitas Negeri Manado. Mereka memilih menggunakan metode *Extreme Programming (XP)* dalam pengembangan sistem ini dan menggunakan Golang untuk mengembangkan *REST-API*-nya. Hal ini menunjukkan bahwa mereka fokus pada pengembangan sistem informasi akademik dengan pendekatan *XP* dan menggunakan Golang.
2. Sigit Priadi dan Moh Idris melakukan penelitian terkait penerapan *REST-API* dalam pembangunan layanan web menggunakan Golang, namun tanpa menerapkan metode *Extreme Programming*. Ini menunjukkan bahwa fokus penelitian mereka lebih pada penggunaan *REST-API* dalam pengembangan layanan web.
3. Penelitian yang dilakukan oleh Lulu Agustin juga terkait dengan penerapan *REST-API* dalam pembangunan layanan web menggunakan Golang, tanpa menyebutkan penggunaan metode pengembangan tertentu. Hal ini menunjukkan bahwa penelitiannya lebih terfokus pada penggunaan *REST-API* dengan Golang.
4. Riska Aryanti, Eka Fitriani, Dian Ardiansyah, dan Atang Saepudin membahas pengembangan sistem informasi akademik, namun menggunakan metode

Rapid Application Development (RAD). Ini menunjukkan bahwa fokus penelitian mereka adalah pada pengembangan sistem informasi akademik dengan pendekatan *RAD*.

5. Gedmi Sari Melati Rambe dan Rita Novita Sari melakukan penelitian yang fokus terhadap pengembangan aplikasi sistem informasi akademik berbasis mobile dan *REST-API*-nya, tanpa menyebutkan bahasa pemrograman apa yang digunakan. Ini menunjukkan bahwa fokus penelitian mereka adalah pada pengembangan aplikasi mobile untuk sistem informasi akademik.

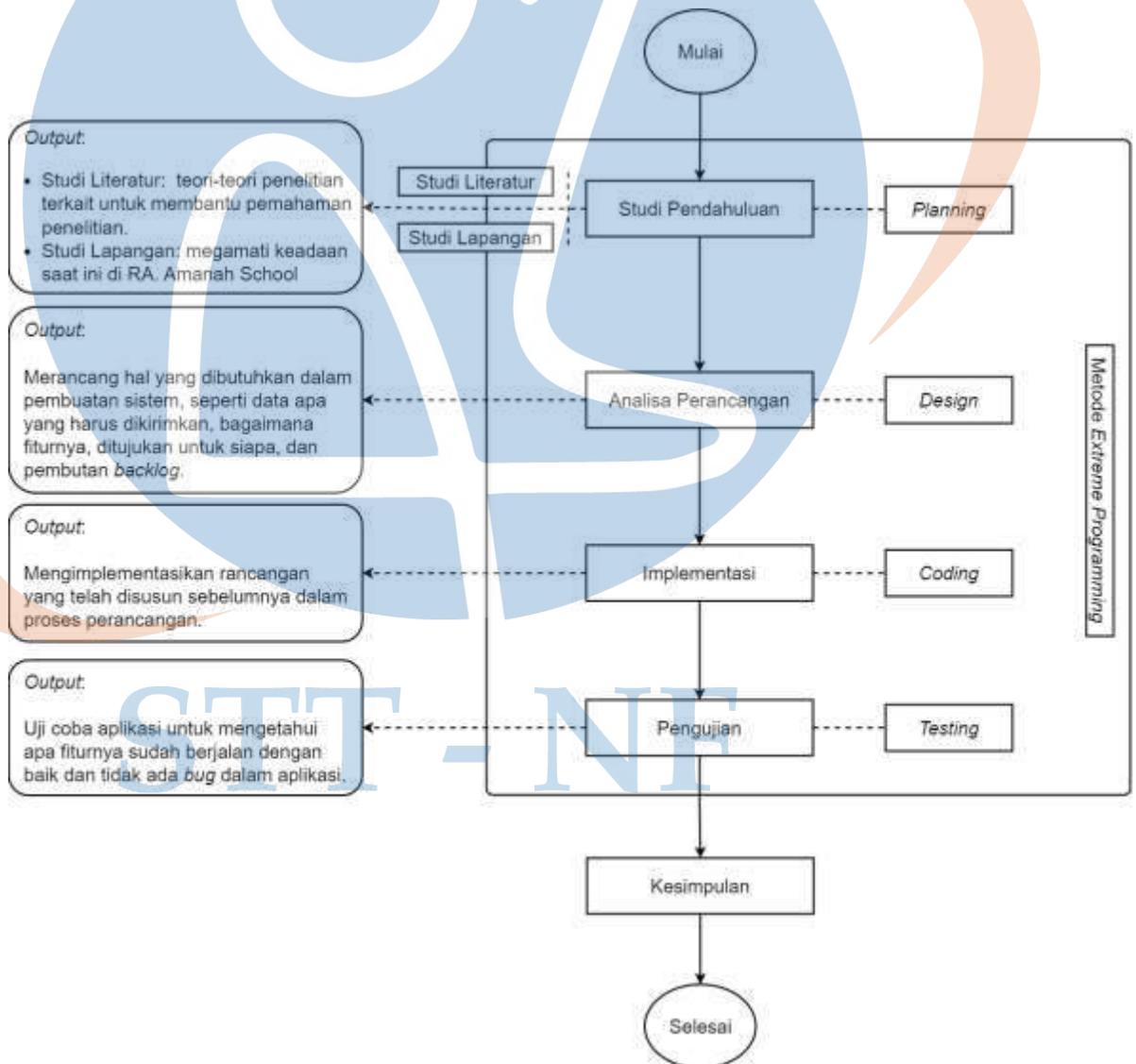
Dari uraian sebelumnya, penelitian ini lebih mendekati penelitian yang dikembangkan oleh Galang Prang, Olivia Liando, Alfrina Mewengkang yang berjudul Sistem Informasi Pencapaian Indikator Kinerja Utama 6 di Universitas Negeri Manado, dan dapat disimpulkan bahwa belum ada penelitian yang membahas pembangunan *REST-API* untuk sistem informasi akademik dengan menggunakan bahasa pemrograman Go (Golang) dan metode *Extreme Programming (XP)*. Oleh karena itu, penelitian ini yang berjudul “**Rancang Bangun *REST-API* Sistem Informasi Akademik Menggunakan Golang: Studi Kasus di RA. Amanah School**” akan mengisi kekosongan tersebut dengan fokus pada pengembangan *REST-API* untuk sistem informasi akademik dengan teknologi dan pendekatan tersebut.

STT - NF

BAB III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Berikut adalah langkah-langkah yang akan dilakukan dalam menjalankan proyek tugas akhir dengan menerapkan metode *Extreme Programming*. Tahapan penelitian ini mencakup detail alur kerja untuk setiap proses yang telah dirancang, sehingga penelitian dapat dilaksanakan dengan terstruktur dan jelas.



Gambar 3. 1 Tahapan Penelitian

3.1.1 Studi Pendahuluan

Pada tahap awal penelitian, dilakukan peninjauan pustaka secara komprehensif terhadap artikel dan jurnal ilmiah terkait desain aplikasi untuk sistem informasi akademik. Hasil dari studi pendahuluan ini menjadi landasan fundamental dalam merumuskan model penelitian yang menuntun arah penelitian dan mendefinisikan elemen-elemen krusial untuk mencapai tujuan penelitian yang ditetapkan. Model penelitian ini berfungsi sebagai panduan dasar yang memandu peneliti dalam menyelesaikan penelitian dengan efektif dan efisien.

3.1.2 Analisa Perancangan

Pada tahap perancangan, penulis merancang seluruh aspek pengembangan sistem informasi akademik secara menyeluruh. Hal ini meliputi perancangan skema *database*, menentukan fitur-fitur yang akan diimplementasikan, format pengiriman data, hingga menyusun tahapan *backlog* berdasarkan cerita pengguna yang diperoleh dari hasil wawancara. Semua elemen ini menjadi fondasi untuk membangun alur kerja *REST-API* yang handal dan efisien.

3.1.3 Implementasi

Pada tahap ini, proses implementasi ini dilakukan secara sistematis dan terstruktur berdasarkan prioritas yang telah ditetapkan dalam *product backlog*. Ini melibatkan proses pembuatan kebutuhan pengguna hingga pembuatan *REST-API* menggunakan Golang.

3.1.4 Pengujian

Pada tahap pengujian, sistem yang telah dikembangkan diuji menggunakan *blackbox testing* untuk memastikan bahwa sistem berjalan dengan baik sesuai dengan rancangan pengujian.

3.1.5 Kesimpulan

Tahap akhir penelitian ini berfokus pada penyimpulan hasil temuan. Peneliti memaparkan solusi yang telah diimplementasikan untuk mengatasi permasalahan

komunikasi informasi di RA. Amanah School. Kesimpulan ini memuat evaluasi terkait efektivitas solusi dalam menangani permasalahan, serta pencapaian tujuan penelitian yang telah ditetapkan sebelumnya.

3.2 Rancangan Penelitian

Desain penelitian mendetailkan langkah-langkah yang akan ditempuh dalam penelitian ini, meliputi pendekatan untuk mengatasi hambatan yang dihadapi, jenis penelitian yang dipilih, serta metode pengumpulan data yang akan diterapkan.

3.2.1 Jenis Penelitian

Penelitian ini termasuk dalam kategori Penelitian Pengembangan (*R&D*). Metode penelitian *R&D* dipilih karena dianggap cocok untuk mengatasi masalah dengan cara menemukan solusi inovatif melalui pengembangan produk atau teknologi baru. Pendekatan ini melibatkan identifikasi akar permasalahan, analisis kebutuhan, dan perancangan solusi yang efektif.

Studi kasus yang diangkat dalam penelitian ini adalah rancang bangun *REST-API* sistem informasi akademik RA. Amanah School. Penelitian ini melibatkan analisis kebutuhan pengguna, perancangan *REST-API*, implementasi, pengujian, dan evaluasi. Adapun luaran dari penelitian ini adalah:

1. Arsitektur *REST-API* yang dirancang dengan mengikuti prinsip desain *RESTful* yang baik.
2. *REST-API* menyediakan *endpoint* yang memungkinkan pengguna untuk mengakses berbagai data dan fungsi dalam sistem informasi akademik.
3. Penarikan kesimpulan untuk mengukur efektivitas solusi *REST-API* yang dikembangkan dalam penelitian ini. Evaluasi ini akan memberikan gambaran tentang sejauh mana implementasi *REST-API* berhasil memenuhi kebutuhan pengguna dan mencapai tujuan yang diharapkan

3.2.2 Metode Analisis Data

Dalam penelitian ini, penulis memilih metode kualitatif sebagai pendekatan utama. Hal ini dikarenakan untuk mendapatkan pemahaman yang mendalam dan

menyeluruh mengenai kebutuhan dan permasalahan yang dihadapi pengguna. Metode kualitatif memungkinkan pengguna untuk memberikan tanggapan yang terbuka dan bebas, tanpa dibatasi oleh pilihan jawaban yang telah ditentukan. Dengan demikian, pengguna dapat mengungkapkan pendapat dan pengalaman mereka secara detail, sehingga peneliti dapat memperoleh informasi yang lebih kaya dan mendalam. Pendekatan kualitatif ini memberikan ruang bagi pengguna untuk didengarkan dan berperan aktif dalam proses penelitian. Hal ini diharapkan dapat meningkatkan kualitas dan relevansi hasil penelitian, sehingga dapat memberikan manfaat yang lebih besar bagi pengguna.

3.2.3 Metode Pengumpulan Data

Dalam penelitian ini, data dan informasi dikumpulkan melalui sejumlah metode, yaitu:

1. Wawancara dengan pengguna untuk memahami kebutuhan dan harapan mereka secara detail. Tujuannya adalah untuk mendapatkan informasi kontekstual tentang pengalaman, kendala, dan fitur ideal yang diharapkan pengguna dalam sistem.
2. Studi kepustakaan merupakan langkah awal yang krusial dalam membangun *REST-API* untuk sistem informasi akademik RA. Amanah School. Hal ini dilakukan dengan mengumpulkan informasi dan pengetahuan terkait dari berbagai sumber terpercaya, seperti buku, jurnal, artikel, dan makalah. Proses ini dilakukan secara cermat dan sistematis untuk memastikan relevansi dan kredibilitas informasi yang diperoleh, guna mendukung pengembangan *REST-API* yang efektif dan optimal.
3. Observasi dilakukan terhadap RA. Amanah School untuk mengamati secara langsung perilaku, interaksi, dan proses penyampaian informasi yang dilakukan di sana. Tujuannya adalah untuk mendapatkan informasi yang akurat dan mendalam tentang bagaimana RA. Amanah School beroperasi.

3.2.4 Metode Pengujian

Berdasarkan uraian sebelumnya, penelitian ini menerapkan desain pengujian untuk memperoleh hasil evaluasi yang sejalan dengan penelitian dan menunjukkan bahwa *REST-API* untuk sistem informasi akademik RA. Amanah School sesuai dengan ekspektasi penulis. Desain pengujian yang digunakan dalam perancangan *REST-API* ini menggunakan metode *black box testing*. Hasil pengujian ini diharapkan dapat menjadi bahan pertimbangan bagi pengguna dalam menggunakan sistem informasi akademik tersebut.

Pada penelitian ini, penulis menggunakan metode *Black Box* untuk menguji fungsionalitas *REST-API* yang dirancang. Pengujian dilakukan dengan menggunakan aplikasi Postman, yang memungkinkan pengujian fungsionalitas *REST-API* melalui penggunaan permintaan *HTTP*.

3.2.5 Metode Implementasi dan Evaluasi

Tahap implementasi dalam penelitian ini akan fokus pada demonstrasi hasil dari struktur yang direncanakan untuk sistem informasi akademik RA. Amanah School menggunakan metode *Extreme Programming (XP)*, yang akan dijelaskan dengan memperlihatkan beberapa tangkapan layar dari hasil desain *REST-API* tersebut. Setelah itu, evaluasi dilakukan untuk menilai kinerja fitur-fitur dalam *REST-API* sistem informasi akademik RA. Amanah School dengan menggunakan pendekatan pengujian *black box testing*. Tujuannya adalah untuk memastikan bahwa fitur-fitur tersebut beroperasi dengan baik dan sesuai dengan kebutuhan yang ditetapkan.

3.2.6 Lingkungan Pengembangan

Lingkungan Pengembangan mengacu pada tempat di mana penelitian dilakukan, rangkaian alat, perangkat lunak, dan sumber daya lain yang digunakan oleh pengembang perangkat lunak atau aplikasi untuk membuat, menguji, dan menyusun kode. Ini mencakup semua elemen yang diperlukan untuk mendukung proses pengembangan perangkat lunak dari awal hingga akhir.

3.2.6.1 Lokasi Penelitian

Penelitian ini memilih RA. Amanah School sebagai tempat penelitian karena adanya kekurangan dalam penyampaian informasi akademik di sekolah tersebut. Diharapkan bahwa penelitian ini akan berkontribusi dalam meningkatkan kualitas pendidikan dan proses penyampaian informasi di lembaga tersebut.

3.2.6.2 Alat dan Bahan

Penelitian ini menggunakan berbagai alat dan bahan untuk mengumpulkan data dengan cermat. Alat-alat tersebut membantu dalam proses penelitian, seperti:

1. Kebutuhan Alat

Penelitian tugas akhir ini menggunakan laptop sebagai perangkat keras utama.

Berikut adalah spesifikasi laptop yang digunakan:

- Acer Aspire E1-431
- Windows 10 Home 64-bit
- RAM 6GB
- Intel® Core™ i5-2401M CPU @ 2,39GHz (4CPUs) 2.GHz

2. Kebutuhan Bahan

Penelitian ini menggunakan beberapa perangkat lunak dengan spesifikasi tertentu untuk membantu proses penelitian dan mencapai tujuan yang diinginkan.

Perangkat lunak tersebut antara lain:

- *Text Editor* : Visual Studio Code (versi 1.87.2)
- *Go Version* : go1.20 windows/amd64 (64-bit)
- *Platform API* : Postman
- *Desain diagram* : app.diagrams.net (versi web)

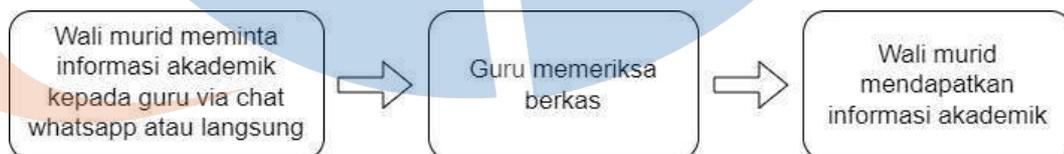
BAB IV IMPLEMENTASI DAN EVALUASI

4.1 Analisis dan Perancangan Sistem

Tahap analisis dan perancangan sistem pada proyek ini berfokus pada pemahaman mendalam mengenai permasalahan penyampaian informasi akademik yang dihadapi RA. Amanah School. Hal ini dilakukan untuk mendefinisikan kebutuhan sistem secara komprehensif dalam konteks penelitian. Tahapan ini melibatkan beberapa analisis, seperti analisis *user requirement*, identifikasi aktor, *product backlog*, pembuatan *activity diagram*, *class diagram*, *sequence diagram*, dan *use case diagram*.

4.1.1 Analisis Sistem Berjalan

Saat ini, sistem akses informasi akademik di RA. Amanah School masih terbilang terbatas. Para wali murid hanya dapat memperoleh informasi akademik anak mereka apabila mereka secara proaktif meminta kepada guru, baik secara langsung saat bertemu di sekolah ataupun melalui pesan WhatsApp. Hal ini berarti bahwa informasi akademik tidak secara otomatis tersedia dan mudah diakses oleh wali murid, dan mereka harus berinisiatif untuk memintanya.



Gambar 4. 1 Alur Informasi Akademik RA. Amanah School

Pada Gambar 4.1 Alur Informasi Akademik RA. Amanah School proses pengambilan informasi akademik di RA. Amanah School dimulai dengan inisiatif wali murid untuk meminta informasi kepada guru. Permintaan ini dapat dilakukan secara langsung saat bertemu guru di sekolah atau melalui pesan WhatsApp. Guru kemudian akan memeriksa berkas-berkas yang relevan untuk mendapatkan informasi yang dibutuhkan. Setelah informasi terkumpul, guru akan memberikannya kepada wali murid.

Sistem pengambilan informasi akademik di RA. Amanah School, seperti yang dijelaskan sebelumnya, memiliki keterbatasan dalam hal kemudahan akses. Ketergantungan pada inisiatif wali murid untuk bertanya kepada guru yang bersangkutan menjadikannya tidak praktis dan berpotensi menimbulkan hambatan. Hal ini dikarenakan informasi hanya bisa diperoleh jika wali murid aktif bertanya kepada guru yang bertanggung jawab. Jika guru tersebut berhalangan hadir di sekolah atau tidak dapat dihubungi melalui pesan singkat, maka proses pengambilan informasi akademik menjadi terhambat. Situasi ini dapat menimbulkan ketidaknyamanan bagi wali murid yang membutuhkan informasi akademik secara cepat dan mudah. Terkait dengan keterbatasan akses informasi akademik di RA. Amanah School, pembuatan sistem informasi akademik dinilai sebagai solusi yang tepat. Sistem ini memungkinkan guru untuk menyediakan informasi akademik yang terbaru dan mudah diakses oleh wali murid, kapanpun dan dimanapun.

4.1.2 User Story

Berdasarkan hasil wawancara dengan calon pengguna, penulis berhasil mengidentifikasi beberapa cerita pengguna (*user stories*) yang menggambarkan kebutuhan dan harapan mereka terhadap aplikasi. Berikut merupakan *user requirement* yang diperoleh:

Tabel 4. 1 User Story

No.	Sebagai	Saya Ingin	Sehingga
1	Admin	Input data guru	Saya dapat mengelola data guru dengan mudah.
2	Admin	Input data siswa	Saya dapat mengelola data siswa dengan mudah.
3	Guru, Admin	Melihat daftar guru	Saya dapat mengetahui informasi semua guru dengan mudah.

No.	Sebagai	Saya Ingin	Sehingga
4	Guru, Admin	Melihat daftar siswa	Saya dapat mengetahui informasi semua siswa dengan mudah.
5	Admin	Merubah data guru	Saya dapat memastikan bahwa data guru telah benar.
6	Guru, Admin	Merubah data siswa	Saya dapat memastikan bahwa data siswa telah benar.
7	Guru, Admin	<i>Input</i> data nilai siswa	Saya dapat memantau perkembangan belajar siswa dengan mudah.
8	Guru, Admin	Merubah data nilai siswa	Saya dapat memastikan bahwa data nilai yang saya <i>input</i> sudah benar.
9	Guru, Wali Murid	Melihat nilai siswa	Saya bisa memantau perkembangan belajar anak saya.
10	Guru, Admin	<i>Input</i> data kehadiran siswa	Saya dapat memantau kehadiran siswa dengan mudah.
11	Guru, Admin	Merubah data kehadiran siswa	Saya dapat memastikan bahwa data kehadiran siswa telah benar.
12	Guru, Admin, Wali Murid	Melihat daftar kehadiran	Saya dapat memantau kehadiran anak saya di sekolah.

No.	Sebagai	Saya Ingin	Sehingga
13	Guru, Admin	Input pembayaran SPP	Saya dapat memantau pembayaran SPP siswa dengan mudah.
14	Guru, Admin	Merubah data pembayaran SPP	Saya dapat memastikan bahwa data pembayaran SPP siswa telah benar.
15	Guru, Admin, Wali Murid	Melihat daftar pembayaran SPP	Saya dapat mengetahui riwayat pembayaran SPP anak saya.

4.1.3 Identifikasi Aktor

Berdasarkan uraian *user stories* sebelumnya, dapat disimpulkan bahwa beberapa aktor penting berperan dalam pengembangan sistem informasi akademik.

Aktor-aktor tersebut adalah:

Tabel 4. 2 Identifikasi Aktor

No.	Aktor	Deskripsi
1	Admin	Pengguna istimewa yang memiliki akses penuh untuk mengelola aplikasi. Tanggung jawab mereka meliputi pengelolaan data guru dan siswa, serta sistem pembayaran SPP. Dengan peran sentral ini, admin memastikan kelancaran operasional dan keefektifan aplikasi.
2	Guru	Guru adalah seorang pendidik profesional yang memiliki tugas utama mendidik, mengajar, membimbing, mengarahkan, melatih, menilai, dan mengevaluasi peserta didik. Guru dapat mengelola data siswa, daftar kehadiran, penilaian, dan juga pembayaran SPP.

No.	Aktor	Deskripsi
3	Wali Murid	Orang yang memiliki tanggung jawab atas siswa. Pengguna dapat memantau perkembangan nilai siswa, daftar kehadiran, dan juga melihat riwayat pembayaran SPP.

