**PROPOSAL PROYEK AKHIR**

**RANCANG BANGUN WEBSITE SISTEM INFORMASI AKADEMIK SISWA PADA SMK NEGERI 1 MEMPURA MENGGUNAKAN METODE *PROTOTYPE***



**Disusun oleh:**

**Wiwin Sigalingging**

**NIM 2155301162**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK CALTEX RIAU**

**2025**

# RINGKASAN

SMK Negeri 1 Mempura merupakan salah satu institusi pendidikan yang terletak di Kecamatan Mempura, Kabupaten Siak, Provinsi Riau. Pengelolaan data akademik di SMK Negeri 1 Mempura masih menggunakan perangkat lunak seperti Microsoft Word dan Microsoft Excel yang menyebabkan beberapa kendala, seperti ketidakefisienan dalam pemrosesan data, kerentanan terhadap kesalahan input atau duplikasi data, dan kesulitan dalam pencarian data. Untuk mengatasi permasalahan tersebut, dirancang sebuah sistem informasi akademik siswa berbasis website yang terintegrasi untuk mendukung pengelolaan data akademik di SMK Negeri 1 Mempura seperti data guru, data siswa, nilai, absensi, kelas dan kalender akademik. Sistem ini dikembangkan menggunakan metode *prototype*, yang memungkinkan pengguna memberikan umpan balik pada setiap tahap pengembangan dan meminimalkan risiko kesalahan desain. Metode ini memastikan sistem yang dihasilkan sesuai dengan kebutuhan pengguna sebelum versi final dibangun. Dengan adanya sistem ini, diharapkan proses administrasi dan kegiatan belajar mengajar dapat berjalan lebih efektif, efisien, dan terorganisir.

**Kata kunci**: Sistem Informasi, SMK Negeri 1 Mempura, *Prototype*, Microsoft Word, Microsoft Excel

# DAFTAR ISI

[RINGKASAN i](#_Toc189922506)

[DAFTAR ISI ii](#_Toc189922507)

[DAFTAR GAMBAR iv](#_Toc189922508)

[DAFTAR TABEL v](#_Toc189922509)

[I. PENDAHULUAN 1](#_Toc189922510)

[1.1 Latar Belakang 1](#_Toc189922511)

[1.2 Perumusan Masalah 2](#_Toc189922512)

[1.3 Batasan Masalah 3](#_Toc189922513)

[1.4 Tujuan dan Manfaat 3](#_Toc189922514)

[1.4.1 Tujuan 3](#_Toc189922515)

[1.4.2 Manfaat 3](#_Toc189922516)

[1.5 Metodologi Penelitian 3](#_Toc189922517)

[1.6 Sistematika Penulisan 4](#_Toc189922518)

[II. TINJAUAN PUSTAKA 6](#_Toc189922519)

[2.1. Penelitian Terdahulu 6](#_Toc189922520)

[2.2. Landasan Teori 8](#_Toc189922521)

[2.2.1 SMK Negeri 1 MEMPURA 8](#_Toc189922522)

[2.2.2 Sistem Informasi 8](#_Toc189922523)

[2.2.3 *Metode Prototype* 9](#_Toc189922524)

[2.2.4 *Framework* 12](#_Toc189922525)

[2.2.5 Laravel 13](#_Toc189922526)

[2.2.6 MySQL 14](#_Toc189922527)

[2.2.7 UML 15](#_Toc189922528)

[2.2.8 *Black-box Testing* 16](#_Toc189922529)

[III. PERANCANGAN 17](#_Toc189922530)

[3.1. Desain Penelitian 17](#_Toc189922531)

[3.2. Requirements Analysis 18](#_Toc189922532)

[3.2.1 Hasil Wawancara 18](#_Toc189922533)

[3.2.2 Kebutuhan Fungsional 19](#_Toc189922534)

[3.2.3 Kebutuhan Non Fungsional 19](#_Toc189922535)

[3.3. Implementasi *Prototype* 20](#_Toc189922536)

[3.3.1 *Design* 20](#_Toc189922537)

[3.3.2 *Prototyping* 31](#_Toc189922538)

[3.3.3 *Customer Evaluation* 36](#_Toc189922539)

[3.3.4 *Review* & *Updation* 36](#_Toc189922540)

[3.4. Development & Testing 36](#_Toc189922541)

[3.4.1 *Development* 36](#_Toc189922542)

[3.4.2 *Testing* 36](#_Toc189922543)

[3.5. *Maintenance* 41](#_Toc189922544)

[IV. JADWAL DAN ANGGARAN BIAYA 42](#_Toc189922545)

[4.1. Jadwal 42](#_Toc189922546)

[4.2. Perkiraan Biaya 42](#_Toc189922547)

[DAFTAR PUSTAKA 43](#_Toc189922548)

[LAMPIRAN A DOKUMENTASI WAWANCARA A-1](#_Toc189922549)

[LAMPIRAN B PENGUMPULAN KEBUTUHAN *USER* B-1](#_Toc189922550)

[LAMPIRAN C *WIREFRAME* C-1](#_Toc189922551)

# DAFTAR GAMBAR

Gambar 2.1 Metode Prototype………………………………………………………..9

[Gambar 3.1 Desain Penelitian 17](#_Toc188845417)

[Gambar 3.2 Use Case Diagram 21](#_Toc188845418)

[Gambar 3.3 Class Diagram 30](#_Toc188845419)

[Gambar 3.4 ERD 31](#_Toc188845420)

[Gambar 3.5 Antarmuka Form Login 32](#_Toc188845421)

[Gambar 3.6 Antarmuka Data Siswa 32](#_Toc188845422)

[Gambar 3.7 Antarmuka Data Guru 33](#_Toc188845423)

[Gambar 3.8 Antarmuka Kelas 34](#_Toc188845424)

[Gambar 3.9 Antarmuka Nilai Siswa 34](#_Toc188845425)

[Gambar 3.10 Absensi Siswa 35](#_Toc188845426)

[Gambar 3.11 Kalender Akademik 36](#_Toc188845427)

# DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu………………………………………………………..7

[Tabel 3.1 Kebutuhan Fungsional 19](#_Toc188847063)

[Tabel 3.2 Kebutuhan Non Fungsional 19](#_Toc188847064)

[Tabel 3.3 Melakukan Login 21](#_Toc188847065)

[Tabel 3.4 Melihat Dashboard 22](#_Toc188847066)

[Tabel 3.5 Mengelola Data Siswa 22](#_Toc188847067)

[Tabel 3.6 Mengelola Kelas 23](#_Toc188847068)

[Tabel 3.7 Mengelola Data Guru 25](#_Toc188847069)

[Tabel 3.8 Mengelola Absensi Siswa 26](#_Toc188847070)

[Tabel 3.9 Mengelola Nilai Siswa 27](#_Toc188847071)

[Tabel 3.10 Mengelola Kalender Akademik 29](#_Toc188847072)

[Tabel 3.11 Black-box Testing 37](#_Toc188847073)

[Tabel 3.12 Bobot Jawaban Kuesioner UAT 39](#_Toc188847074)

[Tabel 3.13 Usability Testing (Admin) 39](#_Toc188847075)

[Tabel 3.14 Usability Testing (Guru) 40](#_Toc188847076)

[Tabel 3.15 Usability Testing (Siswa) 40](#_Toc188847077)

Tabel 4.1 Jadwal……………………………………………………………………...41

Tabel 4.2 Perkiraan Biaya…………………………………………………………….42

# PENDAHULUAN

## 1.1 Latar Belakang

SMK Negeri 1 Mempura merupakan salah satu institusi pendidikan yang terletak di Kecamatan Mempura, Kabupaten Siak. Sebagai Sekolah Menengah Kejuruan, pengelolaan data akademik seperti data guru, data siswa, nilai, absensi, kelas dan kalender akademik memiliki peran penting dalam mendukung kelancaran proses administrasi dan kegiatan belajar mengajar. Namun, saat ini pengelolaan data akademik tersebut masih menggunakan perangkat lunak seperti Microsoft Worddan Microsoft Excel*.* Sehingga, menyebabkan kendala operasional yang menghambat efektivitas dan efisiensi kerja. Beberapa permasahan utama yang dihadapi dalam pengelolaan data akademik, seperti ketidakefisienan dalam pemrosesan data, kerentanan terhadap kesalahan input atau duplikasi data, dan kesulitan dalam pencarian data.

Salah satu alternatif untuk menyelesaikan permasalahan tersebut adalah dengan mengembangkan sistem informasi yang mampu mengelola data akademik di SMK Negeri 1 Mempura secara terintegrasi, akurat, dan efisien. Dalam dunia pendidikan, penerapan sistem informasi akademik menjadi langkah yang strategis untuk meningkatkan efektivitas dan efisiensi dalam pengelolaan data akademik, sehingga proses administrasi dan kegiatan belajar mengajar dapat berjalan lebih efektif dan efisien. Sistem ini memungkinkan sekolah untuk menyimpan, mengelola, dan mengakses data akademik secara digital, sehingga proses pencatatan nilai, absensi, dan informasi lainnya dapat dilakukan secara lebih cepat, akurat, dan efisien. Selain itu, data yang tersimpan dalam sistem akan lebih aman dan mudah diakses kapan saja diperlukan.