4.1.4 Product Backlog

Berdasarkan cerita pengguna (*user stories*), peneliti telah merangkum fitur-fitur yang diperlukan dalam sistem informasi akademik. Rangkuman ini menjelaskan berbagai tugas (*task*) yang harus dilakukan dalam pengembangan aplikasi, prioritas fitur, dan estimasi *Story Points* yang digunakan untuk mengukur kompleksitas atau waktu pengerjaan setiap task.

Setiap Task dalam pengembangan aplikasi diberi nilai prioritas berdasarkan kepentingannya. Kompleksitas *task* diukur dengan skala *Story Points*, di mana 1 nilai setara dengan 2-6 jam pengerjaan, dan 2 nilai setara dengan 7-10 jam. Deskripsi aplikasi juga disertakan untuk memperjelas tujuan dan fungsinya. Informasi ini diharapkan dapat membantu pengembangan sistem informasi akademik yang lebih terstruktur dan efisien.

Tabel 4. 3 Product Backlog

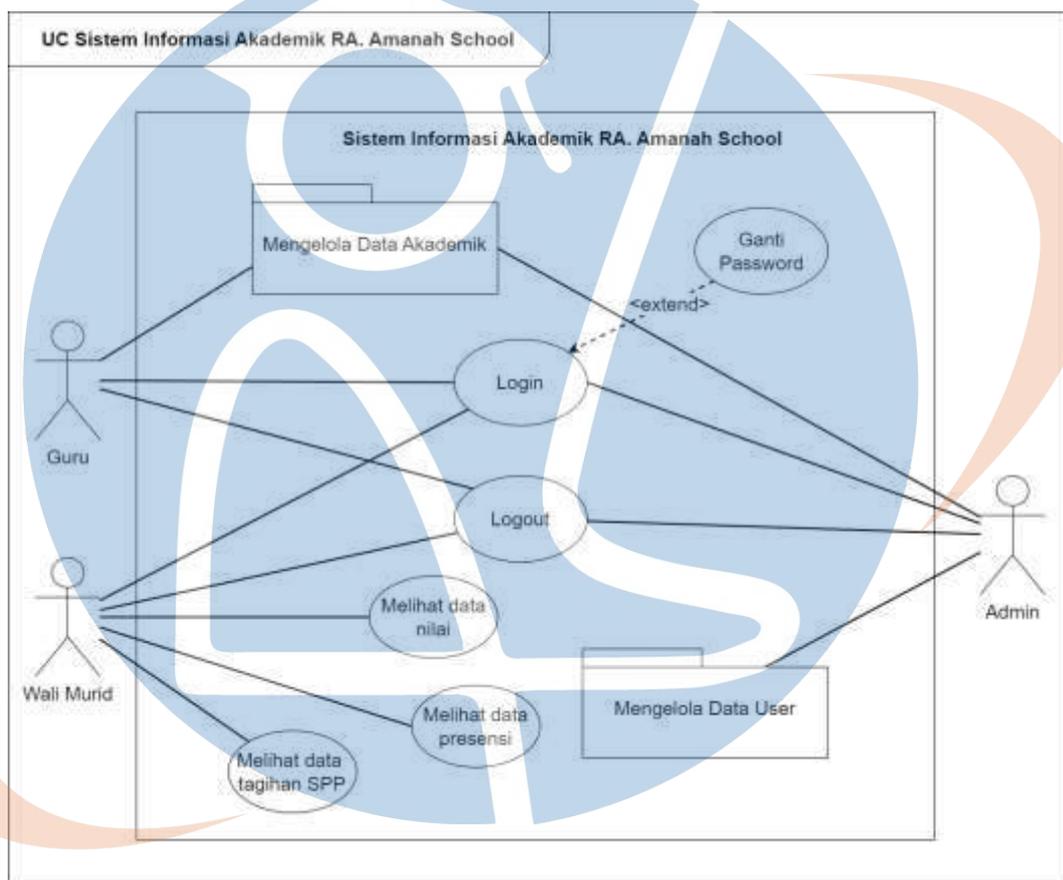
No.	Task	Prioritas	Story Points	Deskripsi
1	Pembuatan <i>User Requirement</i>	Tinggi	1	Merancang dan mencatat kebutuhan pengguna dengan detail untuk mengenali fitur inti dari aplikasi.
2	Pembuatan skema <i>database</i>	Tinggi	1	Merancang skema <i>database</i> untuk penyimpanan data sistem informasi akademik.

No.	Task	Prioritas	Story Points	Deskripsi
3	<i>Migration database table</i>	Tinggi	1	Memastikan struktur <i>database</i> sesuai desain dengan melakukan migrasi skema <i>database</i> ke dalam sistem basis data.
4	<i>API login dan logout</i>	Tinggi	1	Membuat <i>API</i> yang memungkinkan pengguna <i>login</i> dan <i>logout</i> dengan menggunakan token <i>JWT</i> .
5	<i>API pembayaran tagihan</i>	Tinggi	1	Membuat <i>API</i> untuk mengelola data pembayaran tagihan <i>SPP</i> .
6	<i>API presensi siswa</i>	Tinggi	1	Membuat <i>API</i> untuk mengelola data kehadiran.
7	<i>API nilai perkembangan siswa</i>	Tinggi	1	Membuat <i>API</i> untuk mengelola data penilaian.
8	<i>API untuk mengelola data guru</i>	Tinggi	2	Membuat <i>API</i> untuk mengelola data guru.
9	<i>API untuk mengelola data siswa</i>	Tinggi	2	Membuat <i>API</i> untuk mengelola data siswa

4.1.5 Use Case Diagram

Use case diagram penggunaan Sistem Informasi Akademik RA. Amanah School menggambarkan interaksi antara pengguna (aktor) dengan sistem melalui

berbagai kasus penggunaan. Diagram ini menunjukkan bagaimana pengguna berinteraksi dengan sistem untuk mendapatkan informasi akademik, seperti nilai, absensi, dan tagihan SPP. Diagram ini membantu memvisualisasikan fungsionalitas sistem dan bagaimana pengguna berinteraksi dengannya. Hal ini penting untuk memahami bagaimana sistem bekerja dan bagaimana pengguna dapat menggunakannya untuk mendapatkan informasi yang mereka butuhkan.



Gambar 4. 2 Use Case Diagram

Pada Gambar 4.1 *Use Case Diagram* menggambarkan interaksi antara 3 aktor (Admin, Guru, dan Wali Murid) dengan Sistem Informasi Akademik RA. Amanah School. Masing-masing aktor memiliki peran dan akses yang berbeda dalam sistem.

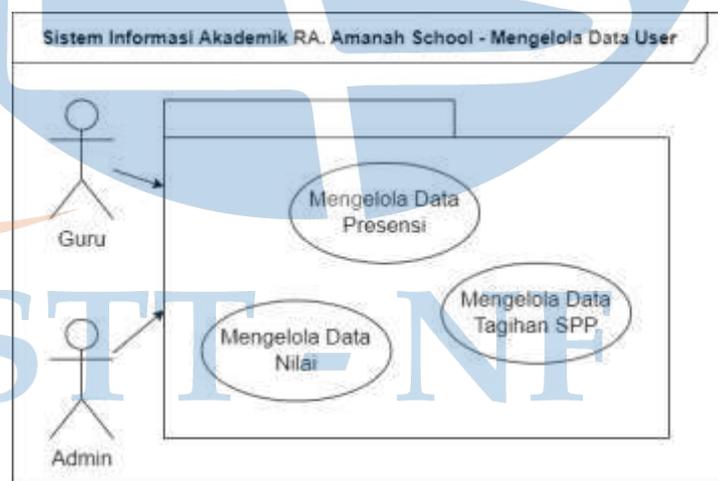
1. Admin memiliki kewenangan penuh atas pengelolaan data pengguna dan data akademik. Mereka dapat menambah, menghapus, dan mengubah data pengguna, termasuk Guru dan Siswa. Hal ini penting untuk memastikan keakuratan data dalam sistem.

2. Guru berperan dalam mengelola data akademik siswa. Guru dapat menambahkan, mengubah, dan menghapus data nilai, presensi, dan informasi akademik lainnya. Akses ini memungkinkan Guru untuk memantau kemajuan belajar siswa dan memberikan penilaian yang akurat.
3. Wali Murid memiliki akses terbatas pada sistem. Mereka hanya dapat melihat informasi akademik yang telah dimasukkan oleh Guru, seperti nilai, presensi, dan tagihan SPP. Orang tua dapat memantau perkembangan belajar anak mereka dan memastikan pembayaran SPP tepat waktu.

Secara keseluruhan, *Use Case Diagram* ini menggambarkan bagaimana Sistem Informasi Akademik RA. Amanah School memfasilitasi interaksi yang terstruktur antara Admin, Guru, dan Wali Murid untuk mengelola dan mengakses informasi akademik dengan mudah dan efisien.

4.1.4.1 *Use Case Diagram* Mengelola Data Akademik

Pada gambar 4.1 *Use Case Diagram* terdapat *package* Mengelola Data Akademik, dalam *package* tersebut menjelaskan bagaimana interaksi Guru dan Admin dengan sistem pada saat mengelola data akademik.



Gambar 4. 3 *Use Case Diagram* Mengelola Data Akademik

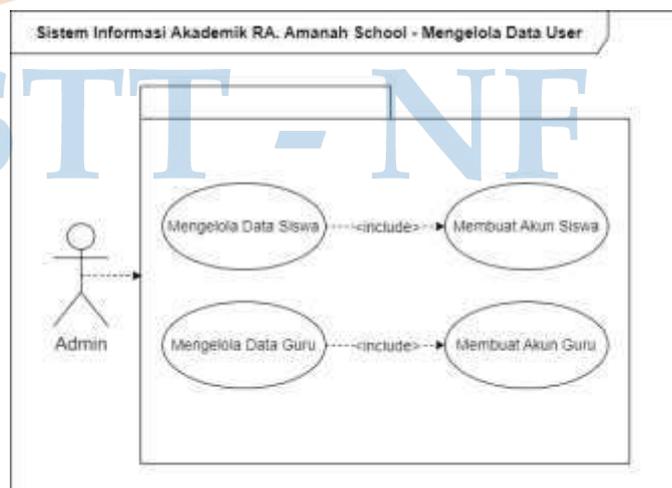
Guru dan Admin memiliki peran penting dalam mengelola data akademik siswa di Sistem Informasi Akademik RA. Amanah School. Mereka dibekali dengan berbagai akses untuk melakukan berbagai aktivitas penting, antara lain:

1. Menambahkan Data: Guru dan Admin dapat memasukkan data akademik baru ke dalam sistem, seperti nilai ujian, tugas, dan informasi kehadiran. Hal ini memungkinkan Guru dan Admin untuk mendokumentasikan kemajuan belajar siswa secara akurat dan komprehensif.
2. Mengubah Data: Guru dan Admin memiliki kewenangan untuk mengubah data akademik yang telah dimasukkan sebelumnya. Hal ini berguna untuk memperbaiki kesalahan input atau memperbarui informasi terbaru terkait perkembangan belajar siswa.
3. Menghapus Data: Dalam beberapa kasus, Guru mungkin perlu menghapus data akademik tertentu. Hal ini dapat dilakukan dengan alasan tertentu, seperti data yang sudah tidak relevan atau duplikasi.

Kemampuan Guru dan Admin untuk menambahkan, mengubah, dan menghapus data akademik merupakan elemen penting dalam memastikan kelancaran proses belajar mengajar di RA. Amanah School. Sistem ini memungkinkan Guru untuk memantau kemajuan belajar siswa secara efektif dan memberikan penilaian yang akurat.

4.1.4.2 Use Case Diagram Mengelola Data User

Pada gambar 4.1 *use case diagram* juga bisa kita lihat terdapat *package* mengelola data *user*. Dalam *package* tersebut berisi interaksi antara Admin dengan sistem dalam mengelola data *user*.



Gambar 4. 4 Use Case Diagram Mengelola Data User

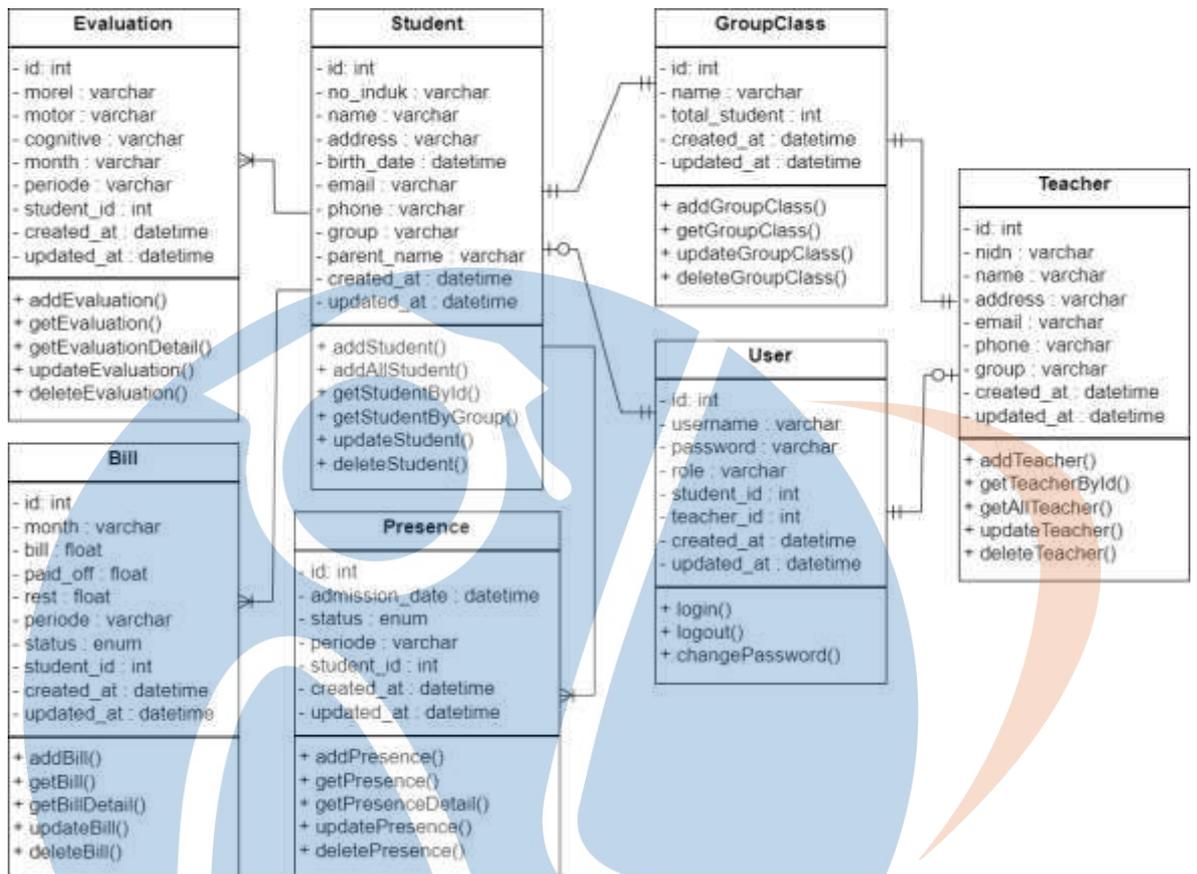
Admin memiliki tanggung jawab penting dalam mengelola data pengguna di Sistem Informasi Akademik RA. Amanah School. Kewenangan mereka meliputi:

1. **Menambahkan Data Pengguna:** Admin dapat memasukkan data pengguna baru ke dalam sistem, termasuk informasi profil, kontak, dan peran pengguna (Guru atau Wali Murid). Setiap kali Admin menambahkan data pengguna baru, sistem secara otomatis akan membuat akun untuk pengguna tersebut. Hal ini memudahkan pengguna untuk mengakses sistem dan menggunakan fitur-fiturnya sesuai dengan peran mereka.
2. **Mengubah Data Pengguna:** Admin memiliki akses untuk mengubah data pengguna yang telah ada, seperti informasi profil, kontak, atau peran pengguna. Hal ini berguna untuk memperbarui informasi pengguna agar tetap akurat dan relevan.
3. **Menghapus Data Pengguna:** Dalam kondisi tertentu, Admin mungkin perlu menghapus data pengguna tertentu. Hal ini dapat dilakukan dengan alasan tertentu, seperti akun pengguna yang tidak aktif atau data yang sudah tidak relevan.

Peran Admin dalam mengelola data pengguna sangat penting untuk menjaga keamanan dan kelancaran operasi sistem. Sistem ini memastikan bahwa hanya pengguna yang sah yang memiliki akses ke informasi dan fitur-fitur yang sesuai dengan peran mereka.

4.1.6 Class Diagram

Class diagram sistem informasi akademik RA. Amanah School menggambarkan hubungan antar kelas, objek, atribut, dan operasi dalam sistem. Diagram ini membantu proses desain sistem dan membuat spesifikasi sistem aplikasi menjadi lebih jelas dan terstruktur.



Gambar 4. 5 Class Diagram

Pada Gambar 4.4 *Class Diagram* menggambarkan struktur sistem informasi akademik RA. Amanah School dengan menggunakan tujuh kelas, yaitu *student*, *teacher*, *user*, *bill*, *presence*, *group class*, dan *evaluation*. Setiap kelas memiliki atribut dan operasi untuk mendefinisikan karakteristik dan interaksinya dalam sistem. Atribut pada setiap kelas mendefinisikan karakteristik kelas tersebut. Operasi pada setiap kelas memungkinkan kelas untuk berinteraksi dengan data dan kelas lain. Hampir semua kelas memiliki operasi yang sama, yaitu menambahkan data baru ke atribut kelas, menghapus data dari atribut kelas, mengubah data pada atribut kelas, melihat data pada atribut kelas. Akan tetapi, untuk kelas *user* memiliki operasi yang berbeda, diantaranya yaitu *login* untuk masuk ke sistem, *logout* untuk keluar dari sistem, dan *change password* untuk mengubah kata sandi.

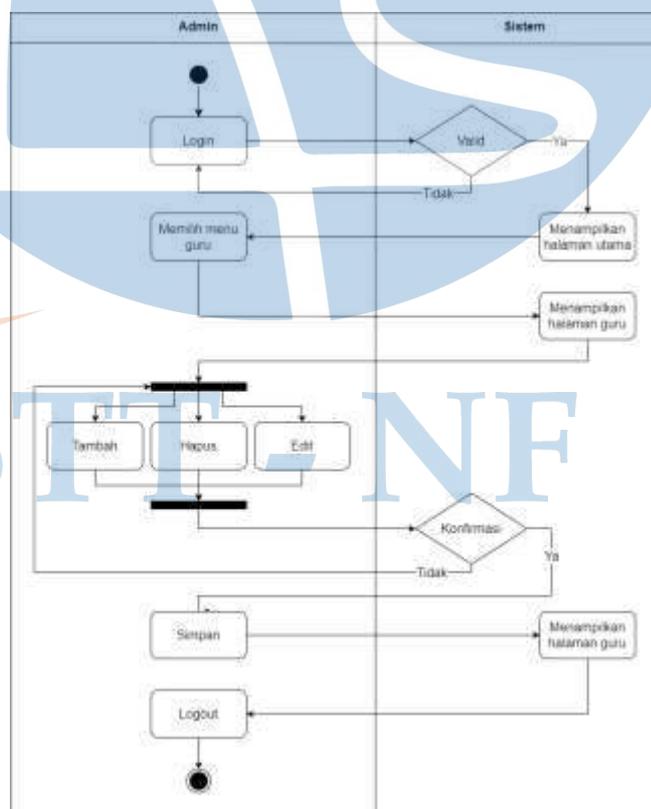
Diagram Kelas ini membantu memahami struktur dan fungsionalitas sistem informasi akademik RA. Amanah School dengan menunjukkan bagaimana kelas-kelas dalam sistem saling terhubung dan berinteraksi satu sama lain.

4.1.7 Activity Diagram

Activity diagram adalah diagram UML (*Unified Modeling Language*) yang digunakan untuk memvisualisasikan alur kerja (*workflow*) atau proses bisnis dalam suatu sistem. Diagram ini menunjukkan langkah-langkah yang terlibat dalam suatu proses, dan bagaimana langkah-langkah tersebut saling terkait. Diagram ini dapat membantu analis sistem, pemangku kepentingan, dan pengembang untuk memahami proses bisnis, mengidentifikasi inefisiensi, dan meningkatkan sistem informasi akademik.

1. Activity Diagram Mengelola Data Guru

Activity diagram ini menggambarkan alur kerja **Admin** dalam mengelola data guru di Sistem Informasi Akademik RA. Amanah School. Diagram ini menunjukkan langkah-langkah yang harus dilalui oleh admin, mulai dari *login* hingga *logout*, dan bagaimana sistem memverifikasi data, menyimpan perubahan, dan menampilkan informasi yang relevan.



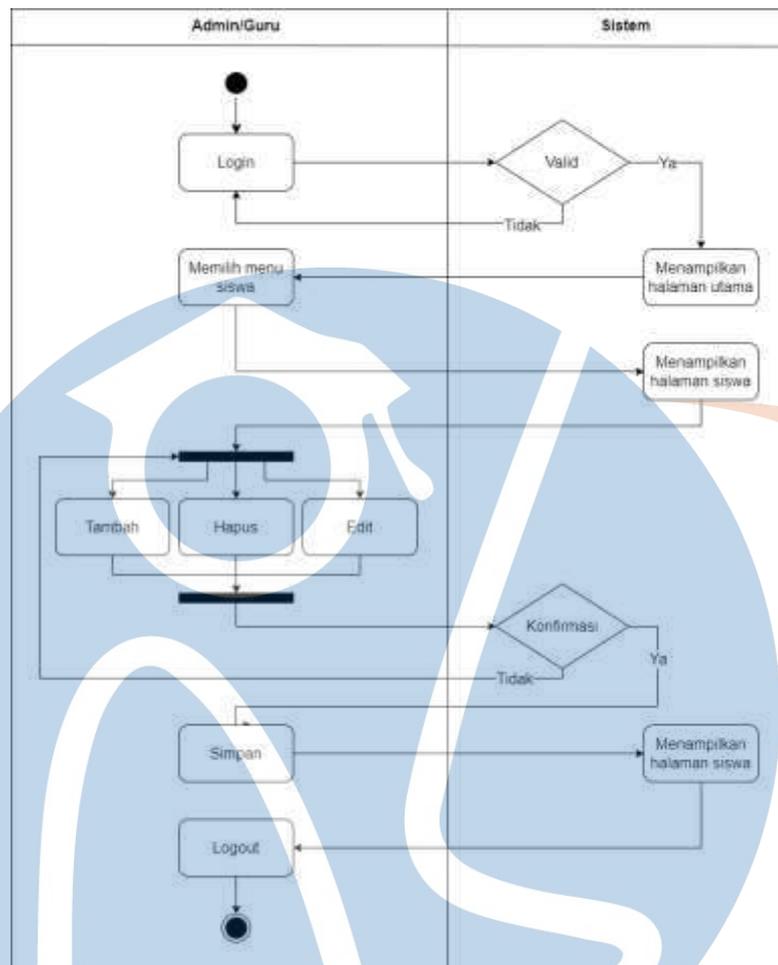
Gambar 4. 6 Activity Diagram Mengelola Data Guru

Pada gambar 4.2 *Activity Diagram* Mengelola Data Guru, admin memulai proses dengan melakukan *login* ke Sistem Informasi Akademik RA. Amanah School. Sistem kemudian akan memverifikasi data *login* yang dimasukkan. Jika data valid, sistem akan menampilkan halaman utama. Di halaman utama, admin memilih menu guru untuk mengakses halaman kelola data guru. Admin dapat memilih salah satu dari tiga aktivitas, seperti menambah data guru, menghapus data guru, atau mengubah data guru. Setelah menyelesaikan aktivitas yang dipilih, sistem akan melakukan konfirmasi dan menyimpan perubahan ke dalam *database*. Admin kemudian akan diarahkan kembali ke halaman guru. Jika sudah selesai, admin dapat melakukan *logout* dari sistem.