Sistem informasi akademik adalah sistem yang mengolah berbagai data pendidikan atau akademik dengan menggunakan teknologi komputer baik berbasis hardware maupun software sehingga memberikan kemudahan dalam mengelola data dan informasi terkait dunia pendidikan. Solahudin (2021) menyatakan bahwa sistem informasi akademik adalah sebuah sistem yang dikembangkan untuk mengelola data akademik, seperti data siswa, penilaian, absensi siswa, dan sebagai media informasi dari guru ke siswa sehingga lebih efektif dan terkelola dengan baik.

Dalam pengembangan perangkat lunak, terdapat berbagai metode yang dapat digunakan untuk memastikan sistem yang dibangun sesuai dengan kebutuhan pengguna. Metode perangkat lunak berfungsi sebagai kerangka kerja untuk merancang, mengembangkan, dan mengelola proses pembangunan perangkat lunak secara sistematis dan efisien. Salah satu metode yang populer yang digunakan adalah metode *prototype*. Metode *prototype* adalah teknik pendekatan yang mengutamakan pembuatan model awal untuk mendapatkan umpan balik dari pengguna agar memahami dan mengindentifikasi kebutuhan dari sistem yang akan dibangun. Keunggulan utama dari metode *prototype* adalah mempercepat pengembangan sistem dan memungkinkan fleksibilitas dalam melakukan perbaikan berdasarkan umpan balik pengguna. Dengan demikian, metode *prototype* membantu mengurangi resiko kesalahan dan kegagalan desain karena memungkinkan perbaikan sebelum membangun versi produk yang lengkap dan tepat sasaran sesuai dengan kebutuhan pengguna.

Sehingga penelitian ini bertujuan untuk mengembangkan sistem informasi akademik di SMK Negeri 1 Mempura menggunakan metode *prototype*. Sistem ini dirancang untuk mempermudah, mempercepat, dan menata pengelolaan data akademik secara lebih terorganisir. Proses sebelumnya yang menggunakan Microsoft Word dan Microsofot Excel akan digantikan oleh sistem digital yang lebih andal. Dengan demikian, potensi kesalahan dapat diminimalkan, dan akses data menjadi lebih mudah serta lebih cepat.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka perumusan masalah dalam penelitian ini adalah:

1. Pengelolaan data akademik di SMK Negeri 1 Mempura masih menggunakan perangkat lunak seperti Microsoft Word dan Microsoft Excel.
2. Pengelolaan data akademik menggunakan perangkat lunak seperti Microsoft Word dan Microsoft Excel menyebabkan beberapa kendala, seperti ketidakefisienan dalam pemrosesan data, kerentanan terhadap kesalahan input atau duplikasi data, dan kesulitan dalam pencarian data.

## 1.3 Batasan Masalah

Adapun batasan masalah dalam pembuatan proyek akhir ini adalah:

1. Sistem ini dirancang dan dibangun untuk mengelola data akademik seperti data guru, data siswa, nilai, absensi, kelas dan kalender akademik di SMK Negeri 1 Mempura.
2. Pengembangan sistem menggunakan metode *prototype* yang memungkinkan pembuatan *prototype* awal untuk melakukan perbaikan berdasarkan umpan balik pengguna.

## 1.4 Tujuan dan Manfaat

### 1.4.1 Tujuan

Adapun tujuan dari penelitian ini adalah untuk merancang dan membangun sistem informasi akademik siswa pada SMK Negeri 1 Mempura menggunakan metode *prototype*.

### 1.4.2 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah:

1. Meningkatkan efektivitas dan efesiensi dalam pengelolaan data akademik di SMK Negeri 1 Mempura.
2. Mengurangi kesalahan input dan duplikasi data akademik yang dikelola.
3. Mempermudah pencarian serta pengelolaan data.
4. Mempercepat, mempermudah, dan menata pengelolaan data akademik agar lebih terorganisir dan efesien.

## 1.5 Metodologi Penelitian

Adapun metode penelitian yang digunakan dalam pembuatan proyek ini adalah:

1. Wawancara

Teknik pengumpulan data dengan mengadakan interaksi untuk mendapatkan informasi dan kebutuhan dengan pihak SMK Negeri 1 Mempura mengenai permasalahan yang ada.

1. Studi Literatur

Melakukan studi literatur dengan membaca dan mempelajari buku, jurnal, serta artikel yang relevan dengan pengembangan sistem informasi akademik.

1. Perancangan

Setelah melakukan wawancara dan studi literatur, tahapan perancangan untuk melakukan proses analisis dan desain sistem guna mendapatkan pendekatan sistem yang lebih baik.

1. Implementasi

Implementasi merupakan tahap yang dilakukan setelah perancangan, pada tahap ini, hasil dari analisis dan perancangan akan dilakukan implementasi menjadi sebuah sistem.

1. Pengujian

Setelah implementasi, dilakukan pengujian untuk memastikan bahwa sistem berjalan sesuai dengan yang diharapkan. Pengujian ini mencakup pengujian fungsional untuk setiap fitur utama sistem, serta pengujian pengguna *(user testing)*. Evaluasi dilakukan untuk mengidentifikasi dan memperbaiki ketidaksesuaian atau kesalahan yang ditemukan selama pengujian.

## 1.6 Sistematika Penulisan

Sistematika penulisan laporan proyek akhir ini secara keseluruhan terdiri dari empat bab, masing-masing terdiri dari beberapa sub bab. Adapun pokok pembahasan dari masing-masing bab tersebut secara garis besar sebagai berikut:

**BAB I PENDAHULUAN**

Bab ini menguraikan tentang latar belakang masalah, perumusan masalah dan ruang lingkup masalah, tujuan dan manfaat penelitian, metodologi penelitian dan sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Bab ini menguraikan beberapa hasil penelitsian terdahulu dan landasan teori yang diperlukan untuk merancang sistem.

**BAB III PERANCANGAN**

Bab ini menjelaskan tentang perancangan sistem terdiri dari perancangan sistem yang akan dibangun.

**BAB IV JADWAL DAN PERKIRAAN BIAYA**

Bab ini berisi informasi mengenai jadwal pengerjaan proyek akhir dan perkiraan biaya yang dibutuhkan untuk pengerjaan proyek akhir.

# TINJAUAN PUSTAKA

## Penelitian Terdahulu

Penelitian pertama dilakukan oleh Santoso & Amanullah (2022) yang merancang sistem informasi akademik berbasis *website* menggunakan metode *Research and Development (R&D)* pada SMA YATPI Godong. Sistem ini mencakup modul seperti absensi siswa, absensi guru, nilai dan jadwal pelajaran. Tujuan penelitian ini adalah untuk mempermudah pengolahan data akademik dan informasi akademik di SMA tersebut. Hasilnya akhirnya adalah sebuah sistem informasi akademik berbasis web yang mampu mendukung kebutuhan sekolah dalam mengelola data akademik dengan lebih baik dan cepat.

Penelitian kedua dilakukan oleh Harahap (2022) yang mengembangkan sistem informasi akademik berbasis web untuk SMK Negeri 4 Banda Aceh. Penelitian ini menggunakan metode *Research and Development (R&D)* yang berfokus pada modul data guru, kelas, jadwal mata pelajaran dan nilai. Penelitian ini bertujuan untuk menciptakan mode kerja pendidikan yang lebih efektif, efisien dan mudah dioperasikan. Hasil akhirnya adalah sistem informasi akademik yang mempermudah pelaksanaan operasional pendidikan di SMK Negeri 4 Banda Aceh.

Penelitian ketiga oleh Pratiwi dkk. (2020) yang mengembangkan sistem informasi akademik berbasis web di SMP Rahmat Islamiyah menggunakan metode *Research and Development (R&D)*. Sistem ini fokus pada modul nilai, jadwal dan data alumni. Tujuan penelitian ini adalah untuk membantu dan mempercepat proses pencatatan data guru, siswa, kelas, jadwal pelajaran dan data alumni. Penelitian ini menghasilkan sistem informasi akademik berbasis *website* yang mampu mendukung kebutuhan sekolah dan siswa secara efisien di SMP Rahmat Islamiyah.

Penelitian keempat dilakukan oleh Sadewa dkk. (2024) yaitu merancang sistem informasi akademik berbasis web di SMK Al-Islam Surakarta menggunakan metode *waterfall*. Sistem ini berfokus pada modul pendaftaran siswa, absensi, nilai dan penyampaian informasi. Hasilnya adalah sistem informasi akademik yang membantu sekolah dalam mengelola data akademik dengan lebih mudah dan terstruktur.