2. *Activity Diagram* Mengelola Data Siswa

Activity diagram ini menggambarkan alur kerja admin dan guru dalam mengelola data siswa di Sistem Informasi Akademik RA. Amanah School. Diagram ini menunjukkan langkah-langkah yang harus dilalui oleh kedua pihak, mulai dari *login* hingga *logout*, dan bagaimana sistem memverifikasi data, menyimpan perubahan, dan menampilkan informasi yang relevan.

STT - NF

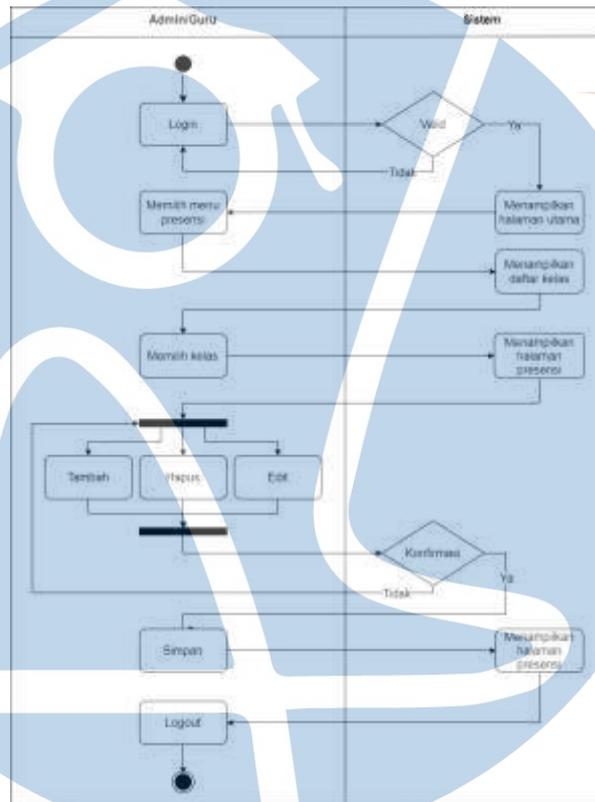


Gambar 4. 7 Activity Diagram Mengelola Data Siswa

Gambar 4.3 Activity Diagram Mengelola Data Siswa, baik admin maupun guru memulai proses dengan melakukan *login* ke Sistem Informasi Akademik RA. Amanah School. Sistem kemudian akan memverifikasi data *login* yang dimasukkan. Bila data valid, sistem akan menampilkan halaman utama. Di halaman utama, admin dan guru memilih menu Siswa untuk mengakses halaman kelola data siswa. Mereka bisa memilih salah satu dari tiga aktivitas, yaitu menambah data siswa, menghapus data siswa, atau mengubah data siswa. Setelah menyelesaikan aktivitas yang dipilih, sistem akan melakukan konfirmasi dan menyimpan perubahan ke dalam *database*. Baik admin maupun guru kemudian akan diarahkan kembali ke halaman siswa. Jika sudah selesai, admin dan guru dapat melakukan *logout* dari sistem.

3. Activity Diagram Mengelola Data Presensi

Activity diagram ini menunjukkan proses yang harus dilakukan oleh admin dan guru untuk mengelola data presensi di Sistem Informasi Akademik RA. Amanah School. Diagram ini juga menunjukkan bagaimana sistem memverifikasi data, menyimpan perubahan, dan menampilkan informasi yang relevan.



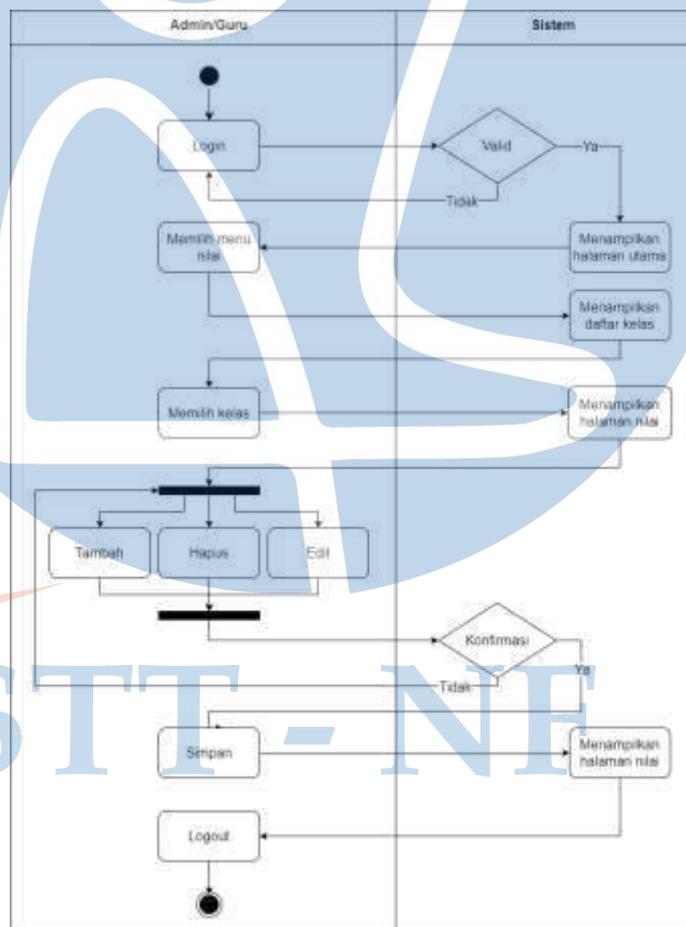
Gambar 4. 8 Activity Diagram Mengelola Data Presensi

Dapat dilihat pada gambar 4.4 Activity Diagram Mengelola Data Presensi, admin dan guru memulai proses dengan melakukan *login* ke Sistem Informasi Akademik RA. Amanah School. Sistem kemudian akan memverifikasi data login yang dimasukkan. Bila data valid, sistem akan menampilkan halaman utama. Di halaman utama, admin dan guru memilih menu presensi. Sebelum mengelola presensi, mereka akan diarahkan ke menu daftar kelas untuk memilih kelas mana yang akan dikelola. Setelah memilih kelas, mereka akan diarahkan ke halaman presensi siswa. Di halaman presensi, mereka dapat memilih salah satu dari tiga aktivitas: menambah data presensi, menghapus data presensi, atau mengubah data

presensi. Setelah menyelesaikan aktivitas yang dipilih, sistem akan melakukan konfirmasi dan menyimpan perubahan ke dalam *database*. Baik Admin maupun Guru kemudian akan diarahkan kembali ke halaman presensi. Jika sudah selesai, Admin dan Guru dapat melakukan *logout* dari sistem.

4. Activity Diagram Mengelola Data Nilai

Activity diagram ini menggambarkan langkah-langkah yang diambil oleh admin dan guru dalam mengelola data nilai di Sistem Informasi Akademik RA. Amanah School. Diagram ini juga memaparkan bagaimana sistem memverifikasi data, menyimpan perubahan, serta menampilkan informasi yang relevan.



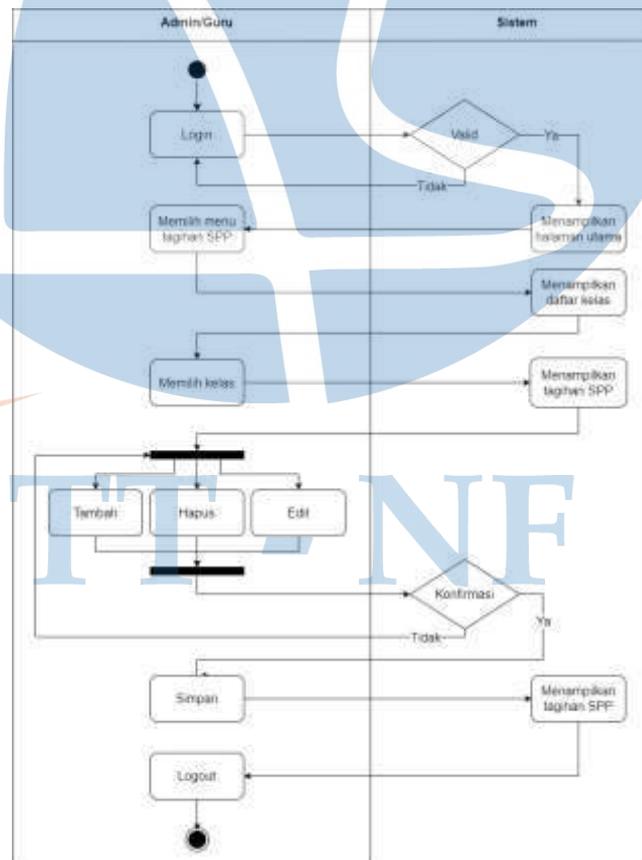
Gambar 4. 9 Activity Diagram Mengelola Data Nilai

Baik admin maupun guru memulai proses dengan *login* ke Sistem Informasi Akademik RA. Amanah School. Sistem kemudian memverifikasi data yang dimasukkan. Sistem akan menampilkan halaman utama jika data valid. Di halaman

utama, admin dan guru dapat memilih menu nilai. Namun, sebelum mereka dapat mengelola data nilai, mereka akan diarahkan ke menu daftar kelas untuk memilih kelas mana yang akan dikelola. Setelah memilih kelas, mereka akan diarahkan ke halaman nilai siswa, di mana mereka dapat memilih untuk menambah, menghapus, atau mengubah data nilai. Setelah menyelesaikan tugas-tugas tersebut, sistem akan mengkonfirmasi dan menyimpan perumahan. Selanjutnya, guru dan admin akan diarahkan kembali ke halaman nilai. Jika sudah selesai, admin dan guru dapat melakukan *logout* dari sistem.

5. Activity Diagram Mengelola Data Tagihan SPP

Activity diagram ini menunjukkan proses yang harus dilakukan oleh admin dan guru untuk mengelola data tagihan SPP di Sistem Informasi Akademik RA. Amanah School. Diagram ini juga menunjukkan bagaimana sistem memverifikasi data, menyimpan perubahan, dan menampilkan informasi yang relevan.

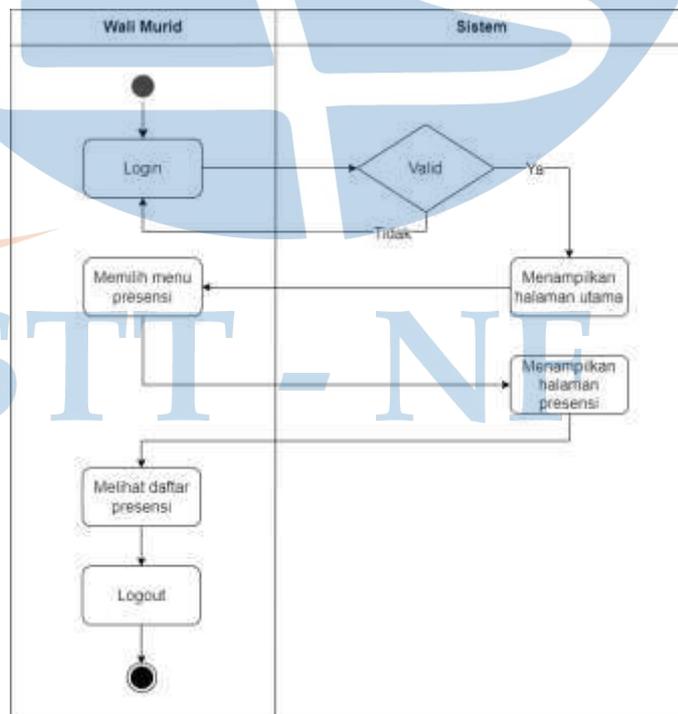


Gambar 4. 10 Activity Diagram Mengelola Data Tagihan SPP

Untuk memulai proses, admin atau guru masuk ke Sistem Informasi Akademik RA. Amanah School. Kemudian, sistem memverifikasi data yang dimasukkan. Halaman utama sistem akan ditampilkan jika data valid. Di halaman utama, admin dan guru dapat memilih menu Tagihan SPP. Sebelum mereka dapat mengelola data tagihan SPP, mereka harus memilih kelas dari menu daftar kelas. Setelah memilih kelas, mereka akan dibawa ke halaman tagihan SPP, di mana mereka dapat menambah, menghapus, atau mengubah data tagihan SPP. Setelah tugas-tugas tersebut diselesaikan, sistem akan mengonfirmasi dan menyimpan perubahan. Setelah itu, admin atau guru akan diarahkan kembali ke halaman nilai. Setelah selesai, admin dan guru dapat keluar dari sistem.

6. *Activity Diagram* Wali Murid Melihat Data Presensi

Activity diagram ini menggambarkan alur kerja wali murid dalam memantau daftar hadir siswa di Sistem Informasi Akademik RA. Amanah School. Diagram ini menunjukkan langkah-langkah yang harus dilalui oleh wali murid, mulai dari *login* hingga *logout*, dan bagaimana sistem memverifikasi data dan menampilkan informasi yang relevan.

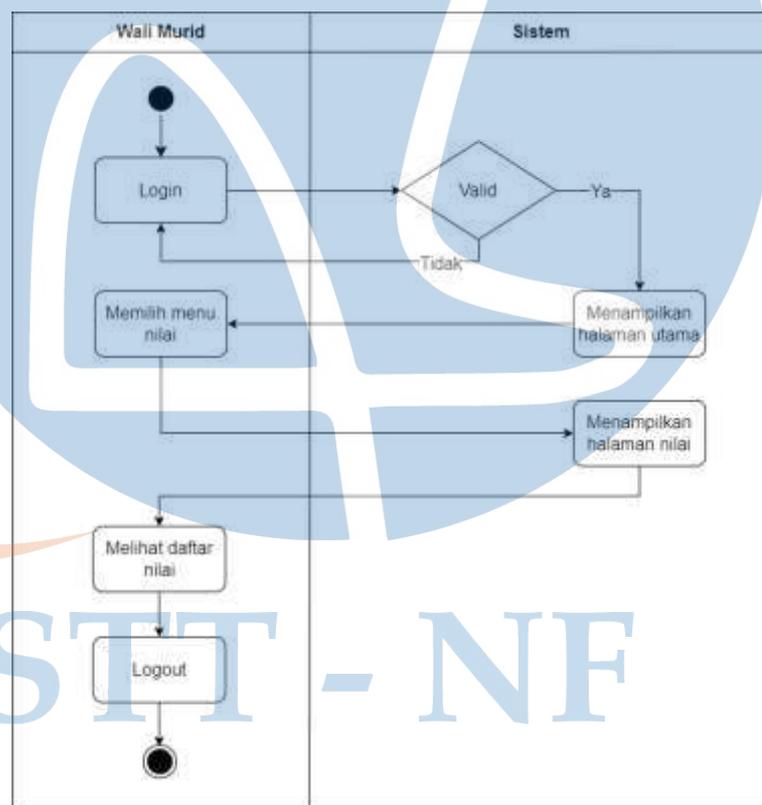


Gambar 4. 11 *Activity Diagram* Wali Murid Melihat Data Presensi

Untuk memulai proses, wali murid *login* ke Sistem Informasi Akademik RA Amanah School. Sistem kemudian memverifikasi data yang dimasukkan. Jika data valid, halaman utama akan ditampilkan. Di halaman utama, wali murid dapat memilih menu presensi. Kemudian, wali murid akan diarahkan ke halaman presensi untuk memantau daftar hadir siswa. Setelah itu, wali murid dapat *logout* dari sistem.

7. Activity Diagram Wali Murid Melihat Data Nilai

Activity diagram ini menunjukkan cara wali murid menggunakan Sistem Informasi Akademik RA Amanah School untuk mengakses data nilai. Diagram ini juga menunjukkan bagaimana sistem memverifikasi data dan menampilkan informasi nilai.



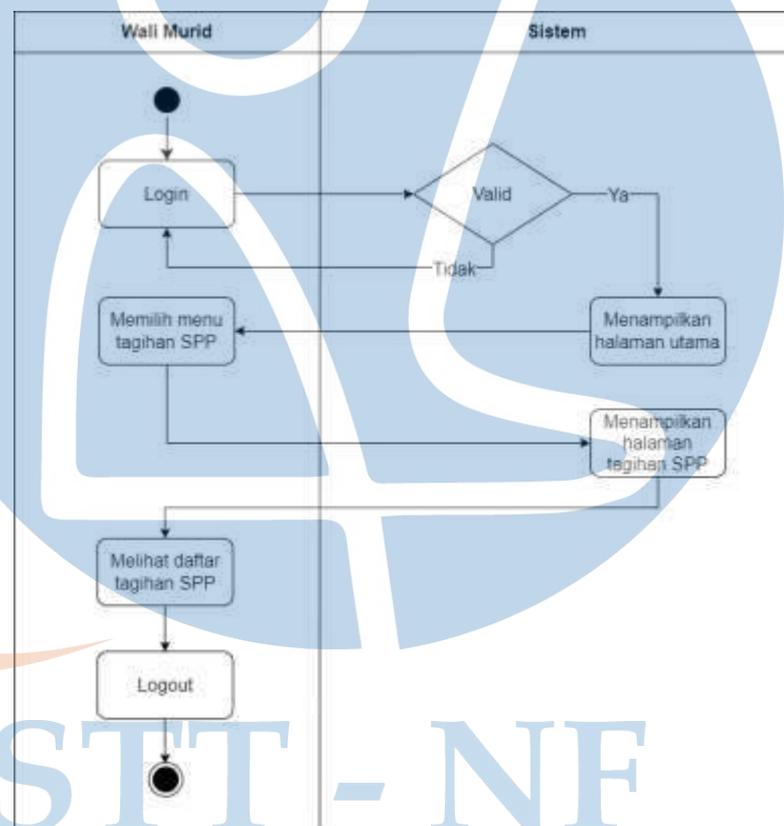
Gambar 4. 12 Activity Diagram Wali Murid Melihat Data Nilai

Untuk memulai proses, wali murid masuk ke Sistem Informasi Akademik RA Amanah School. Kemudian sistem memverifikasi data yang mereka masukkan. Halaman utama akan ditampilkan jika informasi valid. wali murid dapat memilih menu nilai di halaman utama. Kemudian wali murid akan diarahkan ke halaman

nilai, di mana mereka dapat memantau perkembangan siswa. Setelah itu, wali murid dapat *logout* dari sistem.

8. *Activity Diagram* Wali Murid Melihat Data Tagihan SPP

Activity diagram ini menggambarkan alur kerja wali murid dalam memantau tagihan SPP di Sistem Informasi Akademik RA. Amanah School. Diagram ini menunjukkan langkah-langkah yang harus dilalui oleh wali murid, mulai dari *login* hingga *logout*, dan bagaimana sistem memverifikasi data dan menampilkan informasi tagihan.



Gambar 4. 13 *Activity Diagram* Wali Murid Melihat Data Tagihan SPP

Wali murid memulai proses dengan melakukan login ke Sistem Informasi Akademik RA Amanah School. Sistem kemudian akan memverifikasi data *login* yang dimasukkan. Bila data valid, sistem akan menampilkan halaman utama. Di halaman utama, wali murid memilih menu Tagihan SPP. Wali murid kemudian akan diarahkan ke halaman tagihan SPP. Di sini, mereka dapat memantau tagihan

yang sudah dibayarkan. Setelah selesai, wali murid dapat melakukan *logout* dari sistem.

4.1.8 Rancangan Pengujian

Seperti yang telah dijelaskan sebelumnya, penelitian ini menerapkan metode *black box testing* untuk mengevaluasi sistem yang telah dikembangkan. Berikut adalah rincian skenario pengujian yang akan dilaksanakan:

Tabel 4. 4 Rancangan Pengujian - Kelola Data Guru

Skenario Uji		Kelola Data Guru		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
1	<i>Insert data guru dengan benar. (semua field terisi)</i>	<i>POST, http://localhost:3000/api/teachers</i>	Akan menghasilkan respon <i>Code: 200, Message: "Berhasil menambahkan data guru."</i> , dan <i>Data: (data yang di-input).</i>	Sesuai atau tidak sesuai.
2	<i>Insert data guru dengan salah. (terdapat field yang tidak terisi)</i>	<i>POST, http://localhost:3000/api/teachers</i>	Akan menghasilkan respon <i>Code: 400, Message: "BAD REQUEST."</i> , dan <i>Data: "Terdapat</i>	Sesua atau tidak sesuai.

Skenario Uji		Kelola Data Guru		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
			data yang belum diisi”.	
3	Mengubah data guru.	<i>PUT</i> , <i>http://localhost:3000/api/teachers/:teacherId</i>	Akan menghasilkan respon <i>Code: 200, Message: “Data berhasil diperbarui.”</i> , dan <i>Data:</i> (data guru yang telah diubah).	Sesuai atau tidak sesuai.
4	Menampilkan seluruh data guru.	<i>GET</i> , <i>http://localhost:3000/api/teachers</i>	Akan menghasilkan respon <i>Code: 200, Message: “OK.”</i> , dan <i>Data:</i> (seluruh data siswa).	Sesuai atau tidak sesuai.

Tabel 4.4 Rancangan Pengujian - Kelola Data Guru memuat skenario pengujian untuk mengelola data guru, meliputi *input*, *update*, dan menampilkan data guru. Skenario ini dirancang untuk memastikan fungsionalitas sistem dalam mengelola data guru dengan lengkap dan akurat.

Tabel 4. 5 Rancangan Pengujian - Kelola Data Siswa

Skenario Uji		Kelola Data Siswa		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
1	<i>Insert data siswa dengan benar. (semua field terisi)</i>	<i>POST, http://localhost:3000/api/students</i>	Akan menghasilkan respon <i>Code: 200, Message: "Berhasil menambahkan data siswa."</i> , dan <i>Data: (data yang di-input).</i>	Sesuai atau tidak sesuai.
2	<i>Insert data siswa dengan salah. (terdapat field yang tidak terisi)</i>	<i>POST, http://localhost:3000/api/students</i>	Akan menghasilkan respon <i>Code: 400, Message: "BAD REQUEST."</i> , dan <i>Data: "Terdapat data yang belum diisi"</i> .	Sesuai atau tidak sesuai.
3	<i>Insert data siswa lebih dari satu dengan menggunakan file CSV yang</i>	<i>POST, http://localhost:3000/api/students /all</i>	Akan menghasilkan respon <i>Code: 200, Message: "Berhasil menambahkan</i>	Sesuai atau tidak sesuai.

Skenario Uji		Kelola Data Siswa		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
	benar. (semua <i>field</i> terisi)		data siswa.”, dan <i>Data:</i> (seluruh data yang di- <i>input</i>).	
4	Mengubah data siswa.	<i>PUT</i> , <i>http://localhost:3000/api/students</i> <i>/:studentId</i>	Akan menghasilkan respon <i>Code:</i> 200, <i>Message:</i> “Data berhasil diperbarui.”, dan <i>Data:</i> (data siswa yang telah diubah).	Sesuai atau tidak sesuai.
5	Menampilkan seluruh data siswa berdasarkan kelompok kelas.	<i>GET</i> , <i>http://localhost:3000/api/students</i> <i>/group/:groupName</i>	Akan menghasilkan respon <i>Code:</i> 200, <i>Message:</i> “OK.”, dan <i>Data:</i> (seluruh data siswa berdasarkan kelompok kelas).	Sesuai atau tidak sesuai.

Mirip dengan Tabel 4.4, Tabel 4.5 menyajikan skenario pengujian untuk mengelola data siswa. Skenario ini mencakup operasi *input*, *update*, dan menampilkan data siswa. Tujuannya adalah untuk memverifikasi fungsionalitas sistem dalam mengelola data siswa secara menyeluruh dan presisi.

Tabel 4. 6 Rancangan Pengujian - Kelola Data Kehadiran

Skenario Uji		Kelola Data Kehadiran		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
1	<i>Input data kehadiran dengan benar. (semua field terisi)</i>	<i>POST, http://localhost:3000/api/presences</i>	Akan menghasilkan respon <i>Code: 200, Message: "Berhasil menambahkan data kehadiran", dan Data: (data yang di-input)</i>	Sesuai atau tidak sesuai.
2	<i>Input data kehadiran dengan salah. (terdapat field yang tidak terisi)</i>	<i>POST, http://localhost:3000/api/presences</i>	Akan menghasilkan respon <i>Code: 401, Message: "BAD REQUEST", dan Data: "Terdapat data yang belum diisi".</i>	Sesuai atau tidak sesuai.

Skenario Uji		Kelola Data Kehadiran		
No.	Pengujian	<i>HTTP Method/End Point</i>	Hasil Yang Diharapkan	Kesimpulan
3	Mengubah data kehadiran.	<i>PUT,</i> <i>http://localhost:3000/api/presences/:presenceId</i>	Akan menghasilkan respon <i>Code: 200, Message: "Data berhasil diperbarui."</i> , dan <i>Data:</i> (data kehadiran yang telah diubah).	Sesuai atau tidak sesuai.
4	Menampilkan seluruh data kehadiran berdasarkan id dan periode yang tersedia di <i>database</i> .	<i>GET,</i> <i>http://localhost:3000/api/presence/detail/:studentId/:periode</i>	Akan menghasilkan respon <i>Code: 200, Message: "OK"</i> , dan <i>Data:</i> (data kehadiran berdasarkan id dan periode).	Sesuai atau tidak sesuai.

Tabel 4.6 berfokus pada pengujian fungsionalitas sistem dalam mengelola data kehadiran siswa. Skenario pengujian yang dimuat mencakup operasi *input*, *update*, dan menampilkan data kehadiran. Diharapkan melalui skenario ini, sistem dapat terverifikasi kemampuannya dalam mengelola data kehadiran siswa secara menyeluruh dan presisi.

Tabel 4. 7 Rancangan Pengujian - Kelola Data Penilaian

Skenario Uji		Kelola Data Penilaian		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
1	Input data penilaian dengan benar. (semua <i>field</i> terisi)	POST, <i>http://localhost:3000/api/evaluations</i>	Akan menghasilkan respon <i>Code</i> : 200, <i>Message</i> : “Berhasil menambahkan data penilaian”, dan <i>Data</i> : (data yang di-input)	Sesuai atau tidak sesuai.
2	Input data penilaian dengan salah. (terdapat <i>field</i> yang tidak terisi)	POST, <i>http://localhost:3000/api/evaluations</i>	Akan menghasilkan respon <i>Code</i> : 401, <i>Message</i> : “BAD REQUEST”, dan <i>Data</i> : “Terdapat data yang belum diisi”.	Sesuai atau tidak sesuai.
3	Mengubah data penilaian.	PUT, <i>http://localhost:3000/api/evaluations/:evaluationId</i>	Akan menghasilkan respon <i>Code</i> : 200, <i>Message</i> : “Data berhasil	Sesuai atau tidak sesuai.

Skenario Uji		Kelola Data Penilaian		
No.	Pengujian	<i>HTTP Method/End Point</i>	Hasil Yang Diharapkan	Kesimpulan
			diperbarui.” dan <i>Data:</i> (data penilaian yang telah diubah)	
4	Menampilkan seluruh data penilaian berdasarkan id dan periode yang tersedia di <i>database</i> .	<i>GET,</i> <i>http://localhost:3000/api/evaluati</i> <i>on/detail/:studentId/:periode</i>	Akan menghasilkan respon <i>Code:</i> 200, <i>Message:</i> “OK” dan <i>Data:</i> (data penilaian berdasarkan id dan periode yang dipilih)	Sesuai atau tidak sesuai.

Tabel 4.7 Rancangan Pengujian – Kelola Data Penilaian berfokus pada pengujian fungsionalitas sistem dalam mengelola data nilai siswa. Skenario pengujian yang dimuat mencakup operasi input, update, dan menampilkan data kehadiran. Diharapkan melalui skenario ini, sistem dapat terverifikasi kemampuannya dalam mengelola data kehadiran siswa secara menyeluruh dan presisi.

Tabel 4. 8 Rancangan Pengujian - Kelola Data Tagihan SPP

Skenario Uji		Kelola Data Tagihan SPP		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
1	Input data pembayaran tagihan SPP dengan benar. (semua <i>field</i> terisi)	<i>POST</i> , <i>http://localhost:3000/api/bills</i>	Akan menghasilkan respon <i>Code</i> : 200, <i>Message</i> : “Berhasil menambahkan data pembayaran tagihan SPP”, dan <i>Data</i> : (data yang di-input)	Sesuai atau tidak sesuai.
2	Input data pembayaran tagihan SPP salah benar. (terdapat <i>field</i> yang tidak terisi)	<i>POST</i> , <i>http://localhost:3000/api/bills</i>	Akan menghasilkan respon <i>Code</i> : 401, <i>Message</i> : “ <i>BED REQUEST</i> ”, dan <i>Data</i> : “Terdapat data yang belum diisi”.	Sesuai atau tidak sesuai.
3	Mengubah data pemabayaran tagihan SPP	<i>PUT</i> , <i>http://localhost:3000/api/bills/:bil</i> <i>Id</i>	Akan menghasilkan respon <i>Code</i> : 200, <i>Message</i> :	Sesuai atau tidak sesuai.

Skenario Uji		Kelola Data Tagihan SPP		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
			“Data berhasil diperbarui”, dan <i>Data</i> : (data pembayaran tagihan SPP yang telah diubah)	
4	Menampilkan seluruh data tagihan berdasarkan id dan periode yang tersedia di <i>database</i> .	<i>GET</i> , <i>http://localhost:3000/api/bill/detail/:studentId/:periode</i>	Akan menghasilkan respon <i>Code</i> : 200, <i>Message</i> : “OK” dan <i>Data</i> : (data tagihan berdasarkan id dan periode yang dipilih)	Sesuai atau tidak sesuai.

Tabel 4.8 berfokus pada pengujian fungsionalitas sistem dalam mengelola data tagihan siswa. Skenario pengujian yang dimuat mencakup operasi input, update, dan menampilkan data kehadiran. Diharapkan melalui skenario ini, sistem dapat terverifikasi kemampuannya dalam mengelola data kehadiran siswa secara menyeluruh dan presisi.

Tabel 4. 9 Rancangan Pengujian - Kelola Data Dengan Akun Unauthorized

Skenario Uji		Kelola Data Dengan Akun Unauthorized		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
1	Input data guru dengan akun guru atau siswa	POST, <i>http://localhost:3000/api/teachers</i>	Akan menghasilkan respon Code: 403, Message: "FORBIDDEN", dan Data: "You are not authorized to access this route"	Sesuai atau tidak sesuai
2	Input data nilai dengan akun siswa	POST, <i>http://localhost:3000/api/evaluati ons</i>	Akan menghasilkan respon Code: 403, Message: "FORBIDDEN", dan Data: "You are not authorized to access this route"	Sesuai atau tidak sesuai
3	Input data kehadiran dengan akun siswa	POST, <i>http://localhost:3000/api/presenc es</i>	Akan menghasilkan respon Code: 403, Message: "FORBIDDEN", dan Data:	Sesuai atau tidak sesuai

Skenario Uji		Kelola Data Dengan Akun Unauthorized		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
			"You are not authorized to access this route"	
4	Input data pembayaran SPP tagihan dengan akun siswa	POST, <i>http://localhost:3000/api/bills</i>	Akan menghasilkan respon Code: 403, Message: "FORBIDDEN", dan Data: "You are not authorized to access this route"	Sesuai atau tidak sesuai
5	Input data siswa dengan akun siswa	POST, <i>http://localhost:3000/api/students</i>	Akan menghasilkan respon Code: 403, Message: "FORBIDDEN", dan Data: "You are not authorized to access this route"	Sesuai atau tidak sesuai

Berbeda dengan tabel-tabel sebelumnya yang menguji skenario pengelolaan data dengan akun yang memiliki hak akses, Tabel 4.9 Rancangan Pengujian -

Kelola Data dengan Akun *Unauthorized* berfokus pada pengujian akses data oleh akun yang tidak memiliki kewenangan. Skenario ini dirancang untuk memastikan sistem dapat membatasi akses pada halaman-halaman tertentu bagi pengguna yang tidak memiliki izin yang sesuai.

4.2 Implementasi Bahasa Pemrograman Go

Pada tahap ini, penelitian ini akan menyajikan sejumlah contoh penerapan kode pemrograman menggunakan bahasa Go.

4.2.1 Middleware

Middleware ini bertujuan sebagai lapisan manajemen data dan hak akses untuk sistem tergantung *role* pengguna.

```
// middleware untuk admin
func AuthMiddlewareAdmin(next http.Handler) http.Handler {
    return func(w http.ResponseWriter, r *http.Request, ps http.Primitive) {
        c, err := r.Cookie("token")
        if err != nil {
            if err == http.ErrNoCookie {
                w.Header().Set("Content-Type", "application/json")
                w.WriteHeader(http.StatusUnauthorized)
                response := map[string]string{"message": "Unauthorized"}
                helper.WriteToResponseBody(w, response)
                return
            }
        }

        tokenString := c.Value
        claims := &config.JWTClaims{}

        token, err := jwt.ParseWithClaims(tokenString, claims, func(t *jwt.Token) (interface{}, error) {
            return config.JWT_KEY, nil
        })

        if err != nil {
            v, _ := err.(*jwt.ValidationError)
            switch v.Errors {
            case jwt.ValidationErrorSignatureInvalid:
                w.Header().Set("Content-Type", "application/json")
                w.WriteHeader(http.StatusUnauthorized)
                response := map[string]interface{}{"status": http.StatusUnauthorized, "message": "Unauthorized"}
                helper.WriteToResponseBody(w, response)
                return
            case jwt.ValidationErrorExpired:
                w.Header().Set("Content-Type", "application/json")
                w.WriteHeader(http.StatusUnauthorized)
                response := map[string]interface{}{"status": http.StatusUnauthorized, "message": "Unauthorized"}
                helper.WriteToResponseBody(w, response)
                return
            }
        }

        next.ServeHTTP(w, r)
    }
}
```

Gambar 4. 14 Middleware

4.2.2 Error Handler

Error handler berfungsi untuk menangani kesalahan yang terjadi pada program. Secara umum, jika terjadi kesalahan atau *error* dalam program, sistem akan berhenti beroperasi. Oleh karena itu, *error handler* dibuat untuk mengatasi

masalah tersebut, sehingga ketika terjadi *error*, program tetap berjalan dan sistem akan menampilkan pesan yang menjelaskan penyebab *error* tersebut.

```
func validationError(w http.ResponseWriter, err interface{}) bool {
    exception, ok := err.(validator.ValidationErrors)

    if ok {
        w.Header().Set("Content-Type", "application/json")
        w.WriteHeader(http.StatusBadRequest)

        webResponse := web.WebResponse{
            Code:    http.StatusBadRequest,
            Message: "BAD REQUEST",
            Data:    strings.Split(exception.Error(), ":")[2],
        }

        helper.WriteToResponseBody(w, webResponse)
        return true
    } else {
        return false
    }
}

func notFoundError(w http.ResponseWriter, err interface{}) bool {
    exception, ok := err.(NotFoundErr)

    if ok {
        w.Header().Set("Content-Type", "application/json")
        w.WriteHeader(http.StatusNotFound)

        webResponse := web.WebResponse{
            Code:    http.StatusNotFound,
            Message: "NOT FOUND",
            Data:    exception.Error,
        }
    }
}
```

Gambar 4. 15 Error Handler

4.3 Implementasi *API* Sistem Informasi Akademik RA Amanah School

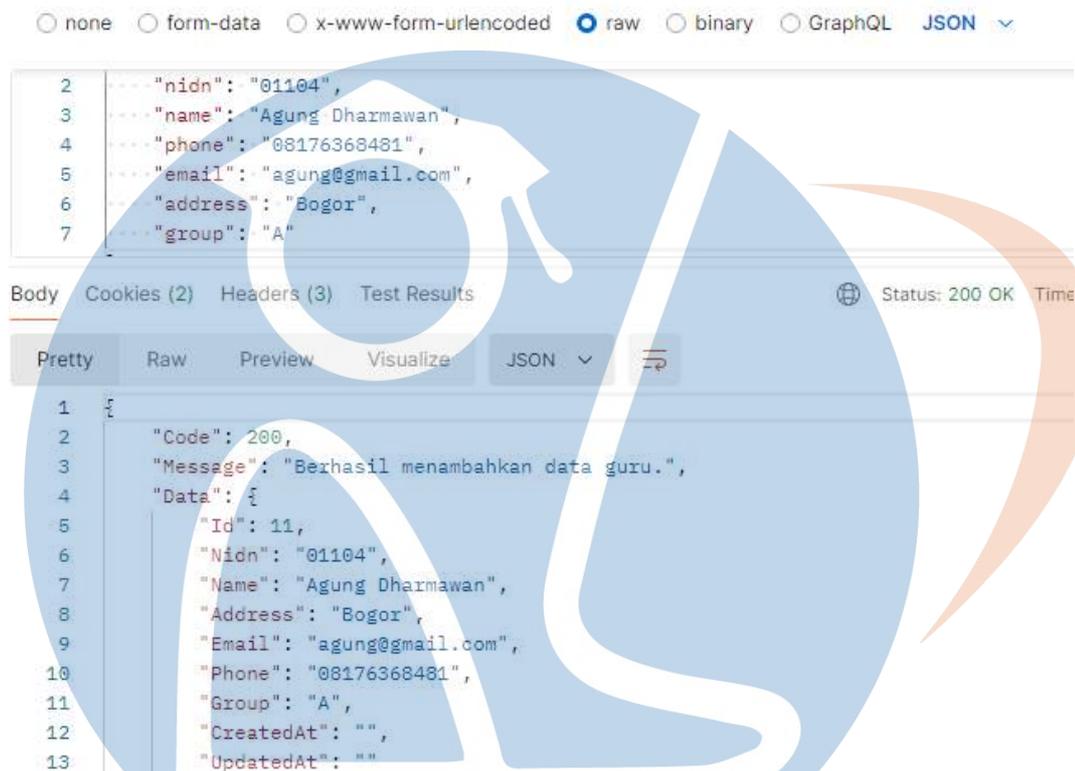
Implementasi *API* Sistem Informasi Akademik di RA Amanah School adalah upaya untuk memanfaatkan teknologi dalam pengelolaan informasi akademik sehingga dapat diakses dengan mudah kapan saja dan di mana saja. Langkah ini merupakan realisasi dari perancangan yang telah dibuat, menjadi sebuah aplikasi backend yang dikembangkan menggunakan bahasa pemrograman Go.

4.3.1 Antarmuka Kelola Data Guru

Pada tahap implementasi antarmuka pengelolaan data guru ini, peneliti akan menyajikan uji implementasi sistem layanan *REST-API* untuk mengelola data guru.

Penjelasan akan difokuskan pada *request* yang dimasukkan dan *respons* yang diterima saat pengguna berinteraksi dengan sistem.

1. *Input Data Guru Dengan Benar*



Gambar 4. 16 *Input Data Guru Dengan Benar*

Pada gambar 4. 19 *input data guru dengan benar*, merupakan implementasi *request* dan *response* dari API dimana admin menambahkan data guru. Ketika data guru berhasil ditambahkan, maka sistem akan otomatis membuat akun *user* untuk guru tersebut dengan *role teacher*.

	id	username	password	role
Hapus	88	admin	\$2a\$10\$PMOEyYFTmGg2K90FFzF8l.SRdEFW4811lGdXVuEFM91...	admin
Hapus	89	daden@gmail.com	\$2a\$10\$m4vVrH6oszlqU1NDVC06ReFlvsNlFXzDi9V1IbTflry...	teacher
Hapus	90	Ahoy@gmail.com	\$2a\$10\$q82/5if7rFEDpYSKEis0EeiKpt2wM4pRU3N5jFvOT0h...	student
Hapus	110	angga@gmail.com	\$2a\$10\$5ldJM0xi.GNKMqG5E5YviZugrXGWDFoU19Xg9VmKY7eD...	teacher
Hapus	112	agung@gmail.com	\$2a\$10\$rtYupPuywd8zba1ObnKzxO8V9YH8kflRsaWdCLSZDMz...	teacher

Gambar 4. 17 *Akun User Teacher*

Dengan akun guru yang telah terdaftar, guru tersebut dapat mengakses Sistem Informasi Akademik RA Amanah School untuk mengelola data siswa, presensi siswa, nilai siswa, dan tagihan SPP siswa.

2. *Input Data Guru Dengan Salah*

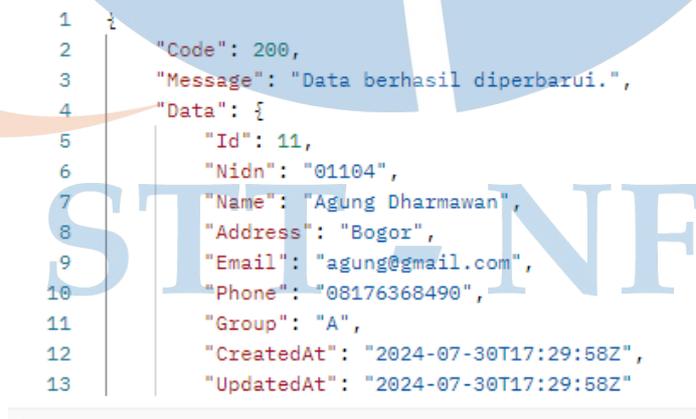


```
3 {
4   "name": "Bunga Dharmawan",
5   "phone": "08176368481",
6   "email": "agung@gmail.com",
7   "address": "Bogor",
8   "group": ""
9 }
10
11 Status: 400 Bad Request Time: 4 ms Size: 194 B
12
13 Pretty Raw Preview Visualize JSON
14
15 {
16   "Code": 400,
17   "Message": "BAD REQUEST",
18   "Data": "Terdapat data yang belum diisi"
19 }
```

Gambar 4. 18 *Input Data Guru Dengan Salah*

Pada gambar 4. 21 *input data guru dengan salah*, merupakan implementasi *request* dan *response* dari *API* dimana admin menambahkan data guru. Akan tetapi, terdapat *field* yang belum terisi, sehingga sistem memberikan *response* “Terdapat data yang belum di isi”.

3. *Mengubah Data Guru*

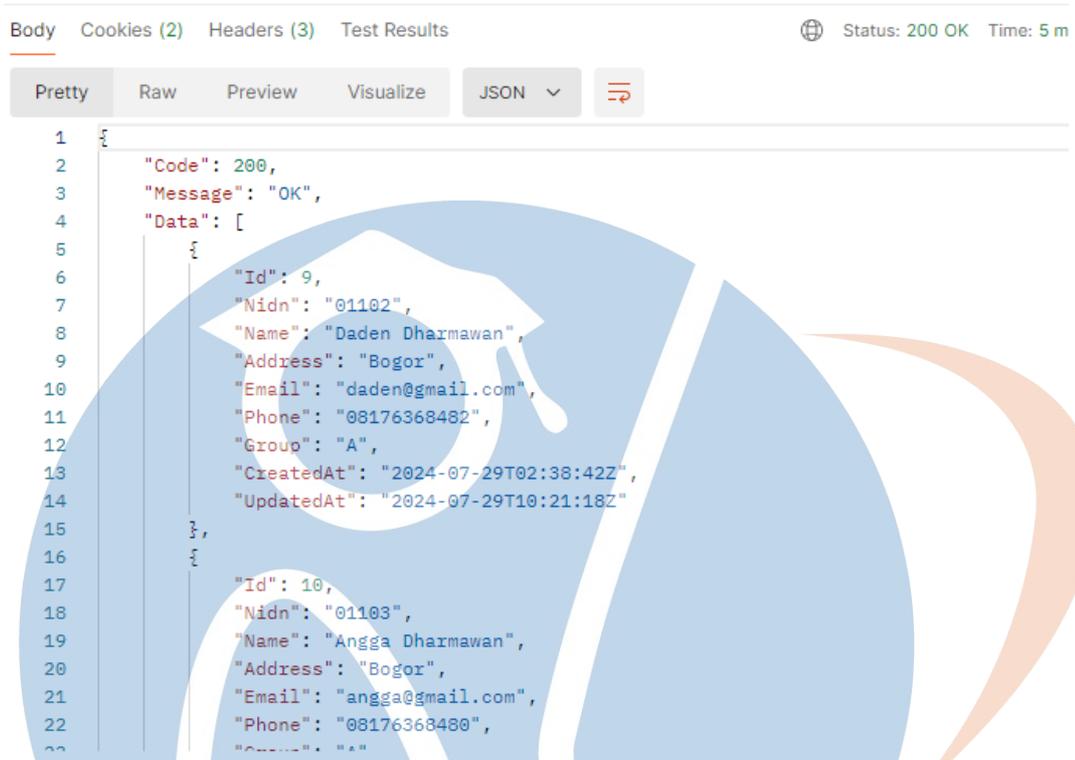


```
1 {
2   "Code": 200,
3   "Message": "Data berhasil diperbarui.",
4   "Data": {
5     "Id": 11,
6     "Nidn": "01104",
7     "Name": "Agung Dharmawan",
8     "Address": "Bogor",
9     "Email": "agung@gmail.com",
10    "Phone": "08176368490",
11    "Group": "A",
12    "CreatedAt": "2024-07-30T17:29:58Z",
13    "UpdatedAt": "2024-07-30T17:29:58Z"
14  }
```

Gambar 4. 19 *Mengubah Data Guru*

Gambar di atas merupakan implementasi *request* dan *response* dari *API* untuk mengubah data guru.