Tabel 2.1 Penelitian Terdahulu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Peneliti** | **Modul** | **Metode** | **Tujuan Penelitian** | **Hasil** |
| Santoso & Amanullah (2022) | Absensi siswa, absensi guru, nilai dan jadwal pelajaran | *Research and Development* *(R&D)* | Mempermudah pengolahan data akademik dan informasi akademik berbasis web. | Sistem informasi akademik pada SMA YATPI Godong |
| Harahap (2022) | Data guru, kelas, jadwal mata pelajaran dan nilai | *Research and Development* *(R&D)* | Mempermudah metode kinerja pendidikan yang lebih efektif, efisien dan mempermudah dalam pengoperasian pendidikan. | Sistem Informasi Akademik Sekolah berbasis web di SMK Negeri 4 Banda Aceh |
| Pratiwi dkk. (2020) | Nilai, jadwal dan data alumni | *Research and Development* *(R&D)* | Membantu dan mempercepat proses pencatatan data guru, siswa, kelas, jadwal pelajaran dan data alumni | Sistem Informasi Akademik berbasis web di SMP Rahmat Islamiyah |
| Sadewa dkk. (2024) | Pendaftaran siswa, absensi, nilai dan penyampaian infromasi | *Waterfall* | mempermudah proses pendaftaran, absensi, pelaporan nilai dan penyampaian informasi. | Sistem informasi akademik di SMK Al-Islam Surakarta |

Penelitian ini merancang dan membangun sistem informasi akademik siswa berbasis *website* untuk SMK Negeri 1 Mempura menggunakan metode *prototype*. Sistem ini dikembangkan mencakup modul data guru, data siswa, nilai, absensi, kelas dan kalender akademik yang bertujuan untuk meningkatkan efektivitas dan efisiensi dalam pengelolaan data akademik. Hasilnya adalah sistem informasi ini diharapkan mampu memberikan solusi yang terintegrasi dan mempermudah manajemen data akademik di SMK Negeri 1 Mempura. Sebagaimana dijelaskan pada Tabel 2.1, yang dimana penelitian sebelumnya menggunakan metode *Research and Development* *(R&D)* dan *Waterfall,* penelitian ini akan melanjutkan kebutuhan spesifik di SMK Negeri 1 Mempura menggunakan metode *prototype.* Alasan menggunakan metode *prototype* karena mempercepat proses pengembangan sistem dengan melakukan perbaikan berdasarkan umpan balik pengguna sehingga sesuai dengan kebutuhan pengguna.

## Landasan Teori

### SMK Negeri 1 MEMPURA

SMK Negeri 1 Mempura merupakan salah satu institusi pendidikan yang terletak di Kecamatan Mempura, Kabupaten Siak, Provinsi Riau. Sekolah ini didirikan pada tahun 2001, pada waktu itu masih menggunakan program kurikulum belajar SMK 2013 REV dan sekarang menggunakan kurikum merdeka. SMK Negeri 1 Mempura mendapat akreditasi grade A dengan nilai 98 (Akreditasi tahun 2019) dari BAN-S/M (Badan Akreditasi Nasional) Sekolah/Madrasah.

SMK Negeri 1 Mempura menyediakan pendidikan kejuruan di berbagai bidang untuk mempersiapkan siswa memasuki dunia kerja dengan keterampilan praktis berdasarkan jurusan yang mereka pilih. Jurusan yang ada pada sekolah ini, yaitu Desain Pemodelan dan Informasi Bangunan (DPIB), Akutansi & Keuangan Lembaga (AKL), Manajemen Perkantoran & Layanan Bisnis (MPLB), Teknik Geospasial, Teknik Audio Video (TAV), Teknik Komputer Jaringan & Telekomunikasi (TKJT), Pengembangan Perangkat Lunak & Gim (PPLG), Teknik Las, Teknik Mesin, Teknik Kendaraan Ringan dan Teknik Sepeda Motor.

### Sistem Informasi

Solahudin (2021) menyatakan sistem informasi adalah sebuah sistem yang mampu menyediakan informasi atau data yang sudah diproses dengan menggunakan teknologi untuk mendukung operasi dan manajemen dalam suatu organisasi sehingga memudahkan dalam pengambilan keputusan yang relevan dan akurat. Sistem informasi terdiri dari beberapa komponen utama yang saling berhubungan agar sistem berjalan dengan baik dalam mengumpulkan, memproses, menyimpan dan menyediakan data menjadi informasi, yaitu *hardware, software, database, network, procedure* dan *people.* Setiap komponen yang ada memiliki fungsi masing-masing yang saling terkait sehingga dapat membentuk sistem informasi yang efektif dan mudah dipahami.

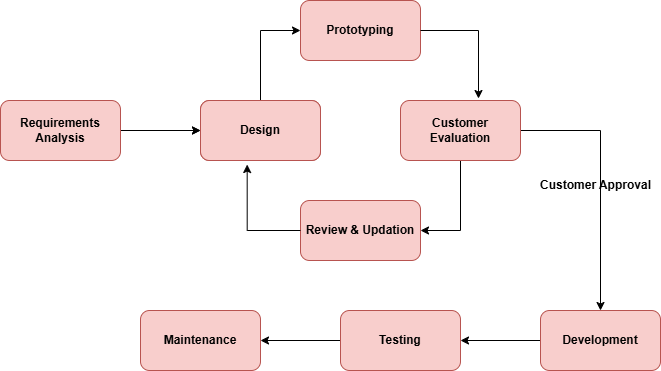
Sedangkan Harahap (2022) menyatakan bahwa sistem informasi menggunakan perangkat keras seperti komputer sebagai alat untuk mengolah data mentah menjadi informasi untuk memecahkan masalah tertentu. Dengan memanfaatkan teknologi, sistem informasi dapat bekerja dengan cepat tanpa membuat kesalahan dan memberikan keputusan yang tepat bagi penggunanya. Keberhasilan dari sistem informasi dilihat dari hubungan yang baik antara manusia, proses dan teknologi. Manusia berperan sebagai pengguna yang mengelola, merancang dan mengembangkan kebutuhan sistem serta memanfaatkan informasi dalam pengambilan keputusan. Proses merupakan langkah dalam mengolah data menjadi informasi yang akurat. Sementara itu, teknologi sebagai alat untuk mengolah data, menyimpan dan memberikan informasi secara cepat dan mudah.

### *Metode Prototype*

Putri dkk. (2024) menjelaskan bahwa metode *prototype* adalah teknik pendekatan yang mengutamakan pembuatan model awal untuk mendapatkan umpan balik dari pengguna agar memahami dan mengindentifikasi kebutuhan dari sistem yang akan dibangun. *Prototype* akan menggambarkan tampilan awal untuk kelanjutan sistem yang akan dibangun, tujuannya adalah untuk memastikan sistem yang akan dikembangkan sesuai dengan kebutuhan pengguna. Keunggulan utama dari metode *prototype* adalah mempercepat pengembangan sistem dan memungkinkan fleksibilitas dalam melakukan perbaikan berdasarkan umpan balik pengguna. Dengan demikian, metode *prototype* membantu mengurangi resiko kesalahan dan kegagalan desain karena memungkinkan perbaikan sebelum membangun sistem yang lengkap dan tepat sasaran sesuai dengan kebutuhan pengguna.

Aprianto & Wardani (2021) menyatakan metode *prototype* merupakan sebuah metode yang dapat menyelesaikan kesalahpahaman dan menjelaskan kebutuhan sistem secara spesifik kepada pengguna. Umumnya, banyak pelanggan hanya memberikan gambaran umum sistem saja tanpa menjelaskan secara detail spesifikasi kebutuhan sistem kepada pengembang perangkat lunak. Spesifikasi kebutuhan ini meliputi detail *input,* proses atau detail *output*. Disisi lain, pengembang salah menafsirkan kebutuhan atau umpan balik pelanggan sehingga menghasilkan *prototype* yang tidak sesuai.

Arizal dkk. (2022)menyatakan bahwa metode *prototype* ini terdiri dari beberapa tahapan, yaitu dimulai dari perencanaan, analisis, perancangan, implementasi hingga pemeliharaan sistem.



Gambar 2.1 Metode Prototype

Tahapan dalam metode *prototype* adalah sebagai berikut:

1. *Requirements Analysis*

Menganalisis kebutuhan secara terperinci dengan melakukan diskusi dengan pengguna untuk mengumpulkan kebutuhan secara spesifik. Analisis kebutuhan meliputi identifikasi spesifikasi fungsional dan non-fungsional sistem berdasarkan masukan dari pengguna.

1. *Design*

Setelah menganalisa kebutuhan pengguna, kemudian merancang sistem yang mencakup arsitektur sistem, rancangan database dan desain sederhana dari hasil diskusi pengguna dan pengembang untuk memberikan gambaran awal tentang sistem yang akan dibuat.

1. *Prototyping*

Tahap *prototyping* adalah pembuatan model awal dari sistem yang dirancang untuk menguji fungsionalitas dan desain sistem. Pada tahapan ini, bertujuan untuk memastikan semua kebutuhan pengguna dipahami dengan baik sebelum melakukan tahap pengembangan.

1. *Customer Evaluation*

Pada tahap *customer* *evaluation* bertujuan untuk mendapatkan umpan balik dari pengguna terhadap *prototype* yang sudah dibuat. Proses ini membantu mengevaluasi apakah *prototype* sudah sesuai dengan kebutuhan pengguna.

1. *Review & Updating*

Tahap ini berfokus pada revisi dan perbaikan *prototype* berdasarkan umpan balik dari hasil evaluasi yang sudah dilakukan sebelumnya. Proses ini akan dilakukan secara berulang-ulang hingga mendapatkan persetujuan dari pengguna.

1. *Development*

Tahapan *development* dapat dilakukan jika pengguna sudah menyetujui *prototype* yang sudah dibuat. Pada tahapan ini merupakan implementasi penuh dari sistem berdasarkan desain yang sudah disetujui yang mencakup penulisan kode program, sehingga menghasilkan sistem yang siap untuk diuji lebih lanjut dan memastikan semua kebutuhan pengguna telah terakomodasi.

1. *Testing*

Tahap *testing* (pengujian) dilakukan untuk memastikan bahwa sistem berjalan sesuai dengan yang diharapkan. Pengujian ini mencakup pengujian fungsional untuk setiap fitur utama sistem, serta pengujian pengguna *(user testing)*. Evaluasi dilakukan untuk mengidentifikasi dan memperbaiki ketidaksesuaian atau kesalahan yang ditemukan selama pengujian.

1. *Maintenance*

Tahap *maintenance* (pemeliharaan) dilakukan agar sistem berjalan dengan baik tanpa kendala setelah digunakan. Pada tahap ini mencakup memperbaiki masalah yang ditemukan dan melakukan perawatan agar sistem tetap aman, cepat dan berguna dalam jangka panjang.

### *Framework*

Ambriani & Nurhidayat (2020) menyatakan *framework* adalah sebuah kerangka kerja yang digunakan *developer* untuk mengembangkan sebuah aplikasi berbasis *desktop* ataupun *website*. Kerangka ini berperan sebagai *template* untuk memudahkan pekerjaan *developer* dalam menulis baris kode yang jauh lebih cepat, mudah dan terstruktur. Dalam mengembangkan sebuah perangkat lunak atau aplikasi, *framework* membantu *developer* dalam menghemat waktu, mempercepat proses pengembangan dan memudahkan dalam pemeliharaan aplikasi atau sistem. Dengan menggunakan *framework, developer* tidak memulai dari nol karena *framework* sudah menyediakan kode-kode yang sudah dirancang sebelumnya dan bersifat *open source*, dapat diakses secara gratis tanpa biaya.

Kansha dkk. (2023) menyatakan bahwa *framework* menyediakan *model* kerangka kerja dasar yang berisi perintah dan fungsi untuk membangun aplikasi web dengan cepat dan lebih mudah karena lebih fokus pada tujuan. *Framework* dapat digunakan dalam mengembangkan sebuah aplikasi berbasis *mobile, website* maupun *desktop*. Jenis-jenis *framework* dalam pembuatan *website* ada tiga, yaitu *Framework CSS, Framework Javascript* dan *Framework Php.* Masing-masing memiliki fungsi dan menggunakan bahasa pemrograman yang berbeda sesuai kebutuhan *developer.*

*Framework* memiliki beberapa kelebihan dalam pengembangan perangkat lunak, diantaranya *framework* menyediakan struktur kode yang terstruktur seperti *Model, View, Controller* (MVC)*,* memiliki fitur keamanan yang tinggi, mempercepat proses pengembang sistem karena mengurangi kode program yang berulang. Namun, *framework* juga memiliki kekurangan yang harus diperhatikan *developer*, seperti *framework* memiliki ukuran yang lebih besar, memiliki keterbatasan dalam menyesuaikan fungsionalitas di luar standar *framework.* Untuk itu, *developer* perlu memutuskan dan menyesuaikan dalam penggunaan *framework* tergantung kebutuhan proyek.

### Laravel

Ambriani & Nurhidayat (2020) menyatakan bahwa salah satu *framework* PHP yang paling popluer adalah laravel. Laravel pertama kali diciptakan oleh Taylor Otwell dan dirilis pada tahun 2011. Taylor Otwell mengembangkan laravel menjadi salah satu kerangka kerja PHP yang populer karena mudah digunakan, mudah dipelajari dan menyediakan fitur-fitur yang lengkap dan modern. Laravel dibangun dan dirancang dengan konsep MVC yaitu *Model, View* dan *Controller.* *Model* merupakan komponen yang mengelola *database*, *view* merupakan komponen yang mengatur tampilan antarmuka dan *controller* bertanggungjawab untuk menghubungkan *model* dan *view.* Penggunaan MVC menjadikan implementasi aplikasi lebih sederhana dan memudahkan *developer* melakukan perubahan pada satu komponen tanpa mempengaruhi komponen lainnya

Sinlae dkk. (2024) juga menyatakan *framework laravel* dalam pembangunan aplikasi website berbasis PHP terbukti menjadi pilihan yang efisien, andal dan berkelanjutan karena mempercepat proses pengembangan aplikasi *website*. Laravel menyediakan dua fitur utama seperti *Eloquent ORM* dan *Blade templating* yang membantu *developer* membangun aplikasi yang aman, terstruktur dan terukur. *Eloquent ORM (Object Relational Mapper)* merupakan fitur utama dari laravel sendiri. *Eloquent* mempermudah *developer* dalam bekerja dengan database menggunakan model yang memungkinkan *developer* menulis *query* SQL yang kompleks. Sedangkan *blade templating* adalah template bawaan laravel yang dirancang untuk membuat tampilan antarmuka pengguna UI/UX yang dinamis dan dapat digunakan kembali.

Laravel juga dilengkapi dengan fitur artisan, sebuah *Command Line Interface (*CLI*)* untuk memproses dan menjalankan perintah-perintah otomatis seperti membuat *model, controller* atau *migration*. Proses migrasi dilakukan agar data melacak setiap perubahan yang ada pada skema *database*. Dengan berbagai fitur unggulan, komunitas yang besar, aman dan dapat diskalakan, laravel menjadi pilihan yang sangat direkomendasikan bagi *developer* untuk membuat aplikasi web yang efesien, aman dan terstruktur.

### MySQL

Menurut Sidharta & Wibowo (2020) MySQL merupakan salah satu jenis DBMS (*Database Management System)* yang mengambil, mengelola dan menyimpan data menggunakan tabel. DBMS digunakan untuk mengolah data dalam jumlah besar atau *big data*. Secara umum, MySQL berguna sebagai *database*. MySQL bersifat *open source* yang menggunakan bahasa pemrograman berupa *Structured Query Language* (SQL). SQL merupakan *database* yang terstruktur dengan merelasikan beberapa tabel dalam *database* maupun merelasikan antar *database.*

Menurut Aipina & Witriyono (2022) MySQL memiliki fungsi *Relational Database Manajemen System* (RDBMS*)* yang menggunakan bahasa SQL untuk mengelola *database*. Bersifat *open source* dengan lisensi *General Public Licence* (GPL), sehingga mudah digunakan pengembang dengan arsitektur *client server* atau *embedded systems*.

MySQL mempunyai dua jenis lisensi, yaitu *Freesoftware* dan *Freeware* Hartati (2022). Keunggulan dari *Freesoftware* yaitu, dapat digunakan secara gratis tanpa biaya, bersifat *open source* dan cocok digunakan untuk mengembangkan sebuah proyek atau aplikasi, sedangkan *Freeware* ditawarkan oleh *Oracle Corpotation* dengan biaya tertentu yang cocok untuk membangun aplikasi komersial atau proyek yang bersifat tertutup.

MySQL sering digunakan dalam berbagai aplikasi seperti big data, sistem informasi, *E-commerce*, dan situs web dinamis. Database MySQL memiliki beberapa kelebihan dibandingkan dengan database lain, diantaranya sebagai berikut:

1. MySQL merupakan *Database Management System* (DBMS) dengan sistem keamanan yang baik.
2. MySQL mendukung *multi-user*, artinya penggunaan database tidak hanya digunakan oleh satu pihak melainkan dapat digunakan oleh banyak orang.
3. MySQL menyediakan field sebagai kunci primer dan kunci *unique.*
4. Mampu menyimpan data berkapasitas sangat besar hingga berukuran GigaByte.
5. Bersifat *open source*.
6. Mendukung berbagai macam data, seperti *integer, timestamp, date, float, text, character,* dan lain-lain.
7. Memiliki struktur tabel yang mudah dipakai dan fleksibel.
8. MySQL mendukung lebih banyak penggunaan *platfrom* dibandingkan SQL server.