4. Menampilkan seluruh data guru



```
1 {
2   "Code": 200,
3   "Message": "OK",
4   "Data": [
5     {
6       "Id": 9,
7       "Nidn": "01102",
8       "Name": "Daden Dharmawan",
9       "Address": "Bogor",
10      "Email": "daden@gmail.com",
11      "Phone": "08176368482",
12      "Group": "A",
13      "CreatedAt": "2024-07-29T02:38:42Z",
14      "UpdatedAt": "2024-07-29T10:21:18Z"
15    },
16    {
17      "Id": 10,
18      "Nidn": "01103",
19      "Name": "Angga Dharmawan",
20      "Address": "Bogor",
21      "Email": "angga@gmail.com",
22      "Phone": "08176368480",
23      "Group": "A"
24    }
25  ]
26 }
```

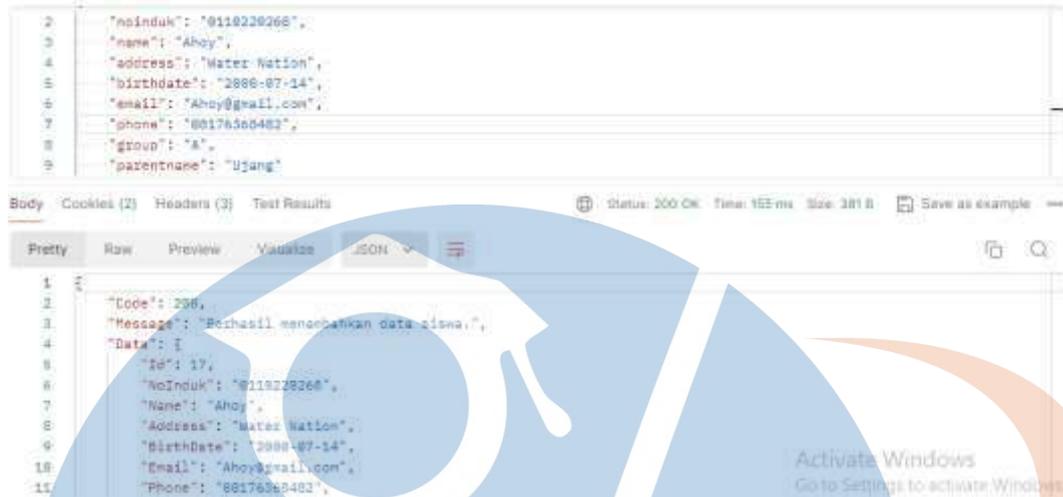
Gambar 4. 20 Menampilkan Seluruh Data Guru

Pada gambar 4. 25 menampilkan seluruh data guru, merupakan implementasi dari API yang berfungsi untuk menampilkan seluruh data guru-guru yang terdaftar.

4.3.2 Antarmuka Kelola Data Siswa

Pada tahap implementasi antarmuka pengelolaan data siswa ini, peneliti akan menyajikan uji implementasi sistem layanan *REST-API* untuk mengelola data siswa. Penjelasan akan difokuskan pada *request* yang dimasukkan dan *respons* yang diterima saat pengguna berinteraksi dengan sistem.

1. *Input* Data Siswa Dengan Benar



Gambar 4. 21 *Input* Data Siswa Dengan Benar

Pada gambar 4. 26 *input* data siswa dengan benar, merupakan implementasi *request* dan *response* dari API dimana admin atau guru menambahkan data siswa. Sama halnya pada saat menambahkan data guru, ketika menambahkan data siswa, sistem akan secara otomatis membuat akun *user* baru berdasarkan data yang dimasukkan.

	id	username	password	role
Hapus	88	admin	\$2a\$10\$PMOEyYFTmGg2K90FFzF8I.SRdEFW49I1IGdXVvUEFM91...	admin
Hapus	89	daden@gmail.com	\$2a\$10\$m4vVrH8oszIqU1NDVC06ReFlvsNfXzDj9V11bTflry...	teacher
Hapus	90	Ahoy@gmail.com	\$2a\$10\$q82/5f7rFEDpYSKEis0EeiKpt2wM4pRU3N5jFvOT6h...	student
Hapus	110	angga@gmail.com	\$2a\$10\$5ldJM0x/.GNKMqgE5YviZugrXGWDFOU19Xg9VmKY7eD...	teacher
Hapus	112	agung@gmail.com	\$2a\$10\$rtYupPuywd8zba1ObnKzxO8V9YH8kfiRsaWdCLSZDMz...	teacher
Hapus	114	Akoy@gmail.com	\$2a\$10\$NNCviEoSIX1tkp9FjLXr.My1g8sCs17Lxx8f61cB9K...	student

Gambar 4. 22 Akun *User Student*

Dengan terbuatnya akun *user student*, maka wali murid dapat mengakses sistem informasi akademik RA. Amanah School untuk memantau informasi terkait presensi, nilai, dan tagihan siswa.

2. *Input Data Siswa Dengan Salah*



Gambar 4. 23 *Input Data Siswa Dengan Salah*

Gambar di atas menunjukkan contoh *request* dan *response* saat terdapat data kosong saat pengisian data siswa. Dalam situasi ini, *API* akan mengirimkan pesan "Terdapat data yang belum diisi". Pada contoh tersebut, data yang kosong adalah nama orang tua atau wali murid dari siswa.

3. *Input Data Siswa Lebih Dari Satu Menggunakan File CSV*



Gambar 4. 24 *Input Data Siswa Lebih Dari Satu Menggunakan File CSV*

Jika sebelumnya pada gambar 4. 26, *input* data siswa dilakukan dengan benar satu per satu, pada gambar 4. 29 menunjukkan contoh penerapan *API* yang dapat meng-*input* data siswa lebih dari satu sekaligus menggunakan *file CSV*. Hal ini tentu sangat membantu admin atau guru dalam menyelesaikan *input* data siswa dengan lebih cepat.

4. Mengubah Data Siswa

id	no_induk	name	address	birth_date	email	phone	group_class	parent_name	created_at	updated_at
15	0110220268	Ahay	Water Nation	2000-07-15	ahay@gmail.com	08178368482	A	Cikuy	2024-07-29 02:42:40	2024-07-29 02:43:44
17	0110220268	Ahoy	Water Nation	2000-07-14	Ahoy@gmail.com	08178368482	A	Ujang	2024-08-08 14:53:41	2024-08-08 14:53:41
21	011012126	Aman	Cikutu	2000-10-01	aman@gmail.com	0813455454	A	Unih	2024-08-08 14:58:56	2024-08-08 14:58:56
22	011013098	Ucuy	Citoke	2000-10-02	Ucuy@gmail.com	0813423487	A	Atuy Mekmoy	2024-08-08 14:58:56	2024-08-08 14:58:56
23	011014134	Amah	Ciburuy	2000-10-03	Amah@gmail.com	0827273123	B	Aming	2024-08-08 14:58:56	2024-08-08 14:58:56
24	011015334	Ahum	Ciletuh	2000-10-04	Ahum@gmail.com	0817767676	B	Imin	2024-08-08 14:58:56	2024-08-08 14:58:56
25	011016241	Bubun	Cikutamahi	2000-10-05	Bubun@gmail.com	0812337676	A	Baban	2024-08-08 14:58:56	2024-08-08 14:58:56

Gambar 4. 25 Data Siswa

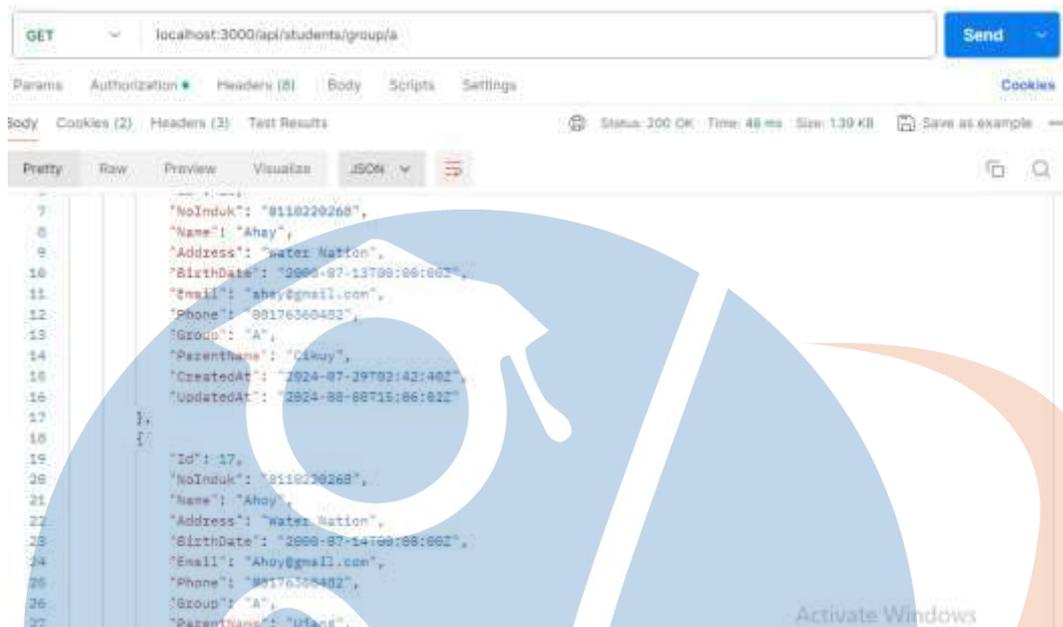
Pada Gambar 4.34 Data Siswa yang berisi informasi mengenai beberapa siswa. Pengujian akan dilakukan dengan mengubah tanggal lahir siswa dengan id 15.

```
3  "Name": "Ahay",
4  "Address": "Water Nation",
5  "BirthDate": "2000-07-13",
Body  Cookies (2)  Headers (3)  Test Results
Status: 200 OK  Time: 103 ms  Size: 414 B  Save as example
Pretty  Raw  Preview  Visualize  JSON
2  "Code": 200,
3  "Message": "Data berhasil diperbarui.",
4  "Data": {
5  "Id": 15,
6  "NoInduk": "0110220268",
7  "Name": "Ahay",
8  "Address": "Water Nation",
9  "BirthDate": "2000-07-13",
10 "Email": "ahay@gmail.com",
11 "Phone": "08178368482",
12 "Group": "A",
13 "ParentName": "Cikuy",
14 "CreatedAt": "2024-07-29T02:42:40Z",
15 "UpdatedAt": "2024-07-29T02:43:44Z"
16 }
```

Gambar 4. 26 Mengubah Data Siswa

Gambar di atas merupakan *request* dan *response* dari implementasi *API* untuk mengubah data siswa. Tanggal lahir siswa dengan id 15 yang sebelumnya 2000-07-15 diubah menjadi 2000-07-13.

5. Menampilkan Seluruh Data Siswa Berdasarkan Kelompok Kelas



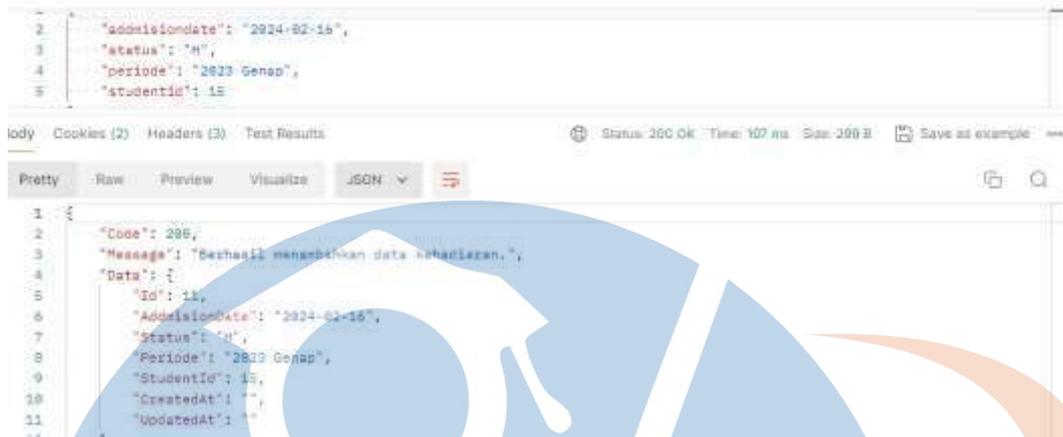
Gambar 4. 27 Menampilkan Seluruh Data Siswa Berdasarkan Kelompok Kelas

Gambar 4. 36 Menampilkan Seluruh Data Siswa Berdasarkan Kelompok Kelas merupakan contoh *response API* dari hasil *request* untuk menampilkan data siswa berdasarkan kelompok kelasnya. Hal ini dapat memudahkan para guru untuk mengidentifikasi para siswa yang berada dalam kelompok kelas tertentu.

4.3.3 Antarmuka Kelola Data Presensi

Tahap implementasi antarmuka pengelolaan data presensi akan diuji dengan mendemonstrasikan fungsionalitas sistem layanan *REST-API* dalam mengelola data presensi. Pengujian akan berfokus pada *request* yang dimasukkan oleh pengguna dan *response* yang diterima oleh sistem sebagai hasil dari interaksi pengguna.

1. *Input Data Presensi Dengan Benar*



```
2 | "admissiondate": "2024-02-16",
3 | "status": "H",
4 | "periode": "2023 Genap",
5 | "studentid": 15

Body Cookies (2) Headers (3) Test Results Status: 200 OK Time: 107 ms Size: 266 B Save as example

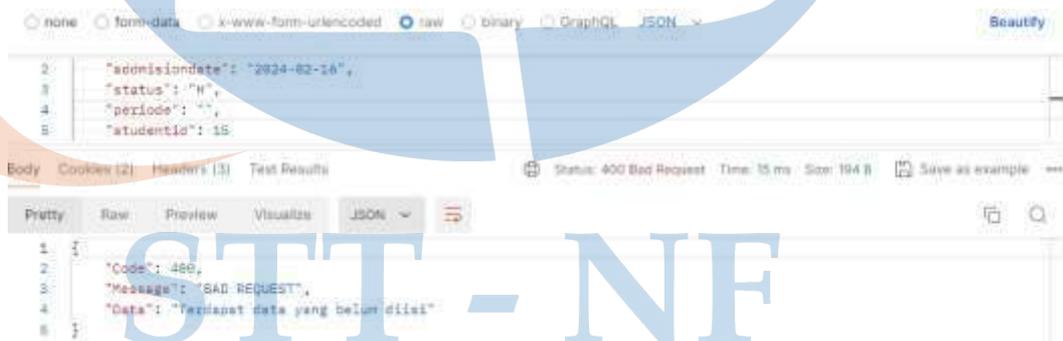
Pretty Raw Preview Visualize JSON

1 |
2 | {
3 |   "Code": 200,
4 |   "Message": "Berhasil menambahkan data kehadiran.",
5 |   "Data": [
6 |     {
7 |       "Id": 11,
8 |       "AdmissionDate": "2024-02-16",
9 |       "Status": "H",
10 |      "Periode": "2023 Genap",
11 |      "StudentId": 15,
12 |      "CreatedAt": "",
13 |      "UpdatedAt": ""
14 |    }
15 |  ]
16 | }
```

Gambar 4. 28 *Input Data Presensi Dengan Benar*

Gambar tersebut menunjukkan *response API* dari sistem informasi akademik RA. Amanah School ketika terjadi *request* untuk menambahkan data presensi baru. *Response* tersebut memuat informasi mengenai status permintaan, yaitu berhasil dengan *code* 200. Selain itu, *response* juga menyertakan data presensi yang baru ditambahkan, seperti ID presensi, tanggal presensi, ID siswa, status kehadiran, dan periode presensi.

2. *Input Data Kehadiran Dengan Salah*



```
2 | "admissiondate": "2024-02-16",
3 | "status": "H",
4 | "periode": "",
5 | "studentid": 15

Body Cookies (2) Headers (3) Test Results Status: 400 Bad Request Time: 15 ms Size: 194 B Save as example

Pretty Raw Preview Visualize JSON

1 |
2 | {
3 |   "Code": 400,
4 |   "Message": "BAD REQUEST",
5 |   "Data": "Terdapat data yang belum diisi"
6 | }
```

Gambar 4. 29 *Input Data Kehadiran Dengan Salah*

Pada Gambar 4.34 *Input Data Kehadiran Dengan Salah* menunjukkan *response* dari hasil *request* untuk menambahkan data presensi, tetapi terdapat data yang kosong atau tidak terisi.

3. Mengubah Data Presensi

id	admission_date	status	periode	student_id	created_at	updated_at
10	2024-01-12	I	2023 Genap	15	2024-07-29 02:50:03	2024-07-29 02:51:50
11	2024-02-16	H	2023 Genap	15	2024-08-08 15:17:20	2024-08-08 15:17:20

Gambar 4. 30 Daftar Presensi

Gambar di atas menunjukkan daftar kehadiran yang tersedia di dalam *database*. Untuk menguji *API* yang berfungsi mengubah data kehadiran, pengujian akan dilakukan untuk mengubah status yang tadinya Izin (I) menjadi H (Hadir) pada id 10.

```
1 {
2   "Code": 200,
3   "Message": "Data berhasil diperbarui.",
4   "Data": {
5     "Id": 10,
6     "AdmissionDate": "2024-01-12",
7     "Status": "H",
8     "Periode": "2023 Genap",
9     "StudentId": 15,
10    "CreatedAt": "2024-07-29T02:50:03Z",
11    "UpdatedAt": "2024-07-29T02:51:50Z"
12  }
}
```

Gambar 4. 31 Mengubah Data Presensi

Kemudian, pada Gambar 4.36 Mengubah Data Presensi merupakan *response* dari hariil *request* untuk mengubah data presensi.

4. Menampilkan Data Presensi Berdasarkan Id dan Periode

```
1 {
2   "Code": 200,
3   "Message": "OK",
4   "Data": [
5     {
6       "Id": 10,
7       "AdmissionDate": "2024-01-12T00:00:00Z",
8       "Status": "H",
9       "Periode": "2023 Genap",
10      "StudentId": 15,
11      "CreatedAt": "2024-07-29T02:50:03Z",
12      "UpdatedAt": "2024-08-08T15:17:20Z"
13    },
14  ],
15  {
16    "Id": 11,
17    "AdmissionDate": "2024-02-16T00:00:00Z",
18    "Status": "H",
19    "Periode": "2023 Genap",
20    "StudentId": 15,
21    "CreatedAt": "2024-08-08T15:17:20Z",
22    "UpdatedAt": "2024-08-08T15:17:20Z"
23  }
}
```

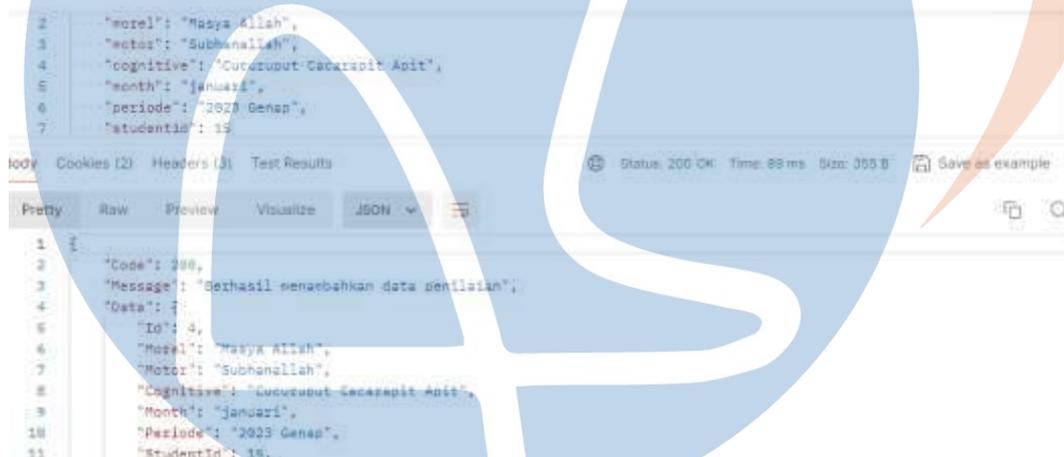
Gambar 4. 32 Menampilkan Data Presensi Berdasarkan Id dan Periode

Informasi yang ditampilkan dalam gambar di atas merupakan hasil dari *request API* untuk menampilkan detail kehadiran siswa. Orang tua siswa dapat menggunakan informasi ini untuk melihat daftar kehadiran siswa di setiap periode semester.

4.3.4 Antarmuka Kelola Data Penilaian

Pada bagian ini merupakan implementasi antarmuka pengelolaan data nilai, fungsionalitas sistem layanan *REST-API* dalam mengelola data nilai akan diuji. Pengujian ini akan berfokus pada *request* yang dimasukkan oleh pengguna dan *response* yang diterima oleh sistem sebagai hasil dari interaksi pengguna.

1. *Input Data Penilaian Dengan Benar*



```
2  {"moral": "Masya Allah",
3  "moral": "Subhanallah",
4  "cognitive": "Cucuruput-Cecarapit-Abit",
5  "month": "Januari",
6  "periode": "2023 Genap",
7  "studentId": 15}

100% Cookies (2) Headers (3) Test Results Status: 200 OK Time: 88 ms Size: 353 B Save as example

Pretty Raw Preview Visualize JSON

1  {
2  "Code": 200,
3  "Message": "Berhasil menambahkan data penilaian",
4  "Data": {
5  "Id": 4,
6  "Moral": "Masya Allah",
7  "Moral": "Subhanallah",
8  "Cognitive": "Cucuruput-Cecarapit-Abit",
9  "Month": "Januari",
10 "Periode": "2023 Genap",
11 "StudentId": 15,
```

Gambar 4. 33 *Input Data Nilai Dengan Benar*

Gambar 4.38 *Input Data Nilai Dengan Benar* menunjukkan hasil implementasi dari permintaan *API* untuk menambahkan data nilai. Hal ini berarti data nilai baru telah berhasil ditambahkan ke dalam sistem.

2. Input Data Presensi Dengan Salah



Gambar 4. 34 Input Data Presensi Dengan Salah

Gambar di atas menunjukkan *response* yang muncul ketika ada data yang kosong atau tidak terisi saat melakukan *request* untuk menambahkan data baru. Hal ini menandakan bahwa proses penambahan data baru tidak dapat dilakukan karena data yang diperlukan tidak lengkap.