### UML

Menurut Ronal dkk. (2022) UML (*Unified Modelling Language*) adalah sebuah bahasa pemodelan yang digunakan untuk menggambarkan sistem perangkat lunak. UML digunakan sebagai alat bantu untuk merancang, membangun, memvisualisasikan dan mendokumentasikan proses sistem secara visual. Sutrisno & Karnadi (2021) mendefinisikan UML meliputi *use case diagram, activity diagram, sequence diagram* dan *class diagram.* Berikut penjelasannya:

1. *Use case diagram*, digunakan untuk menggambarkan hubungan antara pengguna dan sistem yang akan dibuat. *Use case diagram* merupakan langkah awal untuk melakukan pemodelan yang menunjukkan fungsionalitas dan cara kerja suatu sistem.
2. *Activity diagram*, digunakan untuk memodelkan alur kerja dalam sistem yang sedang dirancang, seperti alur aktivitas dan alur tampilannya.
3. *Sequence diagram*, digunakan untuk menunjukkan urutan interaksi antar objek berupa pesan. Pada diagram ini fokus terhadap identifikasi metode didalam sistem dan memodelkan *use case.*
4. *Class diagram*, digunakan untuk menggambarkan struktur kelas, metode, hubungan antar kelas dan atribut.

Menurut Nistrina & Sahidah (2022) menjelaskan UML merupakan pemodelan dengan memvisualisasikan dan alur sistem dengan baik. Pemodelan tersebut dapat berupa kode-kode pemrograman berorientasi objek. UML tidak hanya bertujuan sebagai pengambilan keputusan desain tetapi juga dapat meningkatkan komunikasi di antara pengguna dan pengembang. Berikut konsep dasar UML yang dibagi menjadi 3 konsep yaitu, *structural classification, dynamic behavior* dan *model management.*

1. *Structural classification* adalah pemodelan yang meliputi struktur internal pada elemen-elemen statis dari sistem yang akan dibangun. Contohnya, *component diagram, deployment diagram* dan *class diagram*.
2. *Dynamic behavior* adalah diagram yang menggambarkan interaksi dan perubahan yang terjadi pada sistem. Contohnya, *sequence diagram, activity diagram* dan *state diagram.*
3. *Model management* adalah pendekatan dalam mengelola dan memastikan model struktur dan model perilaku selaras. Contohnya, hubungan antara *class diagram* dan *sequence diagram*.

### *Black-box Testing*

Menurut Wijaya & Astuti (2021) *black-box testing* adalah salah satu teknik pengujian yang mengutamakan fungsionalitas perangkat lunak dan kode program. Pengujian ini dilakukan untuk memastikan sistem bekerja dengan baik tanpa menemukan kesalahan pada program dan fungsi yang merugikan sehingga layak digunakan.

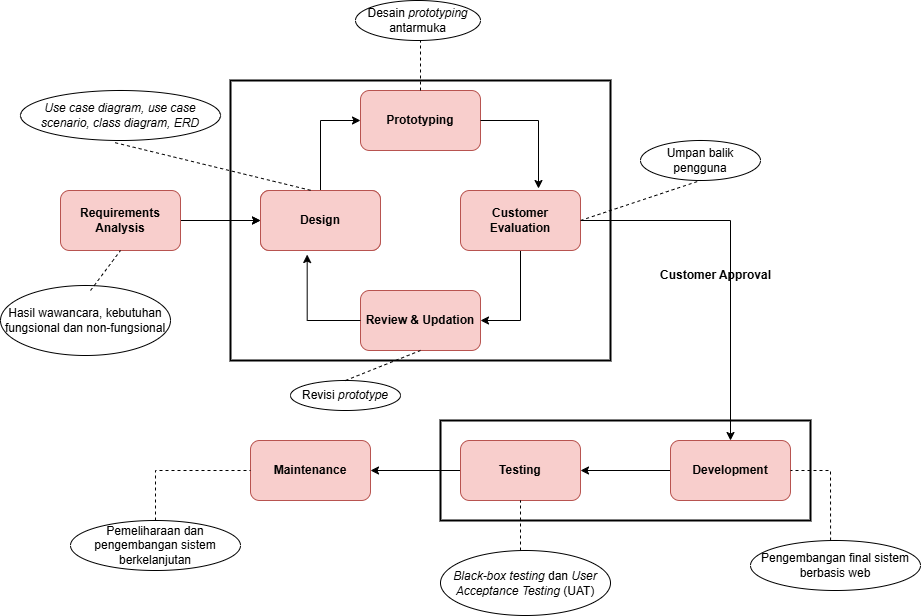
Abdillah dkk. (2023) menjelaskan bahwa *black-box testing* merupakan pengujian yang fokus pada fitur-fitur saja dan tidak memastikan perangkat lunak berfungsi sesuai dengan kebutuhan pengguna. Artinya, hanya berfokus pada fungsional dari sistem atau hasil akhir tanpa melihat data diproses oleh sistem itu sendiri.

*Black-box testing* adalah pengujian untuk memastikan hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode program dari perangkat lunak. Kelebihan utama dari *black-box testing* ini yaitu penguji tidak perlu memeriksa dan membongkar kode program serta dapat dilakukan oleh siapa saja.

# PERANCANGAN

## Desain Penelitian

Penelitian ini menggunakan metode *prototype* agar mempercepat proses pengembangan sistem dengan melakukan perbaikan berdasarkan umpan balik pengguna. Metode ini bertujuan untuk memastikan kebutuhan pengguna dipenuhi dengan baik melalui perancangan, evaluasi dan penyempurnaan *prototype* sebelum pengembangan sistem final. Tahapan dalam metode *prototype* ini yang digunakan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Desain Penelitian

Pertama, proses dilakukan dengan menyesuaikan kebutuhan melalui analisis kebutuhan untuk mengidentifikasi dan mengumpulkan informasi secara spesifik. Setelah itu, dilakukan perancangan desain awal sistem untuk menggambarkan secara cepat tentang konsep dasar sistem informasi akademik. Lalu pada *prototyping* mendesain antarmuka sistem yang dirancang sesuai kebutuhan *user*. Kemudian *prototype* ini akan dievaluasi oleh pengguna untuk mendapatkan umpan balik terhadap *prototype* yang sudah dibuat. Proses ini membantu mengevaluasi apakah *prototype* sudah sesuai dengan kebutuhan pengguna. Berdasarkan umpan balik pengguna, *prototype* diperbarui melalui tahap *review* yang akan dilakukan secara berulang-ulang hingga mendapatkan persetujuan dari pengguna. Setelah desain dan *prototype* disetujui, pengembangan sistem dilakukan secara lengkap berdasarkan desain yang sudah disetujui yang mencakup penulisan kode program, sehingga menghasilkan sistem yang siap untuk diuji lebih lanjut untuk memastikan sistem berjalan sesuai dengan harapan pengguna. Tahapan terakhir adalah pemeliharaan agar sistem berjalan dengan baik tanpa kendala setelah digunakan. Pada tahap ini mencakup memperbaiki masalah yang ditemukan dan melakukan perawatan agar sistem tetap aman, cepat dan berguna dalam jangka panjang.

Tahapan yang telah dilakukan meliputi analisis kebutuhan yang mencakup hasil wawancara, kebutuhan fungsional maupun non-fungsional, tahapan desain seperti *use case diagram, use case scenario, class diagram* dan perancangan ERD, serta tahapan *prototyping*. Sementara itu, tahapan *customer evaluation, review & updation, development, testing* dan *maintenance* akan dilakukan setelah sidang proposal.

## Requirements Analysis

Proses identifikasi kebutuhan pengguna dilakukan melalui wawancara. Melalui tahapan ini, peneliti dapat mengetahui kebutuhan pengguna terhadap sistem yang akan dirancang. Berikut tahapan identifikasi kebutuhan pengguna:

### **Hasil Wawancara**

Dengan wawancara yang telah dilakukan bersama dengan *user,* maka diperoleh kesimpulan dari berdasarkan kebutuhan yaitu:

1. Sistem informasi akademik siswa dapat digunakan admin untuk mengelola data guru, data siswa, kelas dan kalender akademik.
2. Guru dapat mengelola nilai siswa, absensi siswa dan melihat kalender akademik.
3. Siswa dapat menggunakan sistem untuk mengakses nilai, melihat absensi dan kalender akademik.
4. Kepala sekolah dapat mengakses visualisasi data akademik melalui *dashboard*.
5. *User* menginginkan tampilan sistem yang *user friendly.*

### **Kebutuhan Fungsional**

Kebutuhan fungsional sistem mengacu kepada fitur yang dimiliki oleh sistem. Kebutuhan fungsional dalam sistem diharuskan dapat memenuhi kebutuhan yang dimiliki pengguna. Kebutuhan fungsional yang dimiliki sistem dapat dilihat dalam Tabel 3.1.