3. Mengubah Data Nilai

id	morel	motor	cognitive	month	periode	student_id
3	Allahumma Baarik	Subhanallah	Cucuruput Cacarapit Apit	februari	2023 Genap	15
4	Masya Allah	Subhanallah	Cucuruput Cacarapit Apit	januari	2023 Genap	15

Gambar 4. 35 Data Nilai

Untuk implementasi *API* mengubah data siswa akan dilakukan kepada data nilai yang tersedia dalam database. Data yang akan diubah adalah data yang terdapat pada nilai id 3.



Gambar 4. 36 Mengubah Data Nilai

Hasil yang tertera dalam Gambar 4.41 Mengubah Data Nilai merupakan *response* dari hasil *request* untuk merubah data nilai siswa, agar data siswa relevan.

4. Menampilkan Data Penilaian Berdasarkan Id dan Periode

```
    "Id": 3,  
    "Morel": "Allahumma Baarik",  
    "Motor": "Masya Allah",  
    "Cognitive": "Cucuruput Cacarapit Apit",  
    "Month": "februari",  
    "Periode": "2023 Genap",  
    "StudentId": 15,  
    "CreatedAt": "2024-07-29T02:53:12Z",  
    "UpdatedAt": "2024-08-08T15:32:54Z"  
  },  
  {  
    "Id": 4,  
    "Morel": "Masya Allah",  
    "Motor": "Subhanallah",  
    "Cognitive": "Cucuruput Cacarapit Apit",
```

Gambar 4. 37 Menampilkan Data Penilaian Berdasarkan Id dan Periode

Gambar di atas adalah response dari request API untuk menampilkan detail nilai akademik siswa. Dengan adanya API tersebut, wali murid dapat melihat perkembangan nilai dari siswa setiap bulan berdasarkan periode atau tahun ajarannya.

4.3.5 Antarmuka Kelola Data Tagihan SPP

Pada tahap implementasi antarmuka untuk pengelolaan data penilaian ini, peneliti akan menyajikan hasil uji implementasi sistem layanan *REST-API* untuk manajemen data nilai. Fokus penjelasan akan berada pada *request* yang dimasukkan serta *response* yang diterima ketika pengguna berinteraksi dengan sistem tersebut.

1. Input Data Pembayaran Tagihan SPP Dengan Benar



The screenshot shows a REST client interface with a request body and a response body. The request body is a JSON object with the following fields: "month": "Mei", "bill": 500000, "periode": "2023 genap", and "studentId": 15. The response body is a JSON object with the following fields: "Code": 200, "Message": "Berhasil menambahkan data tagihan.", "Data": { "Id": 13, "Month": "Mei", "Bill": 500000, "PaidOff": 0, "Rest": 500000, "Periode": "2023 genap", "Status": "belum lunas", "StudentId": 15. The response status is 200 OK, and the response size is 331 B.

Gambar 4. 38 Input Data Pembayaran Tagihan SPP Dengan Benar

Gambar di atas menunjukkan proses penambahan data tagihan siswa menggunakan *API*. Permintaan (*request*) dikirimkan ke *API* dengan menyertakan informasi tagihan seperti bulan tagihan, nominal tagihan, periode, dan id siswa. *API* kemudian memproses permintaan dan mengirimkan respons (*response*) yang berisi data yang ditambahkan, serta pesan yang menjelaskan hasil operasi.

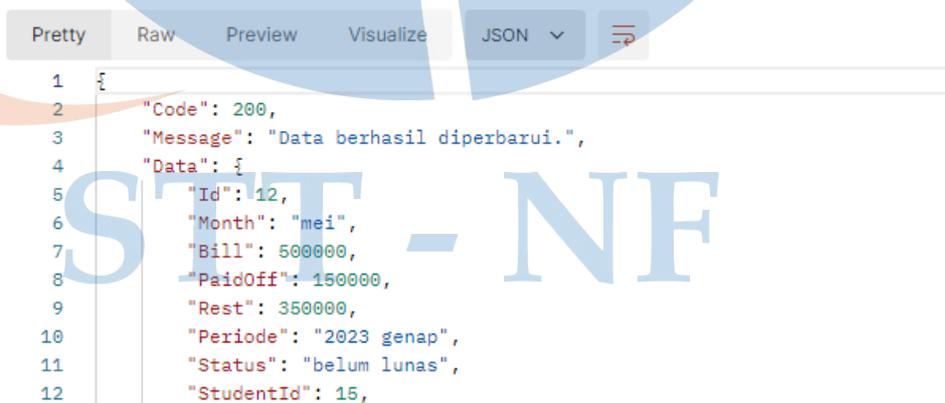
2. *Input Data Pembayaran Tagihan SPP Dengan Salah*



Gambar 4. 39 *Input Data Pembayaran Tagihan SPP Dengan Salah*

Pada Gambar 4.44 *Input Data Pembayaran Tagihan SPP*, merupakan implementasi *API* untuk menambahkan data tagihan SPP menunjukkan input data dengan beberapa kolom kosong. Akibatnya, data tidak dapat diproses, dan sistem akan mengirimkan *response* "Terdapat data yang belum diisi".

3. *Mengubah Data Pembayaran*



Gambar 4. 40 *Mengubah Data Pembayaran*

Gambar sebelumnya menampilkan implementasi *API* untuk mengubah data tagihan SPP. Perubahan data tagihan SPP dilakukan saat transaksi antara wali murid

dan guru. Guru akan mengubah data nominal pembayaran sesuai dengan jumlah yang dibayarkan oleh wali murid.

4. Menampilkan Data Tagihan SPP Berdasarkan Id dan Periode

```
"Id": 12,  
"Month": "mei",  
"Bill": 500000,  
"PaidOff": 150000,  
"Rest": 350000,  
"Periode": "2023 genap",  
"Status": "",  
"StudentId": 15,  
"CreatedAt": "2024-07-29T03:23:32Z",  
"UpdatedAt": "2024-08-08T15:39:03Z",  
},  
{  
"Id": 13,  
"Month": "mei",  
"Bill": 500000,  
"PaidOff": 0,  
"Rest": 500000,  
"Periode": "2023 genap",  
"Status": "belum lunas",  
"StudentId": 15,  
"CreatedAt": "2024-08-08T15:36:13Z",  
"UpdatedAt": "2024-08-08T15:36:13Z"
```

Gambar 4. 41 Menampilkan Data Tagihan SPP Berdasarkan Id dan Periode

Gambar 4.46 menunjukkan implementasi *API* untuk menampilkan data tagihan SPP berdasarkan Id dan Periode. *API* ini digunakan untuk menampilkan data tagihan yang sudah dibayar oleh wali murid. Dari respons tersebut, dapat diketahui apakah tagihan siswa untuk bulan tertentu sudah lunas.

4.3.6 Antarmuka Kelola Data Dengan Akun *Unauthorized*

Proses implementasi antarmuka pengelolaan data dengan akun *unauthorized* akan melibatkan pengujian terhadap sistem layanan *REST-API*. Pengujian ini bertujuan untuk memverifikasi kemampuan sistem dalam mengelola data dengan akun yang tidak memiliki kewenangan. Skenario pengujian yang dirancang akan memastikan bahwa sistem dapat membatasi akses pada halaman-halaman tertentu bagi pengguna yang tidak memiliki izin yang sesuai.

1. *Input Data Duru Dengan Akun Guru atau Siswa*

```
1  {
2  |  ... "username": "daden@gmail.com",
3  |  ... "password": "01102"
4  |  }
```

ody Cookies (2) Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```
1  {
2  |  "message": "Login Berhasil",
3  |  "status": 200
4  |  }
```

Gambar 4. 42 Login Dengan Akun Guru

Pertama, pengujian ini akan dilakukan dengan menggunakan akun guru terlebih dahulu. Pada gambar di atas merupakan implementasi *login* dengan akun guru.

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
2  ... "nidn": "01104",
3  ... "name": "Agung Dharmawan",
4  ... "phone": "08176368481",
5  ... "email": "agung@gmail.com",
6  ... "address": "Bogor",
7  ... "A"
```

Body Cookies (2) Headers (3) Test Results Status: 403 Forbidden Time: 3 ms

Pretty Raw Preview Visualize JSON

```
1  {
2  |  "message": "You are not authorized to access this route",
3  |  "status": 403
4  |  }
```

Gambar 4. 43 Input Data Guru Dengan Akun Guru

Pada Gambar 4.40 *Input Data Guru Dengan Akun Guru* merupakan contoh *response* yang akan didapat ketika mencoba *input* data guru menggunakan akun guru.

2. *Input Data Guru Dengan Akun Siswa*



```
1 {
2   ... "username": "Ahoy@gmail.com",
3   ... "password": "0110220268"
4 }
```

Body Cookies (2) Headers (5) Test Results Status: 200 OK Time: 195 ms

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Login Berhasil",
3   "status": 200
4 }
```

Gambar 4. 44 Login Dengan Akun Siswa

Kemudian, pengujian akan dilakukan dengan menggunakan akun siswa untuk *input* data guru. Pada gambar di atas merupakan implementasi *login* dengan akun siswa.



none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
2 {
3   ... "nidn": "01104",
4   ... "name": "Agung Dharmawan",
5   ... "phone": "08176368481",
6   ... "email": "agung@gmail.com",
7   ... "address": "Bogor",
8   ... "password": "0110220268"
9 }
```

Body Cookies (2) Headers (3) Test Results Status: 403 Forbidden Time: 3 ms

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "You are not authorized to access this route",
3   "status": 403
4 }
```

Gambar 4. 45 *Input Data Guru Dengan Akun Siswa*

Pada Gambar 4.42 *Input Data Guru Dengan Akun Siswa*, sistem memberikan *response* yang sama dengan ketika menambahkan data guru dengan akun guru.

3. Input Data Penilaian Dengan Akun Siswa

```
2 ... "morel": "Masya Allah",
3 ... "motor": "Subhanallah",
4 ... "cognitive": "",
5 ... "month": "januari",
6 ... "periode": "2023 Genap",
7 ... "studentid": 15
```

Body Cookies (2) Headers (3) Test Results Status: 403 Forbidden Time: 3 ms Size: 186 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "You are not authorized to access this route",
3   "status": 403
4 }
```

Gambar 4. 46 Input Data Penilaian Dengan Akun Siswa

Pada Gambar 4.43, implementasi penambahan data penilaian dengan akun siswa menghasilkan respon "You are not authorized to access this route". Hal ini menunjukkan bahwa sistem memblokir akses akun siswa untuk melakukan tindakan tersebut.

4. Input Data Kehadiran Dengan Akun Siswa

```
2 ... "addmisiondate": "2024-02-16",
3 ... "status": "H",
4 ... "periode": "",
5 ... "studentid": 15
```

Body Cookies (2) Headers (3) Test Results Status: 403 Forbidden Time: 3 ms Size: 186 B

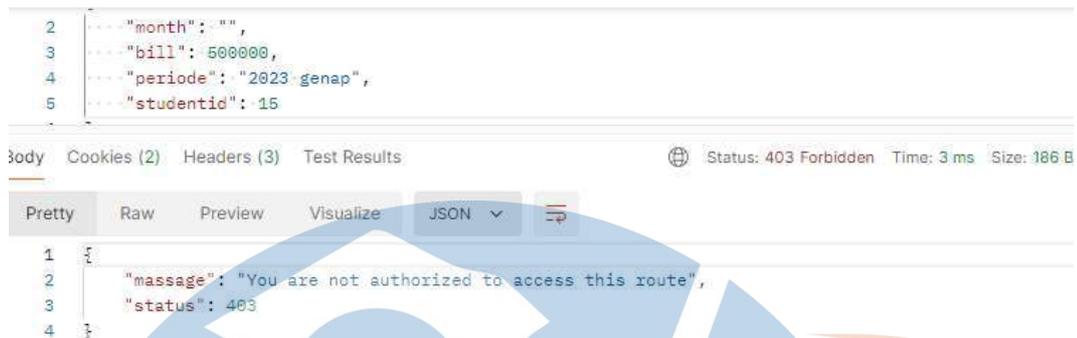
Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "You are not authorized to access this route",
3   "status": 403
4 }
```

Gambar 4. 47 Input Data Kehadiran Dengan Akun Siswa

Pada Gambar 4.44 Input Data Kehadiran Dengan Akun Siswa, implementasi penambahan data kehadiran dengan akun siswa menghasilkan respon "You are not authorized to access this route". Hal ini menunjukkan bahwa sistem memblokir akses akun siswa untuk melakukan tindakan tersebut.

5. Input Data Tagihan SPP Dengan Akun Siswa



```
2 {
3   "month": "",
4   "bill": 500000,
5   "periode": "2023 genap",
6   "studentid": 15
7 }
```

Body Cookies (2) Headers (3) Test Results Status: 403 Forbidden Time: 3 ms Size: 186 B

```
1 {
2   "message": "You are not authorized to access this route",
3   "status": 403
4 }
```

Gambar 4. 48 Input Data Tagihan SPP Dengan Akun Siswa

Pada gambar di atas merupakan implementasi penambahan data tagihan SPP dengan akun siswa menghasilkan respon *"You are not authorized to access this route"*. Hal ini menunjukkan bahwa sistem memblokir akses akun siswa untuk melakukan tindakan tersebut.

6. Input Data Siswa Dengan Akun Siswa



```
2 {
3   "noinduk": "0110220268",
4   "name": "Ahoy",
5   "address": "Water Nation",
6   "birthdate": "2000-07-14",
7   "email": "Ahoy@gmail.com",
8   "phone": "08176368482",
9   "group": "A",
10  "parentname": "Ujang"
11 }
```

Body Cookies (2) Headers (3) Test Results Status: 403 Forbidden Time: 3 ms Size: 186 B