Tabel 3.1 Kebutuhan Fungsional

|  |  |  |
| --- | --- | --- |
| **No** | **Aksi** | **Keterangan** |
| 1 | *Login* | * Setiap *user* dapat *login* menggunakan *email* dan *password* yang sudah terdaftar ke dalam sistem. |
| 2 | Melihat *dashboard* | * Antarmuka *dashboard* akan tampil kepada *user* yang *login* sesuai peran *user*. |
| 3 | Mengelola data siswa | * Admin dapat menambahkan data siswa. * Admin dapat mengubah data siswa. * Admin dapat menghapus data siswa. |
| 4 | Mengelola data guru | * Admin dapat menambah data guru. * Admin dapat mengubah data guru. * Admin dapat menghapus data guru. |
| 5 | Mengelola nilai siswa | * Guru dapat menginput nilai siswa. * Guru dapat mengubah nilai siswa. * Guru dapat menghapus nilai siswa. |
| 6 | Mengelola kelas | * Admin dapat menambah kelas. * Admin dapat mengubah kelas. * Admin dapat menghapus kelas. |
| 7 | Mengelola absensi siswa | * Guru dapat menginput absensi siswa. * Guru dapat mengubah absensi siswa. * Guru dapat menghapus absensi siswa. |
| 8 | Mengelola kalender akademik | * Admin dapat mengupload kalender akademik. |

### **Kebutuhan Non Fungsional**

Kebutuhan non fungsional berhubungan dengan bagaimana sistem menjalankan fungsinya atau kinerja yang dimiliki sistem. Berikut kebutuhan non fungsional yang dimiliki sistem informasi akademik sekolah dapat dilihat pada Tabel 3.2.

Tabel 3.2 Kebutuhan Non Fungsional

|  |  |  |
| --- | --- | --- |
| **No** | **Aksi** | **Keterangan** |
| 1 | Kegunaan | * Sistem mudah digunakan * Sistem memiliki antarmuka yang efektif dan interaksi. |
| 2 | Keamanan | * Sistem dapat menjaga atau melindungi data dari pihak yang tidak bertanggung jawab * Sistem hanya bisa memberi akses kepada pihak yang berkewajiban |
| 3 | Ketersediaan | * Sistem dapat diakses dimana dan kapan *user* memerlukannya. * Sistem dapat dibuka di berbagai *browser*, seperti Chrome, Firebox dan Microsoft edge. |
| 4 | Skabilitas | * Sistem dapat diintegrasikan dengan sistem lainnya seperti sistem keuangan. |
| 5 | Kinerja | * Sistem dapat mengolah data informasi akademik dalam ukuran besar. |

## Implementasi *Prototype*

### ***Design***

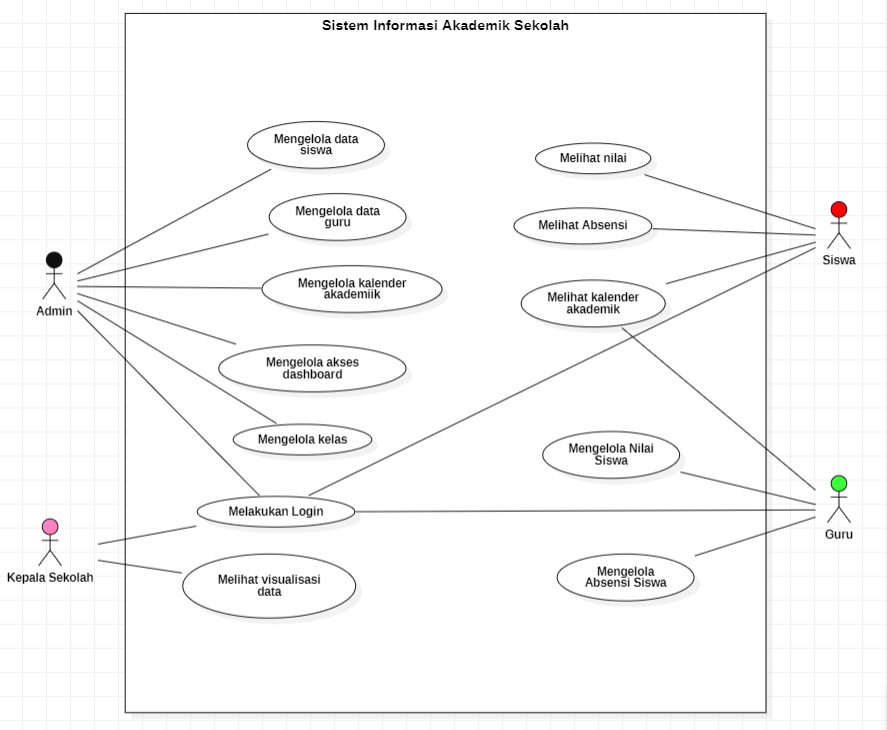
Setelah melakukan analisis terhadap kebutuhan pengguna sistem, maka dirancang desain awal atau desain cepat untuk menggambarkan secara cepat tentang konsep dasar sistem informasi akademik. Desain pada sistem informasi akademik meliputi beberapa aspek sebagai berikut.

1. Unified Modeling Language (UML)

Perancangan *Unified Modeling Language* (UML) terdiri dari diagram, berikut adalah jenis-jenis diagram yang digunakan untuk kebutuhan sistem antara lain:

1. Use Case Diagram

*Use case diagram*, digunakan untuk menggambarkan hubungan antara pengguna dan sistem yang akan dibuat. *Use case diagram* ini berfungsi untuk memberikan gambaran dari fitur-fitur sistem. Fitur-fitur sistem akan dijelaskan dalam *use case diagram* pada Gambar 3.2.



Gambar 3.2 Use Case Diagram

1. Use Case Scenario

Terdapat beberapa *use case scenario* yang dirancang pada penelitian ini. *Use case scenario* pada sistem informasi akademik sekolah pada setiap *user* sebagai berikut:

1. Melakukan *Login*

Aktor : Admin, Kepala Sekolah, Guru, Siswa.

Pre Kondisi : Aktor memiliki akun yang terdaftar di sistem*.*

Post Kondisi : Aktor berhasil masuk ke antarmuka *dashboard* sesuai peran*.*

Tabel 3.3 Melakukan Login

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** | |
| 1. Memasukkan email dan *password* |  |
|  | 1. Validasi email dan *password* |
|  | 1. Menampilkan *dashboard* |
| **Skenario alternatif: Jika data tidak valid** | |
|  | 1. Menampilkan pesan bahwa kesalahan |

1. Melihat Dashboard

Aktor : Admin, Kepala Sekolah.

Pre Kondisi : Aktor telah berhasil *login.*

Post Kondisi : *Dashboard* ditampilkan*.*

Tabel 3.4 Melihat Dashboard

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** | |
| 1. Aktor memilih menu *dashboard* |  |
|  | 1. Menampilkan *dashboard* |
| **Skenario alternatif: Jika data tidak valid** | |
|  | 1. Menampilkan pesan *error* |

1. Mengelola Data Siswa

Aktor : Admin

Pre Kondisi : Admin telah berhasil *login*.

Post Kondisi : Antarmuka data siswa.

Tabel 3.5 Mengelola Data Siswa

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** | |
| 1. Memilih menu data siswa |  |
|  | 1. Menampilkan antarmuka data siswa |
| 1. Klik *button* tambah data siswa |  |
|  | 1. Menampilkan *form* *input* data siswa |
| 1. Masukkan data siswa yang diperlukan |  |
|  | 1. Memvalidasi data masukan |
|  | 1. [a] Data siswa disimpan ke *database* |
|  | 1. [a] Menampilkan pesan data siswa berhasil disimpan |
| 1. Klik *button* *edit* |  |
|  | 1. Menampilkan *form edit* data siswa |
| 1. Mengubah data siswa dan menekan *button* simpan |  |
|  | 1. Melakukan validasi terhadap masukan dan perubahan data siswa |
|  | 1. [a] Menyimpan perubahan kedalam *database* |
|  | 1. [a] Menampilkan pesan data siswa berhasil diperbarui |
| 1. Klik *buttton* hapus |  |
|  | 1. Meminta konfirmasi penghapusan data siswa |
| 1. [a] Klik setuju untuk menghapus data siswa |  |
|  | 1. [a] Data siswa berhasil dihapus dari *database* |
|  | 1. Menampilkan pesan data siswa berhasil dihapus |
| **Skenario alternatif: Jika data tidak valid** | |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal menyimpan data siswa |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal melakukan perubahan data siswa |
| 1. [b] Klik batal untuk menghapus data siswa |  |
|  | 1. [b] Menampilkan pesan penghapusan dibatalkan pada data siswa |

1. Mengelola Kelas

Aktor : Admin

Pre Kondisi : Admin telah berhasil *login*.

Post Kondisi : Antarmuka kelas.

Tabel 3.6 Mengelola Kelas

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** | |
| 1. Memilih menu kelas |  |
|  | 1. Menampilkan antarmuka kelas |
| 1. Klik *button* tambah kelas |  |
|  | 1. Menampilkan tampilan tambah data kelas |
| 1. Masukkan kelas yang diperlukan |  |
|  | 1. Memvalidasi data masukan |
|  | 1. [a] Data kelas disimpan ke *database* |
|  | 1. [a] Menampilkan pesan data siswa berhasil disimpan |
| 1. Klik *button* *edit* |  |
|  | 1. Menampilkan *form edit* data kelas |
| 1. Mengubah data kelas dan menekan *button* simpan |  |
|  | 1. Melakukan validasi terhadap masukan dan perubahan data kelas |
|  | 1. [a] Menyimpan perubahan kedalam *database* |
|  | 1. [a] Menampilkan pesan data kelas berhasil diperbarui |
| 1. Klik *buttton* hapus |  |
|  | 1. Meminta konfirmasi penghapusan data kelas |
| 1. [a] Klik setuju untuk menghapus data kelas |  |
|  | 1. [a] Data kelas berhasil dihapus dari *database* |
|  | 1. Menampilkan pesan data kelas berhasil dihapus |
| **Skenario alternatif: Jika data tidak valid** | |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal menyimpan data kelas |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal melakukan perubahan data kelas |
| 1. [b] Klik batal untuk menghapus data kelas |  |
|  | 1. [b] Menampilkan pesan penghapusan dibatalkan pada data kelas |

1. Mengelola Data Guru

Aktor : Admin

Pre Kondisi : Admin telah berhasil *login*.

Post Kondisi : Antarmuka data guru.

Tabel 3.7 Mengelola Data Guru

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** | |
| 1. Memilih menu data guru |  |
|  | 1. Menampilkan antarmuka data guru |
| 1. Klik *button* tambah data guru |  |
|  | 1. Menampilkan *form* *input* data guru |
| 1. Masukkan data guru yang diperlukan |  |
|  | 1. Memvalidasi data masukan |
|  | 1. [a] Data guru disimpan ke *database* |
|  | 1. [a] Menampilkan pesan data guru berhasil disimpan |
| 1. Klik *button* *edit* |  |
|  | 1. Menampilkan *form edit* data guru |
| 1. Mengubah data guru dan menekan *button* simpan |  |
|  | 1. Melakukan validasi terhadap masukan dan perubahan data guru |
|  | 1. [a] Menyimpan perubahan kedalam *database* |
|  | 1. [a] Menampilkan pesan data guru berhasil diperbarui |
| 1. Klik *buttton* hapus |  |
|  | 1. Meminta konfirmasi penghapusan data guru |
| 1. [a] Klik setuju untuk menghapus data guru |  |
|  | 1. [a] Data guru berhasil dihapus dari *database* |
|  | 1. Menampilkan pesan data guru berhasil dihapus |
| **Skenario alternatif: Jika data tidak valid** | |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal menyimpan data guru |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal melakukan perubahan data guru |
| 1. [b] Klik batal untuk menghapus data guru |  |
|  | 1. [b] Menampilkan pesan penghapusan dibatalkan pada data guru |

1. Mengelola Absensi Siswa

Aktor : Guru

Pre Kondisi : Guru telah berhasil *login*.

Post Kondisi : Antarmuka data absensi siswa.

Tabel 3.8 Mengelola Absensi Siswa

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal** | |
| 1. Guru memilih menu absensi siswa |  |
|  | 1. Menampilkan daftar absensi siswa |
| 1. Guru memilih *button* tambah absensi siswa |  |
|  | 1. Menampilkan *form* *input* absensi siswa |
| 1. Guru memasukkan absensi siswa dan menyimpannya |  |
|  | 1. [a] Absensi siswa disimpan ke *database* |
|  | 1. [a] Menampilkan pesan absensi siswa berhasil disimpan |
| 1. Guru klik *button* *edit* |  |
|  | 1. Menampilkan *form edit* absensi siswa |
| 1. Guru mengubah absensi siswa dan menekan *button* simpan |  |
|  | 1. Melakukan validasi terhadap masukan dan perubahan absensi siswa |
|  | 1. [a] Menyimpan perubahan kedalam *database* |
|  | 1. [a] Menampilkan pesan absensi siswa berhasil diperbarui |
| 1. Guru klik *buttton* hapus |  |
|  | 1. Meminta konfirmasi penghapusan absensi siswa |
| 1. [a] Klik setuju untuk menghapus absensi siswa |  |
|  | 1. [a] Absensi siswa berhasil dihapus dari *database* |
|  | 1. Menampilkan pesan absensi siswa berhasil dihapus |
| **Skenario alternatif: Jika data tidak valid** | |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal menyimpan absensi siswa |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal melakukan perubahan absensi siswa |
| 1. [b] Klik batal untuk menghapus absensi siswa |  |
|  | 1. [b] Menampilkan pesan penghapusan dibatalkan pada absensi siswa |

1. Mengelola Nilai Siswa

Aktor : Guru

Pre Kondisi : Guru telah berhasil *login*.

Post Kondisi : Antarmuka nilai siswa.

Tabel 3.9 Mengelola Nilai Siswa

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal (Guru)** | |
| 1. Guru memilih menu nilai siswa |  |
|  | 1. Menampilkan daftar nilai siswa |
| 1. Guru memilih *button* tambah nilai siswa |  |
|  | 1. Menampilkan *form* *input* nilai siswa |
| 1. Guru memasukkan nilai siswa dan menyimpannya |  |
|  | 1. [a] Nilai siswa disimpan ke *database* |
|  | 1. [a] Menampilkan pesan nilai siswa berhasil disimpan |
| 1. Guru klik *button* *edit* |  |
|  | 1. Menampilkan *form edit* nilai siswa |
| 1. Guru mengubah nilai siswa dan menekan *button* simpan |  |
|  | 1. Melakukan validasi terhadap masukan dan perubahan nilai siswa |
|  | 1. [a] Menyimpan perubahan kedalam *database* |
|  | 1. [a] Menampilkan pesan nilai siswa berhasil diperbarui |
| 1. Guru klik *buttton* hapus |  |
|  | 1. Meminta konfirmasi penghapusan nilai siswa |
| 1. [a] Klik setuju untuk menghapus nilai siswa |  |
|  | 1. [a] Nilai siswa berhasil dihapus dari *database* |
|  | 1. Menampilkan pesan nilai siswa berhasil dihapus |
| **Skenario alternatif: Jika data tidak valid** | |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal menyimpan nilai siswa |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal melakukan perubahan nilai siswa |
| 1. [b] Klik batal untuk menghapus nilai siswa |  |
|  | 1. [b] Menampilkan pesan penghapusan dibatalkan pada nilai siswa |

1. Mengelola Kalender Akademik

Aktor : Admin, Guru, Siswa

Pre Kondisi : Admin, Guru, Siswa telah berhasil *login*.

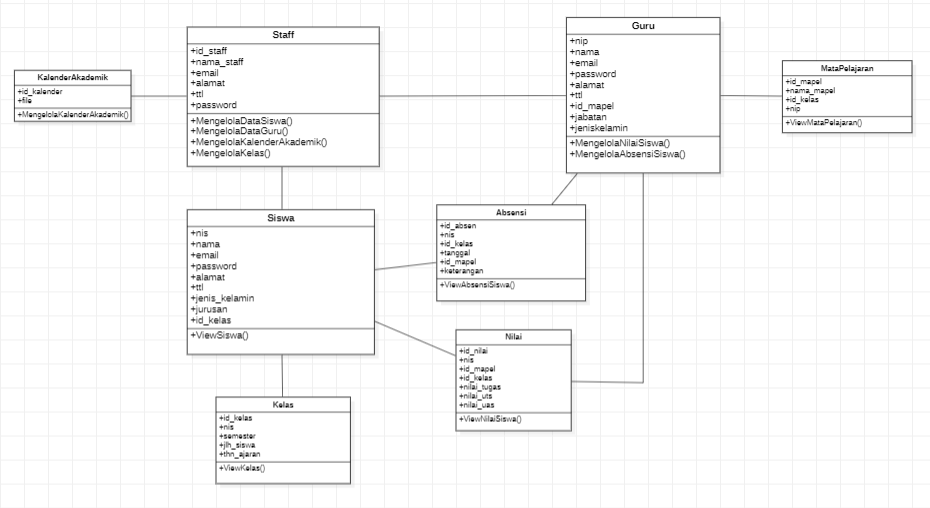
Post Kondisi : Antarmuka kalender akademik.