```
1 {
2   "message": "You are not authorized to access this route",
3   "status": 403
4 }
```

Gambar 4. 49 Input Data Siswa Dengan Akun Siswa

Pada Gambar 4.46, implementasi penambahan data siswa dengan akun siswa menghasilkan respon *"You are not authorized to access this route"*. Hal ini menunjukkan bahwa sistem memblokir akses akun siswa untuk melakukan tindakan tersebut.

4.4 Pengujian Sistem

Pengujian sistem merupakan serangkaian langkah yang terstruktur untuk menguji dan menilai sistem yang telah dirancang. Tujuannya adalah untuk

memastikan bahwa aplikasi berfungsi dengan benar sesuai dengan persyaratan yang telah ditentukan. Penelitian ini menggunakan metode *black box testing* untuk menguji sistem. Pengujian dilakukan dengan cara mencoba kinerja sistem berdasarkan spesifikasi yang dirancang dan membandingkannya dengan hasil yang dicapai. Fokus pengujian adalah pada fungsionalitas fitur sistem. Skenario pengujian dan uji diperoleh dari perancangan pengujian *black box* yang telah disusun sebelumnya.

Tabel 4. 10 Black Box Testing - Kelola Data Guru

Skenario Uji		Kelola Data Guru		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
1	<i>Insert data guru dengan benar. (semua field terisi)</i>	<i>POST, http://localhost:3000/api/teachers</i>	Akan menghasilkan respon <i>Code: 200, Message: "Berhasil menambahkan data guru."</i> , dan <i>Data: (data yang di-input).</i>	Sesuai.
2	<i>Insert data guru dengan salah. (terdapat field yang tidak terisi)</i>	<i>POST, http://localhost:3000/api/teachers</i>	Akan menghasilkan respon <i>Code: 400, Message: "BAD REQUEST."</i> , dan <i>Data: "Terdapat</i>	Sesuai.

Skenario Uji		Kelola Data Guru		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
			data yang belum diisi”.	
3	Mengubah data guru.	<i>PUT,</i> <i>http://localhost:3000/api/teachers/:teacherId</i>	Akan menghasilkan respon <i>Code: 200, Message: “Data berhasil diperbarui.”</i> , dan <i>Data:</i> (data guru yang telah diubah).	Sesuai.
4	Menampilkan seluruh data guru.	<i>GET,</i> <i>http://localhost:3000/api/teachers</i>	Akan menghasilkan respon <i>Code: 200, Message: “OK.”</i> , dan <i>Data:</i> (seluruh data siswa).	Sesuai.

Berdasarkan tabel 4.10 *Black Box Testing - Kelola Data Guru*, keempat skenario pengujian yang dijalankan menunjukkan hasil yang sesuai dengan rancangan yang telah dibuat sebelumnya. Hal ini berarti bahwa fungsionalitas pengelolaan data guru pada sistem telah teruji dan berjalan dengan baik sesuai dengan yang diharapkan.

Tabel 4. 11 Black Box Testing - Kelola Data Siswa

Skenario Uji		Kelola Data Siswa		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
1	<i>Insert data siswa dengan benar. (semua field terisi)</i>	<i>POST, http://localhost:3000/api/students</i>	Akan menghasilkan respon <i>Code: 200, Message: "Berhasil menambahkan data siswa."</i> , dan <i>Data: (data yang di-input).</i>	Sesuai.
2	<i>Insert data siswa dengan salah. (terdapat field yang tidak terisi)</i>	<i>POST, http://localhost:3000/api/students</i>	Akan menghasilkan respon <i>Code: 400, Message: "BAD REQUEST."</i> , dan <i>Data: "Terdapat data yang belum diisi"</i> .	Sesuai.
3	<i>Insert data siswa lebih dari satu dengan menggunakan file CSV yang</i>	<i>POST, http://localhost:3000/api/students /all</i>	Akan menghasilkan respon <i>Code: 200, Message: "Berhasil menambahkan</i>	Sesuai.

Skenario Uji		Kelola Data Siswa		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
	benar. (semua <i>field</i> terisi)		data siswa.”, dan <i>Data:</i> (seluruh data yang di- <i>input</i>).	
4	Mengubah data siswa.	<i>PUT</i> , <i>http://localhost:3000/api/students</i> <i>/:studentId</i>	Akan menghasilkan respon <i>Code:</i> 200, <i>Message:</i> “Data berhasil diperbarui.”, dan <i>Data:</i> (data siswa yang telah diubah).	Sesuai.
5	Menampilkan seluruh data siswa berdasarkan kelompok kelas.	<i>GET</i> , <i>http://localhost:3000/api/students</i> <i>/group/:groupName</i>	Akan menghasilkan respon <i>Code:</i> 200, <i>Message:</i> “OK.”, dan <i>Data:</i> (seluruh data siswa berdasarkan kelompok kelas).	Sesuai.

Berdasarkan tabel 4.11 *Black Box Testing* - Kelola Data Siswa, kelima skenario pengujian yang dijalankan menunjukkan hasil yang sesuai dengan rancangan yang telah dibuat sebelumnya. Hal ini berarti bahwa fungsionalitas pengelolaan data siswa pada sistem telah teruji dan berjalan dengan baik sesuai dengan yang diharapkan.

Tabel 4. 12 *Black Box Testing* - Kelola Data Kehadiran

Skenario Uji		Kelola Data Kehadiran		
No.	Pengujian	<i>HTTP Method/End Point</i>	Hasil Yang Diharapkan	Kesimpulan
1	<i>Input data kehadiran dengan benar. (semua field terisi)</i>	<i>POST, http://localhost:3000/api/presences</i>	Akan menghasilkan respon <i>Code: 200, Message: "Berhasil menambahkan data kehadiran", dan Data: (data yang di-input)</i>	Sesuai atau tidak sesuai.
2	<i>Input data kehadiran dengan salah. (terdapat field yang tidak terisi)</i>	<i>POST, http://localhost:3000/api/presences</i>	Akan menghasilkan respon <i>Code: 401, Message: "BAD REQUEST", dan Data: "Terdapat</i>	Sesuai atau tidak sesuai.

Skenario Uji		Kelola Data Kehadiran		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
			data yang belum diisi”.	
3	Mengubah data kehadiran.	<i>PUT</i> , <i>http://localhost:3000/api/presences/:presenceId</i>	Akan menghasilkan respon <i>Code: 200</i> , <i>Message: “Data berhasil diperbarui.”</i> , dan <i>Data: (data kehadiran yang telah diubah).</i>	Sesuai atau tidak sesuai.
4	Menampilkan seluruh data kehadiran berdasarkan id dan periode yang tersedia di <i>database</i> .	<i>GET</i> , <i>http://localhost:3000/api/presence/detail/:studentId/:periode</i>	Akan menghasilkan respon <i>Code: 200</i> , <i>Message: “OK”</i> , dan <i>Data: (data kehadiran berdasarkan id dan periode).</i>	Sesuai atau tidak sesuai.

Berdasarkan tabel 4.12 *Black Box Testing* - Kelola Data Kehadiran, keempat skenario pengujian yang dijalankan menunjukkan hasil yang sesuai dengan rancangan yang telah dibuat sebelumnya. Hal ini berarti bahwa fungsionalitas

pengelolaan data kehadiran pada sistem telah teruji dan berjalan dengan baik sesuai dengan yang diharapkan.

Tabel 4. 13 Black Box Testing - Kelola Data Penilaian

Skenario Uji		Kelola Data Penilaian		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
1	Input data penilaian dengan benar. (semua <i>field</i> terisi)	POST, <i>http://localhost:3000/api/evaluati ons</i>	Akan menghasilkan respon <i>Code</i> : 200, <i>Message</i> : “Berhasil menambahkan data penilaian”, dan <i>Data</i> : (data yang di-input)	Sesuai atau tidak sesuai.
2	Input data penilaian dengan salah. (terdapat <i>field</i> yang tidak terisi)	POST, <i>http://localhost:3000/api/evaluati ons</i>	Akan menghasilkan respon <i>Code</i> : 401, <i>Message</i> : “BAD REQUEST”, dan <i>Data</i> : “Terdapat data yang belum diisi”.	Sesuai atau tidak sesuai.

Skenario Uji		Kelola Data Penilaian		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
3	Mengubah data penilaian.	<i>PUT,</i> <i>http://localhost:3000/api/evaluations/:evaluationId</i>	Akan menghasilkan respon <i>Code: 200, Message: "Data berhasil diperbarui."</i> dan <i>Data:</i> (data penilaian yang telah diubah)	Sesuai atau tidak sesuai.
4	Menampilkan seluruh data penilaian berdasarkan id dan periode yang tersedia di <i>database</i> .	<i>GET,</i> <i>http://localhost:3000/api/evaluation/detail/:studentId/:periode</i>	Akan menghasilkan respon <i>Code: 200, Message: "OK"</i> dan <i>Data:</i> (data penilaian berdasarkan id dan periode yang dipilih)	Sesuai atau tidak sesuai.

Berdasarkan tabel 4.13 *Black Box Testing - Kelola Data Penilaian*, keempat skenario pengujian yang dijalankan menunjukkan hasil yang sesuai dengan rancangan yang telah dibuat sebelumnya. Hal ini berarti bahwa fungsionalitas pengelolaan data penilaian pada sistem telah teruji dan berjalan dengan baik sesuai dengan yang diharapkan.

Tabel 4. 14 Black Box Testing - Kelola Data Tagihan SPP

Skenario Uji		Kelola Data Tagihan SPP		
No.	Pengujian	HTTP Method/End Point	Hasil Yang Diharapkan	Kesimpulan
1	Input data pembayaran tagihan SPP dengan benar. (semua <i>field</i> terisi)	POST, <i>http://localhost:3000/api/bills</i>	Akan menghasilkan respon <i>Code</i> : 200, <i>Message</i> : “Berhasil menambahkan data pembayaran tagihan SPP”, dan <i>Data</i> : (data yang di-input)	Sesuai atau tidak sesuai.
2	Input data pembayaran tagihan SPP salah benar. (terdapat <i>field</i> yang tidak terisi)	POST, <i>http://localhost:3000/api/bills</i>	Akan menghasilkan respon <i>Code</i> : 401, <i>Message</i> : “BED REQUEST”, dan <i>Data</i> : “Terdapat data yang belum diisi”.	Sesuai atau tidak sesuai.
3	Mengubah data pemabayaran tagihan SPP	PUT, <i>http://localhost:3000/api/bills/:bil</i> <i>Id</i>	Akan menghasilkan respon <i>Code</i> : 200, <i>Message</i> :	Sesuai atau tidak sesuai.

Skenario Uji		Kelola Data Tagihan SPP		
No.	Pengujian	<i>HTTP Method/End Point</i>	Hasil Yang Diharapkan	Kesimpulan
			“Data berhasil diperbarui”, dan <i>Data</i> : (data pembayaran tagihan SPP yang telah diubah)	
4	Menampilkan seluruh data tagihan berdasarkan id dan periode yang tersedia di <i>database</i> .	<i>GET</i> , <i>http://localhost:3000/api/bill/detail/:studentId/:periode</i>	Akan menghasilkan respon <i>Code</i> : 200, <i>Message</i> : “OK” dan <i>Data</i> : (data tagihan berdasarkan id dan periode yang dipilih)	Sesuai atau tidak sesuai.

Berdasarkan tabel 4.14 *Black Box Testing* - Kelola Data Tagihan SPP, keempat skenario pengujian yang dijalankan menunjukkan hasil yang sesuai dengan rancangan yang telah dibuat sebelumnya. Hal ini berarti bahwa fungsionalitas pengelolaan data tagihan SPP pada sistem telah teruji dan berjalan dengan baik sesuai dengan yang diharapkan.

Tabel 4. 15 Black Box Testing - Kelola Data Dengan Akun Unauthorized

Skenario Uji		Kelola Data Dengan Akun <i>Unauthorized</i>		
No.	Pengujian	<i>HTTP Method/End Point</i>	Hasil Yang Diharapkan	Kesimpulan
1	<i>Input data guru dengan akun guru atau siswa</i>	<i>POST, http://localhost:3000/api/teachers</i>	Akan menghasilkan respon <i>Code: 403, Message: "FORBIDDEN", dan Data: "You are not authorized to access this route"</i>	Sesuai atau tidak sesuai
2	<i>Input data nilai dengan akun siswa</i>	<i>POST, http://localhost:3000/api/evaluations</i>	Akan menghasilkan respon <i>Code: 403, Message: "FORBIDDEN", dan Data: "You are not authorized to access this route"</i>	Sesuai atau tidak sesuai
3	<i>Input data kehadiran dengan akun siswa</i>	<i>POST, http://localhost:3000/api/presences</i>	Akan menghasilkan respon <i>Code: 403, Message: "FORBIDDEN", dan Data:</i>	Sesuai atau tidak sesuai

Skenario Uji		Kelola Data Dengan Akun <i>Unauthorized</i>		
No.	Pengujian	<i>HTTP Method/End Point</i>	Hasil Yang Diharapkan	Kesimpulan
			“ <i>You are not authorized to access this route</i> ”	
4	<i>Input data</i> pembayaran SPP tagihan dengan akun siswa	<i>POST,</i> <i>http://localhost:3000/api/bills</i>	Akan menghasilkan respon <i>Code: 403, Message: “FORBIDDEN”, dan Data: “You are not authorized to access this route”</i>	Sesuai atau tidak sesuai
5	<i>Input data</i> siswa dengan akun siswa	<i>POST,</i> <i>http://localhost:3000/api/students</i>	Akan menghasilkan respon <i>Code: 403, Message: “FORBIDDEN”, dan Data: “You are not authorized to access this route”</i>	Sesuai atau tidak sesuai

Berdasarkan tabel 4.14 *Black Box Testing* - Kelola Data Dengan Akun *Unauthorized*, kelima skenario pengujian yang dijalankan menunjukkan hasil yang

sesuai dengan rancangan yang telah dibuat sebelumnya. Hal ini berarti bahwa fungsionalitas pengelolaan data pada sistem telah teruji dan berjalan dengan baik sesuai dengan yang diharapkan.

Tabel-tabel di atas menguraikan hasil pengujian *black box* yang dilakukan terhadap *REST-API* sistem informasi akademik RA. Amanah School. Pengujian ini dilaksanakan oleh pengembang dan melibatkan 26 skenario pengujian yang dirancang untuk mengevaluasi berbagai fungsi dan fitur sistem secara menyeluruh. Setiap skenario pengujian difokuskan pada evaluasi fungsi atau fitur tertentu, dengan tujuan untuk mengidentifikasi potensi kelemahan atau kesalahan dalam sistem. Pengujian *black box* ini dilakukan dengan cara memasukan *input* yang telah ditentukan pada *REST-API* dan mengamati *output* yang dihasilkan. Hasil pengujian dicatat dan dianalisis untuk menentukan apakah sistem berfungsi sesuai dengan yang diharapkan.

4.5 Evaluasi Hasil Pengujian

Dari 26 skenario uji yang telah dilakukan oleh pengembang, tingkat keberhasilannya dapat dihitung sebagai berikut:

- Persentase keberhasilan = $\frac{\text{jumlah skenario uji yang berhasil}}{\text{total skenario uji}} \times 100\%$
- Persentase keberhasilan = $\frac{26}{26} \times 100\%$
- Persentase keberhasilan = 100%

Berdasarkan hasil perhitungan, seluruh fitur yang dievaluasi dalam skenario pengujian mencapai nilai 100%. Hal ini menunjukkan bahwa *REST-API* sistem informasi akademik RA. Amanah School telah berfungsi dengan baik dan sesuai dengan spesifikasi yang dirancang. Tidak ada *bug* atau cacat yang terdeteksi pada *REST-API* selama pengujian. Dengan kata lain, semua fungsi dan fitur yang diuji dalam skenario pengujian berjalan dengan lancar dan menghasilkan *output* yang diharapkan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian tugas akhir yang berjudul "Rancang Bangun Sistem Informasi Akademik *REST-API* Menggunakan Golang: Studi Kasus RA. Amanah School" menghasilkan beberapa kesimpulan, yaitu:

1. Rancangan *REST-API* untuk sistem informasi akademik RA Amanah School dalam penelitian ini melibatkan tiga jenis pengguna: Admin, Guru, dan Wali Murid, dengan masing-masing memiliki hak akses yang berbeda. Admin memiliki akses penuh ke seluruh sistem informasi akademik, memungkinkan mereka untuk mengelola berbagai aspek seperti data guru, siswa, nilai, presensi, dan tagihan SPP, dengan tujuan memastikan pengelolaan akademik yang baik. Saat Admin memasukkan data guru atau siswa, sistem secara otomatis akan membuat akun untuk mereka. Guru memiliki hak akses yang hampir sama dengan Admin, tetapi tidak dapat mengelola data guru atau siswa. Sementara itu, Wali Murid hanya diberi akses untuk melihat informasi terkait nilai, presensi, dan tagihan SPP.
2. Berdasarkan hasil dari semua pengujian yang telah dilakukan, sistem berjalan tanpa ada masalah. Hal ini mengindikasikan bahwa rancangan *REST-API* untuk sistem informasi akademik di RA. Amanah School telah berjalan dengan baik.

5.2 Saran

Dengan mempertimbangkan hasil penelitian tentang rancang bangun *REST-API* untuk sistem informasi akademik RA. Amanah School, ada beberapa saran yang dapat digunakan sebagai pedoman untuk pengembangan penelitian yang lebih baik di masa depan. Masukan dari penulis untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Menambahkan fitur pemulihan kata sandi, agar ketika pengguna lupa kata sandi, proses pemulihan dapat dilakukan dengan mudah melalui email atau metode lain yang aman dan efisien.

2. Sistem yang telah dikembangkan kemudian diimplementasikan ke *website*, sehingga pengguna dapat mengakses informasi akademik dengan lebih mudah dan cepat melalui antarmuka yang ramah pengguna dan responsif.
3. Untuk sistem pembayaran, disarankan menggunakan *payment gateway* agar data pembayaran dapat ditambahkan secara otomatis, tepat waktu, dan efisien.



STT - NF

DAFTAR PUSTAKA

- [1] Administrator, "Peranan Teknologi Informasi Dalam Dunia Pendidikan." Diakses: 5 Februari 2024. [Daring]. Tersedia pada: <https://diskominfo.kedirikab.go.id/baca/peranan-teknologi-informasi-dalam-dunia-pendidikan>
- [2] P. M. Ariansyah dan K. Wijaya, "Rancang Bangun Sistem Informasi Akademik Berbasis Web: Studi Kasus: SD Negeri 18 Tanah Abang," *J. Pengemb. Sist. Inf. Dan Inform.*, vol. 2, no. 3, hlm. 138–156, Nov 2021, doi: 10.47747/jpsii.v2i3.562.
- [3] Felicia Evan, "Apa itu Web Service." Diakses: 5 Februari 2024. [Daring]. Tersedia pada: <https://sis.binus.ac.id/2023/01/05/apa-itu-web-service/>
- [4] A. Pamuji, "RANCANG BANGUN WEB SERVICE MENGGUNAKAN REPRESENTATIONAL STATE TRANSFER UNTUK PENGOLAHAN DATA BARANG," 2020, [Daring]. Tersedia pada: <http://eprints.uty.ac.id/6287/1/Naskah%20Publikasi%205130411508%20Agus%20Pamuji.pdf>
- [5] Hasanuddin, H. Asgar, dan B. Hartono, "RANCANG BANGUN REST API APLIKASI WESHARE SEBAGAI UPAYA MEMPERMUDAH PELAYANAN DONASI KEMANUSIAAN," *J. Inform. Teknol. Dan Sains*, vol. 4, no. 1, hlm. 8–14, Feb 2022, doi: 10.51401/jinteks.v4i1.1474.
- [6] S. Alam, M. Yunus, dan Irmah, "INFORMASI JASA LAUNDRY BERBASIS WEB," *J. Sintaks Log.*, vol. 1, no. 1, hlm. 18–25, Jan 2021, doi: 10.31850/jsilog.v1i1.682.
- [7] The Postman Team, "What are HTTP methods." Diakses: 5 Februari 2024. [Daring]. Tersedia pada: https://blog-postman-com.translate.goog/what-are-http-methods/?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc
- [8] Vika Vitaloka Pramansah, "Sistem Informasi Peningkat Jadwal Imunisasi Pada Anak Usia Dini Menggunakan Metode Scrum Berbasis Android Di Bidan Hana Suroyyah, Am.Keb," *J. Inform. Dan Teknol. Komput. J-ICOM*, vol. 3, no. 1, hlm. 56–66, Apr 2022, doi: 10.33059/j-icom.v3i1.4948.

- [9] A. S. Sari dan R. Hidayat, "Designing website vaccine booking system using golang programming language and framework react JS," 2022.
- [10] Ariata C, "Apa itu MySQL? Pengertian MySQL, Cara Kerja, dan kelebihanannya." Diakses: 5 Februari 2024. [Daring]. Tersedia pada: <https://www.hostinger.co.id/tutorial/apa-itu-mysql>
- [11] F.- Sonata, "Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-Customer," *J. Komunika J. Komun. Media Dan Inform.*, vol. 8, no. 1, hlm. 22, Jun 2019, doi: 10.31504/komunika.v8i1.1832.
- [12] S. Pohan, "PEMODELAN UML UNTUK MENENTUKAN KELULUSAN PENERIMAAN SISWA BARU BERBASIS WEB," *J. Inform.*, vol. 3, no. 2, hlm. 41–51, Okt 2019, doi: 10.36987/informatika.v3i2.214.
- [13] M. R. Hanif, K. Panji, dan Nasrul, "ANALISIS DAN PERANCANGAN SISTEM INFORMASI PEMBAYARAN SEKOLAH BERBASIS EXTREME PROGRAMMING MENGGUNAKAN FRAMEWORK MVC," *J. Inform. Terpadu*, vol. 9, hlm. 8, Mar 2023, doi: <https://doi.org/10.54914/jit.v9i1.639>.
- [14] N. Nugroho, R. Napianto, dan G. Adithama, "Pengembangan Sistem E-Procurement Pada SMK Yadika Baturaja dengan Pendekatan Extreme Programming," vol. 3, no. 1, Diakses: 20 April 2024. [Daring]. Tersedia pada: <https://docplayer.info/209617729-Pengembangan-sistem-e-procurement-pada-smk-yadika-baturaja-dengan-pendekatan-extreme-programming.html>
- [15] Coding Studio Team, "Jenis Testing untuk Kualitas Software Kalian!" Diakses: 4 Maret 2024. [Daring]. Tersedia pada: https://codingstudio.id/blog/jenis-testing/#Apa_itu_Testing
- [16] J. Shadiq, A. Safei, dan R. W. R. Loly, "Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing," *Inf. Manag. Educ. Prof. J. Inf. Manag.*, vol. 5, no. 2, hlm. 97, Jul 2021, doi: 10.51211/imbi.v5i2.1561.

LAMPIRAN

1. *Link Source Code*

Berikut merupakan *link* dari *code* sistem informasi akademik RA. Amanah School yang telah dibangun oleh penulis:

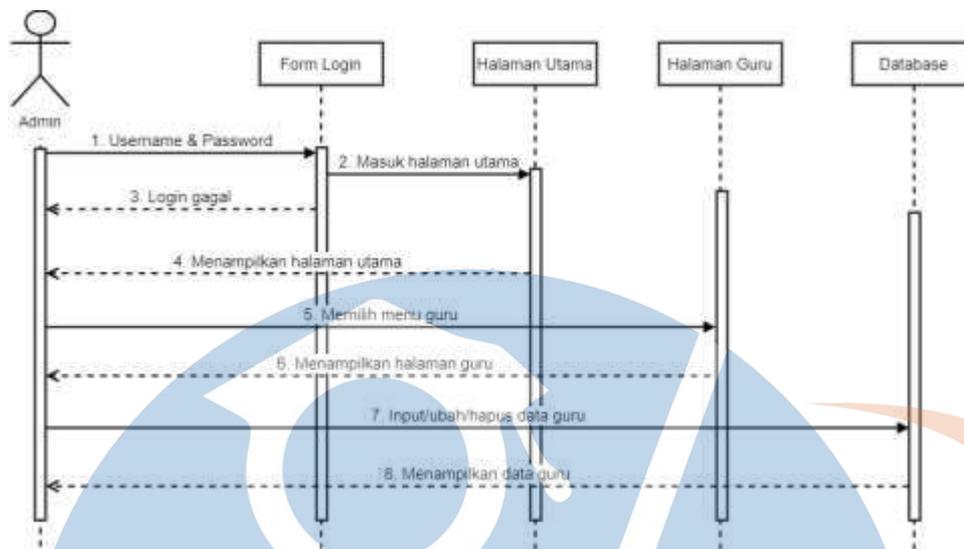
<https://github.com/DadenDharmawan/final-project>

2. *Sequence Diagram*

Sequence diagram adalah diagram interaksi yang menggambarkan urutan pesan antar objek dalam sistem selama periode waktu tertentu. Diagram ini sering digunakan untuk memvisualisasikan skenario interaksi yang kompleks dalam sistem perangkat lunak. Dalam konteks sistem informasi akademik RA. Amanah School, *sequence diagram* dapat digunakan untuk menggambarkan interaksi antara pengguna, seperti admin, guru, dan wali murid dengan sistem saat mereka mengakses data presensi, nilai, tagihan SPP, siswa, dan guru. Diagram ini dapat menunjukkan langkah-langkah yang terlibat dalam setiap interaksi, termasuk pesan yang dikirim dan diterima oleh setiap objek, serta kondisi yang memicu transisi antar langkah.

a) *Sequence Diagram* Mengelola Data Guru

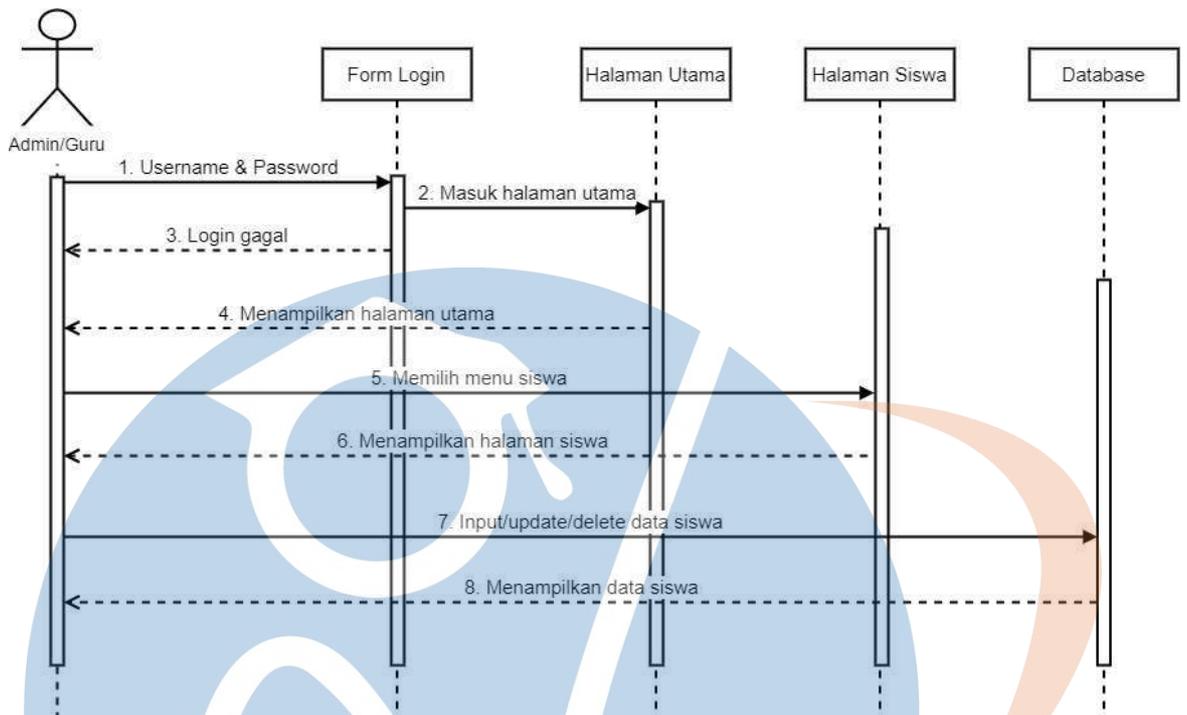
Sequence diagram ini menggambarkan interaksi antara admin dan sistem informasi akademik saat *login* dan mengelola data guru. Diagram ini menunjukkan langkah-langkah yang terlibat dalam setiap interaksi, termasuk pesan yang dikirim dan diterima oleh sistem, serta kondisi yang memicu transisi antar langkah.



Admin memulai proses dengan memasukkan *username* dan *password* di *form login*. Sistem kemudian akan memverifikasi data yang dimasukkan. Bila data valid, sistem akan mengarahkan admin ke halaman utama. Di halaman utama, admin memilih menu guru. Sistem kemudian mengarahkan admin ke halaman guru. Di halaman guru, admin dapat memilih salah satu dari tiga aktivitas: menambah data guru, mengubah data guru, atau menghapus data guru. Setelah menyelesaikan aktivitas yang dipilih, sistem akan menampilkan data guru.

b) *Sequence Diagram* Mengelola Data Siswa

Sequence diagram ini menggambarkan interaksi antara admin dan guru dengan sistem informasi akademik saat *login* dan mengelola data siswa. Diagram ini menunjukkan langkah-langkah yang terlibat dalam setiap interaksi, termasuk pesan yang dikirim dan diterima oleh sistem, serta kondisi yang memicu transisi antar langkah.

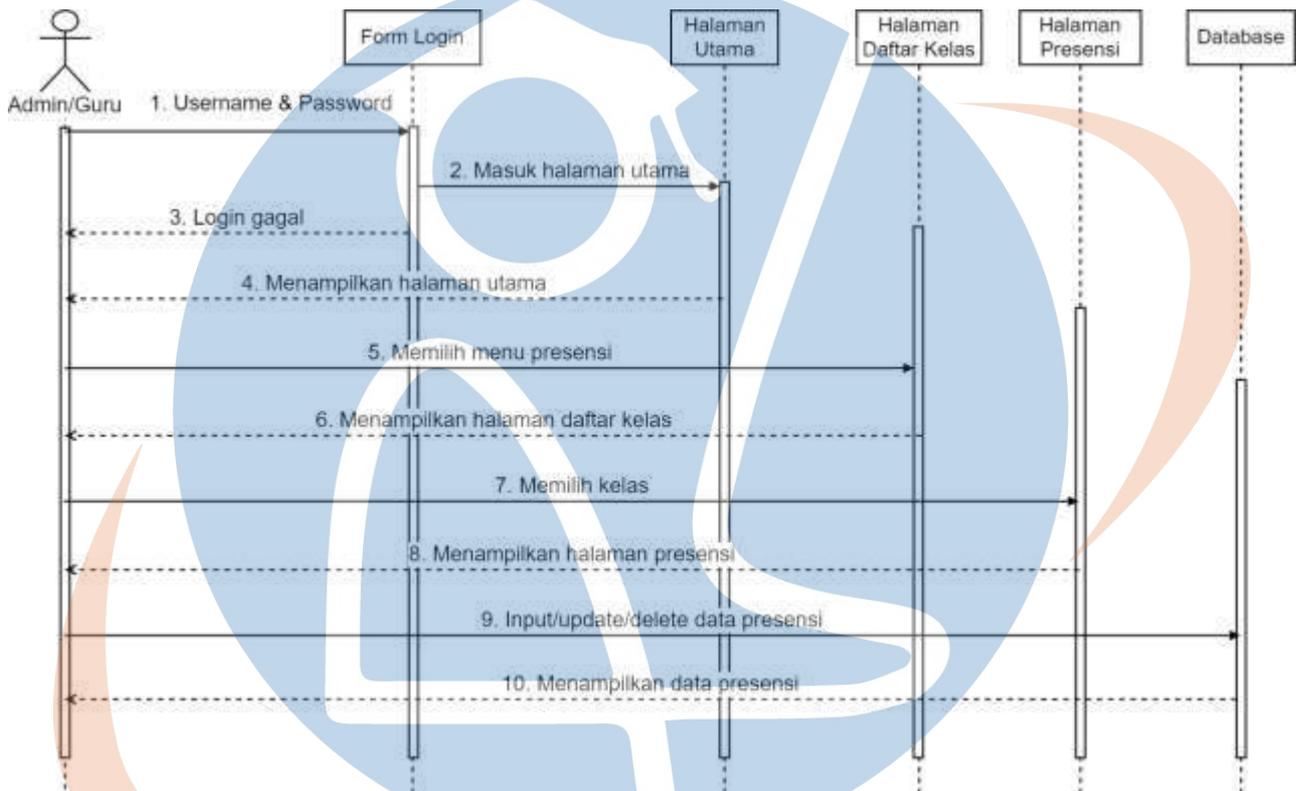


Admin dan guru mengakses sistem dengan memasukkan *username* dan *password*. Sistem memverifikasi data *login* dan mengarahkan mereka ke halaman utama. Di sana, mereka memilih menu siswa dan diarahkan ke halaman siswa. Di halaman ini, admin dan guru dapat memilih antara menambah, mengubah, atau menghapus data siswa. Setelah menyelesaikan aktivitas, sistem akan menampilkan data siswa yang telah diperbarui.

c) *Sequence Diagram* Mengelola Data Presensi

STT - NF

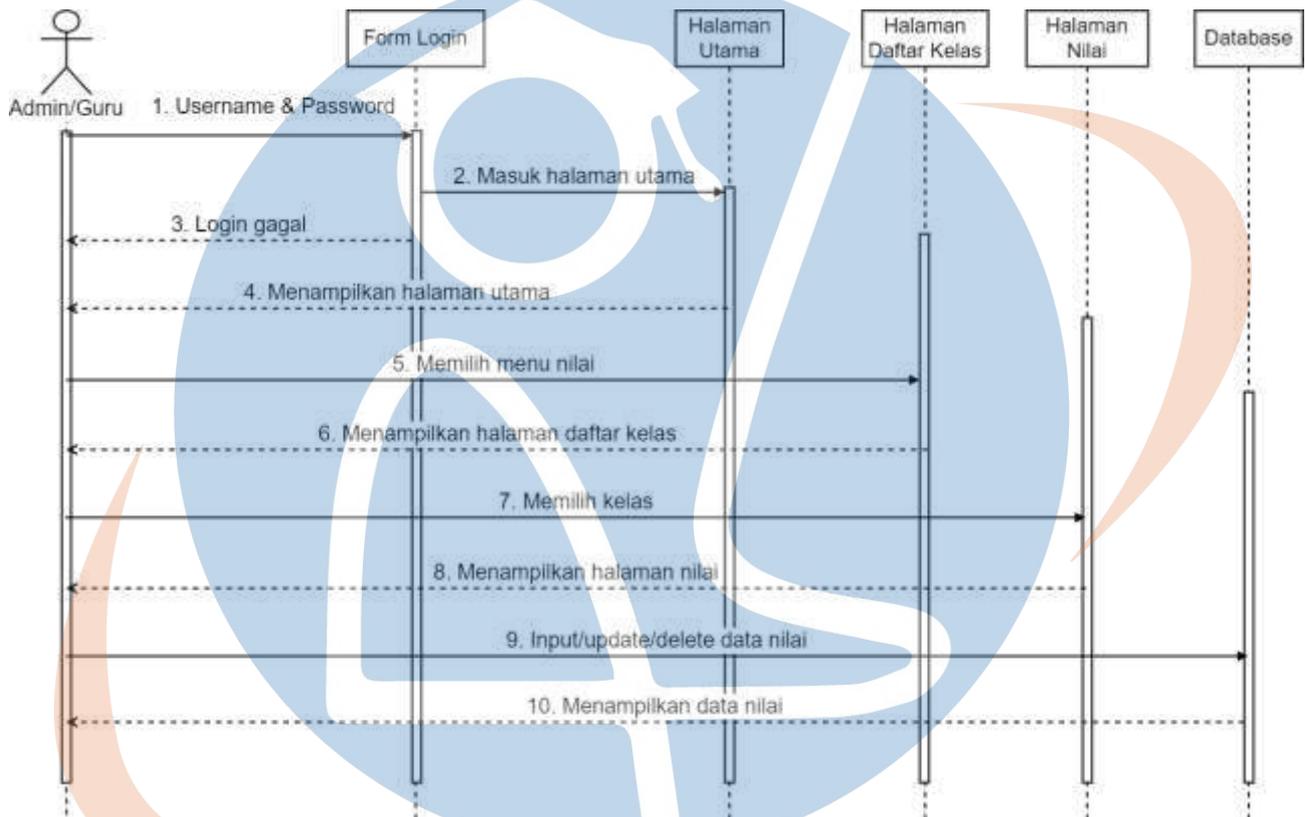
Sequence diagram ini menggambarkan urutan interaksi antara admin dan guru dengan sistem informasi akademik, khususnya dalam proses *login* dan mengelola data presensi. Diagram ini menunjukkan langkah demi langkah yang terlibat dalam setiap interaksi, termasuk pesan yang dipertukarkan antara pengguna dan sistem, serta kondisi yang memicu transisi antar langkah.



Admin dan guru mengakses sistem dengan memasukkan *username* dan *password*. Sistem kemudian memverifikasi data *login* dan membawa mereka ke halaman utama. Di halaman utama, mereka memilih menu presensi dan diarahkan ke daftar kelas. Setelah memilih kelas, mereka diarahkan ke halaman presensi di mana mereka dapat menambah, mengubah, atau menghapus data presensi. Sistem kemudian menampilkan data presensi yang telah diperbarui setelah aktivitas selesai.

d) *Sequence Diagram* Mengelola Data Nilai

Sequence diagram ini menggambarkan urutan interaksi antara admin dan guru dengan sistem informasi akademik, khususnya dalam proses *login* dan mengelola data nilai. Diagram ini menunjukkan langkah demi langkah yang terlibat dalam setiap interaksi, termasuk pesan yang dipertukarkan antara pengguna dan sistem, serta kondisi yang memicu transisi antar langkah.

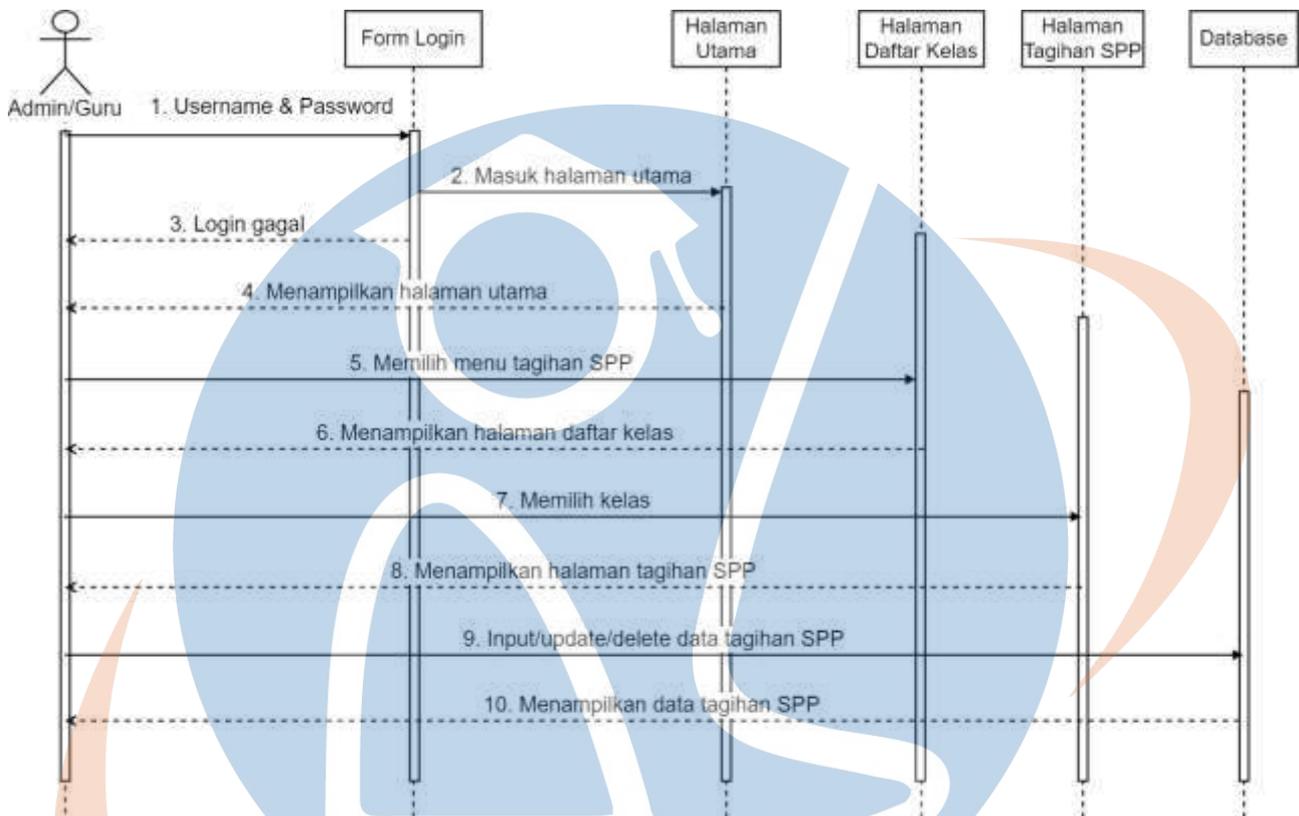


Admin dan guru mengakses sistem dengan memasukkan *username* dan *password*. Sistem kemudian memverifikasi data *login* dan membawa mereka ke halaman utama. Di halaman utama, mereka memilih menu nilai dan diarahkan ke daftar kelas. Setelah memilih kelas, mereka diarahkan ke halaman nilai di mana mereka dapat menambah, mengubah, atau menghapus data nilai. Sistem kemudian menampilkan data nilai yang telah diperbarui setelah aktivitas selesai.

e) *Sequence Diagram* Mengelola Data Tagihan SPP

Sequence diagram ini mengilustrasikan langkah-langkah interaksi antara admin dan guru dengan sistem informasi akademik dalam mengelola data tagihan SPP. Diagram ini memaparkan setiap langkah yang terlibat dalam interaksi,

termasuk pesan yang ditukar antara pengguna dan sistem, serta kondisi yang menyebabkan perpindahan antar langkah.

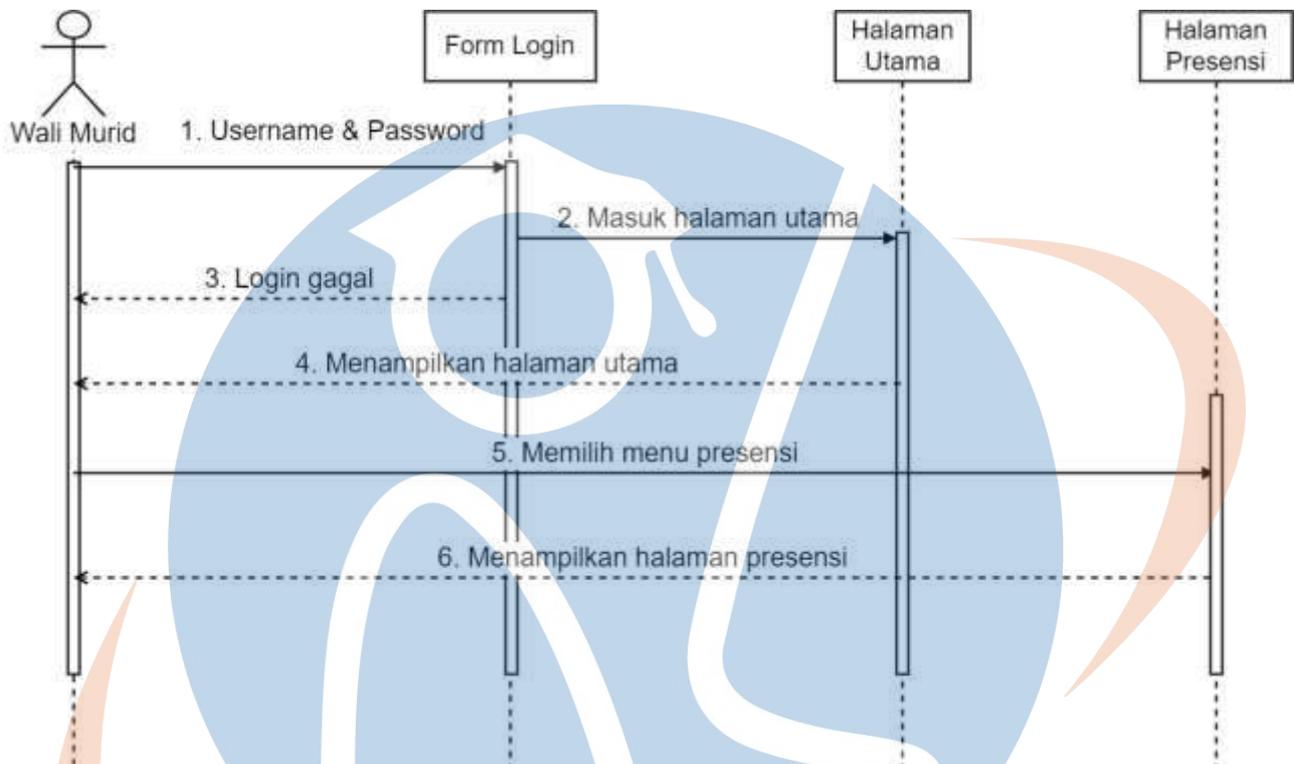


Admin dan guru mengakses sistem dengan memasukkan *username* dan *password*. Setelah data *login* diverifikasi oleh sistem, mereka diarahkan ke halaman utama. Di halaman utama, mereka memilih menu tagihan SPP dan selanjutnya diarahkan ke daftar kelas. Setelah memilih kelas yang diinginkan, mereka akan dibawa ke halaman tagihan SPP di mana mereka dapat menambah, mengubah, atau menghapus data tagihan SPP. Setelah aktivitas ini selesai, sistem akan menampilkan data tagihan SPP yang telah diperbarui.

f) *Sequence Diagram* Wali Murid Melihat Data Presensi

Sequence diagram ini mengilustrasikan langkah-langkah interaksi antara wali murid dengan sistem informasi akademik dalam memantau data presensi siswa. Diagram ini memaparkan setiap langkah yang terlibat dalam interaksi, termasuk

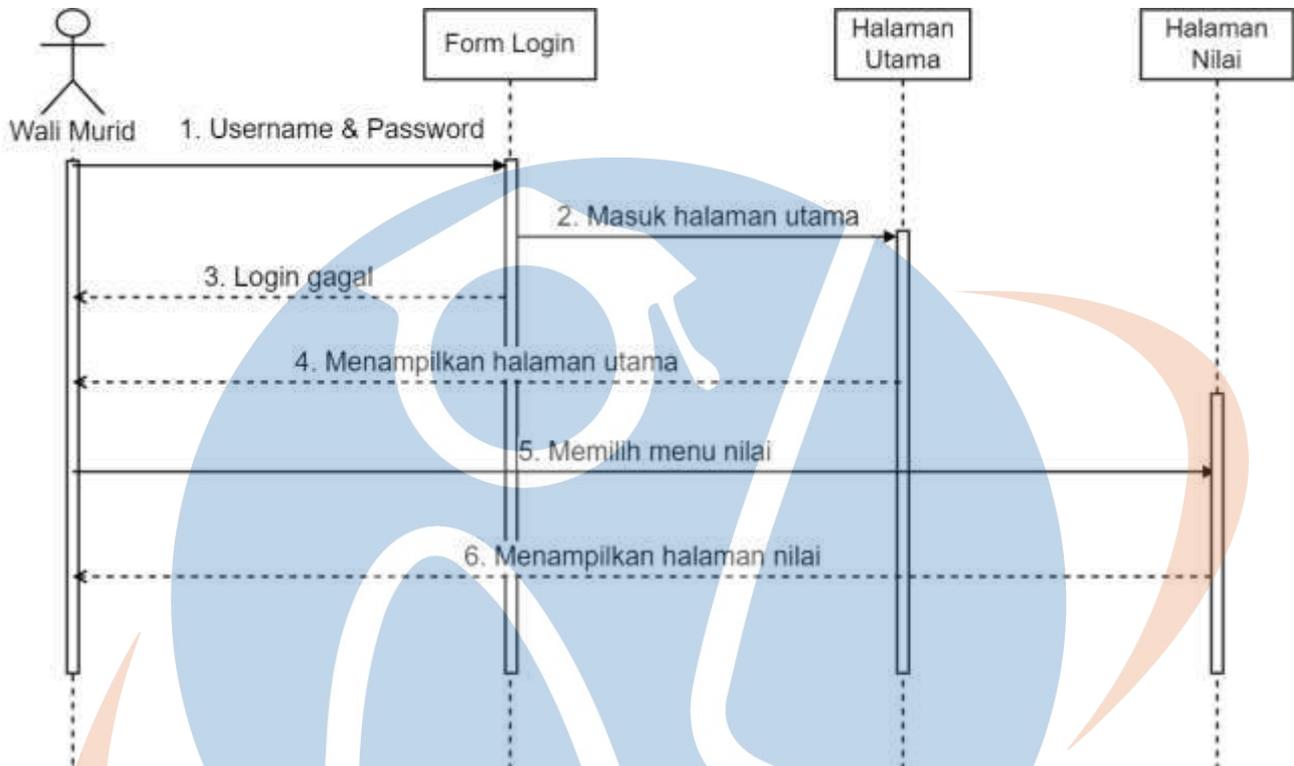
pesan yang ditukar antara pengguna dan sistem, serta kondisi yang menyebabkan perpindahan antar langkah.



Untuk dapat melihat data presensi, wali murid harus melakukan *login* terlebih dahulu dengan memasukkan *username* dan *password* pada *form login*. Setelah data *login* diverifikasi oleh sistem, wali murid akan diarahkan ke halaman utama. Di halaman utama, wali murid memilih menu presensi dan selanjutnya diarahkan ke halaman presensi di mana wali murid dapat memantau data kehadiran siswa.

g) *Sequence Diagram* Wali Murid Melihat Data Nilai

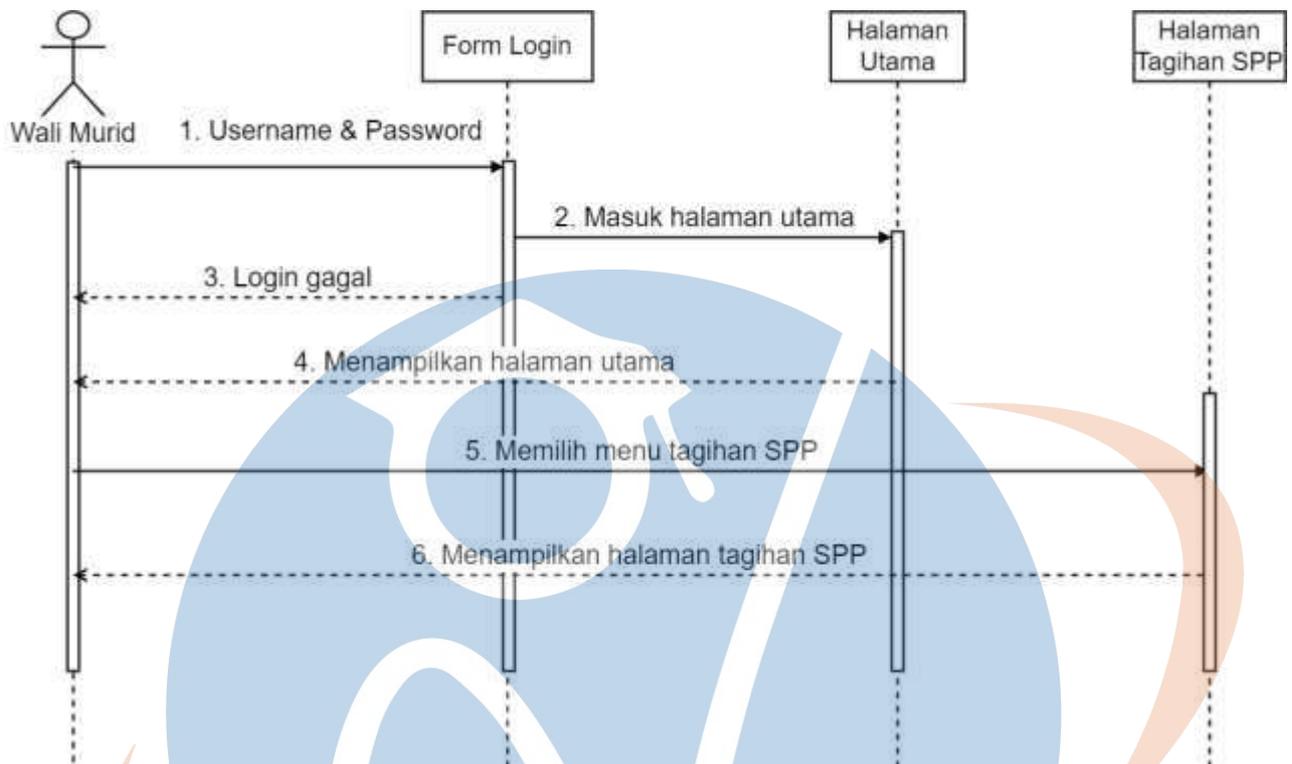
Sequence diagram ini menggambarkan urutan interaksi antara wali murid dan sistem informasi akademik untuk melihat data nilai. Diagram ini menunjukkan setiap langkah yang diambil oleh wali murid dan sistem.



Untuk dapat melihat data nilai, wali murid harus melakukan *login* terlebih dahulu dengan memasukkan *username* dan *password* pada *form login*. Setelah data *login* diverifikasi oleh sistem, wali murid akan diarahkan ke halaman utama. Di halaman utama, wali murid memilih menu nilai dan selanjutnya diarahkan ke halaman nilai di mana wali murid dapat memantau data nilai siswa.

h) *Sequence Diagram* Wali Murid Melihat Data Tagihan SPP

Sequence diagram ini menggambarkan urutan interaksi antara wali murid dan sistem informasi akademik untuk melihat data tagihan SPP. Diagram ini menunjukkan setiap langkah yang diambil oleh wali murid dan sistem.



Untuk dapat melihat data tagihan SPP, wali murid harus melakukan *login* terlebih dahulu dengan memasukkan *username* dan *password* pada *form login*. Setelah data *login* diverifikasi oleh sistem, wali murid akan diarahkan ke halaman utama. Di halaman utama, wali murid memilih menu tagihan SPP dan selanjutnya diarahkan ke halaman tagihan SPP di mana wali murid dapat memantau data tagihan siswa.

STT - NF