Tabel 3.10 Mengelola Kalender Akademik

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario normal (Admin)** | |
| 1. Admin memilih menu kalender akademik |  |
|  | 1. Menampilkan antarmuka kalender akademik |
| 1. Admin menekan menu *upload* kalender akademik |  |
|  | 1. Menampilkan *form upload* file kalender akademik |
| 1. Admin memilih file kalender akademik dan menyimpannya |  |
|  | 1. [a] File kalender akademik disimpan kedalam *database* |
|  | 1. [a] Menampilkan pesan file kalender akademik berhasil disimpan. |
| **Skenario normal (Guru/Siswa)** | |
| 1. Memilih menu kalender akademik |  |
|  | 1. Menampilkan kalender akademik |
| **Skenario alternatif: Jika data tidak valid** | |
|  | 1. [b] Menampilkan pesan data tidak valid |
|  | 1. [b] Menampilkan pesan gagal menyimpan kalender akademik |

1. Class diagram

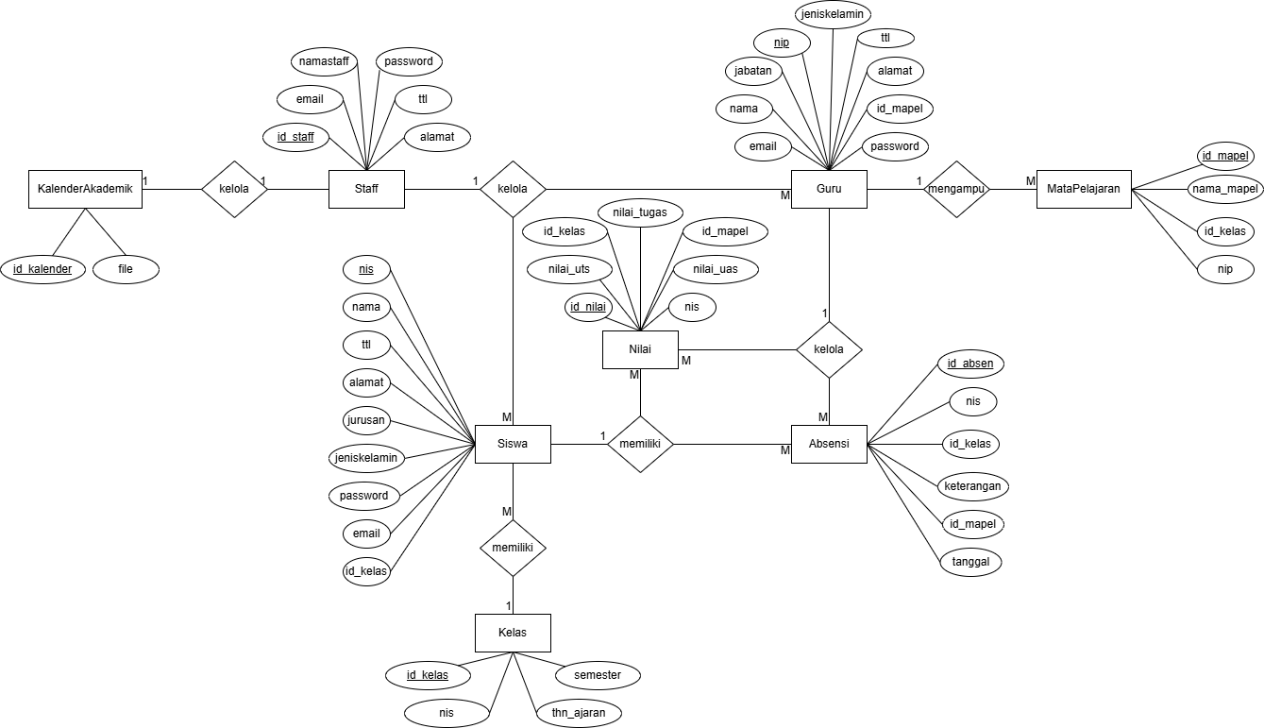
*Class diagram*, digunakan untuk menggambarkan struktur kelas, metode, hubungan antar kelas dan atribut. Berikut adalah *class diagram* dari sistem yang dirancang dapat dilihat pada Gambar 3.3.



Gambar 3.3 Class Diagram

1. Perancangan Entity-Relationship Diagram (ERD)

*Entity-relationship diagram* (ERD) dirancang sesuai dengan hasil analisa sistem informasi akademik sekolah. ERD yang akan digunakan dalam sistem SIAK dapat dilihat pada Gambar 3.4.



Gambar 3.4 ERD

### ***Prototyping***

Setelah analisis dan desain awal, maka tahap selanjutnya adalah merancang antarmuka sistem. Antarmuka sistem dirancang sesuai kebutuhan *user.* Tampilan antarmuka sistem dapat diilustrasikan sebagai berikut.

1. Antarmuka Form Login

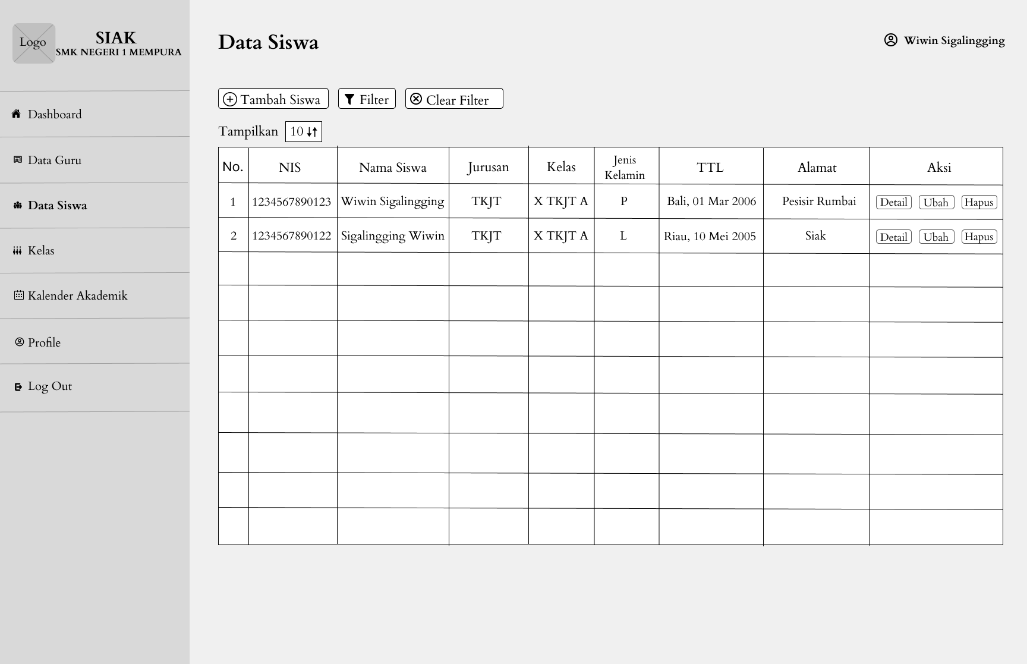
Antarmuka *form login* digunakan untuk mengirimkan email dan *password* ke dalam sistem, data hasil inputan kemudian divalidasi. Antarmuka *form login* ini digunakan untuk pengguna dan hanya diakses oleh pengguna yang memiliki hak akses. Antarmuka *form login* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Antarmuka Form Login

1. Antarmuka Data Siswa

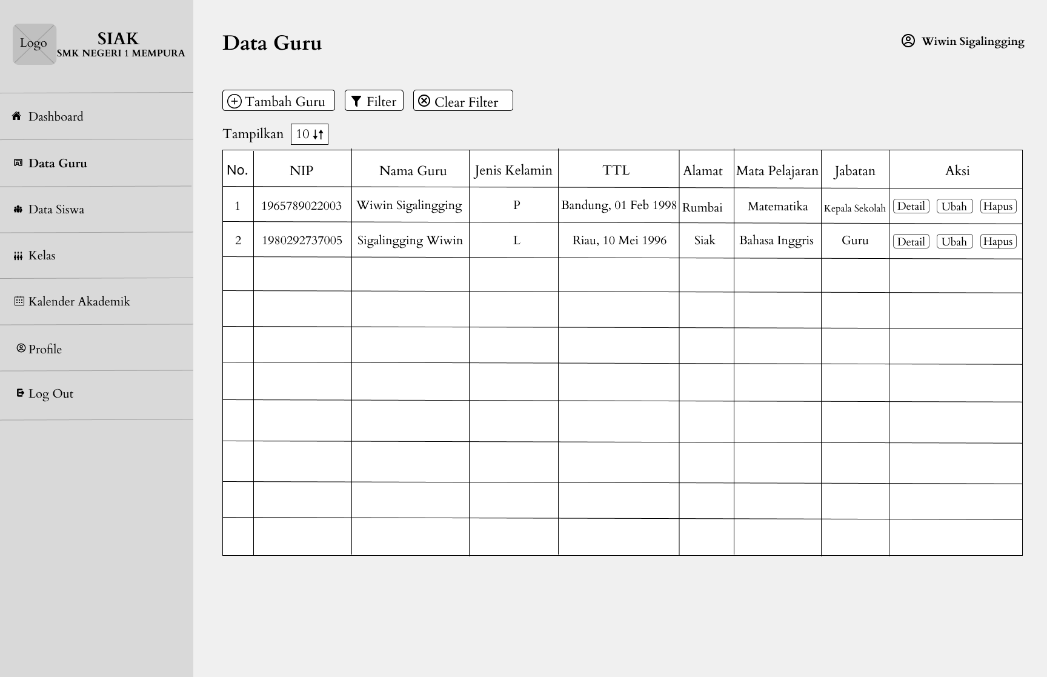
Antarmuka data siswa ini dikelola oleh admin untuk memudahkan dalam menambah, mengubah atau menghapus data siswa. Pada antarmuka ini menampilkan NIS, nama siswa, jurusan, kelas, jenis kelamin, tempat tanggal lahir, dan alamat. Antarmuka data siswa dapat dilihat pada Gambar 3.6.



Gambar 3.6 Antarmuka Data Siswa

1. Antarmuka Data Guru

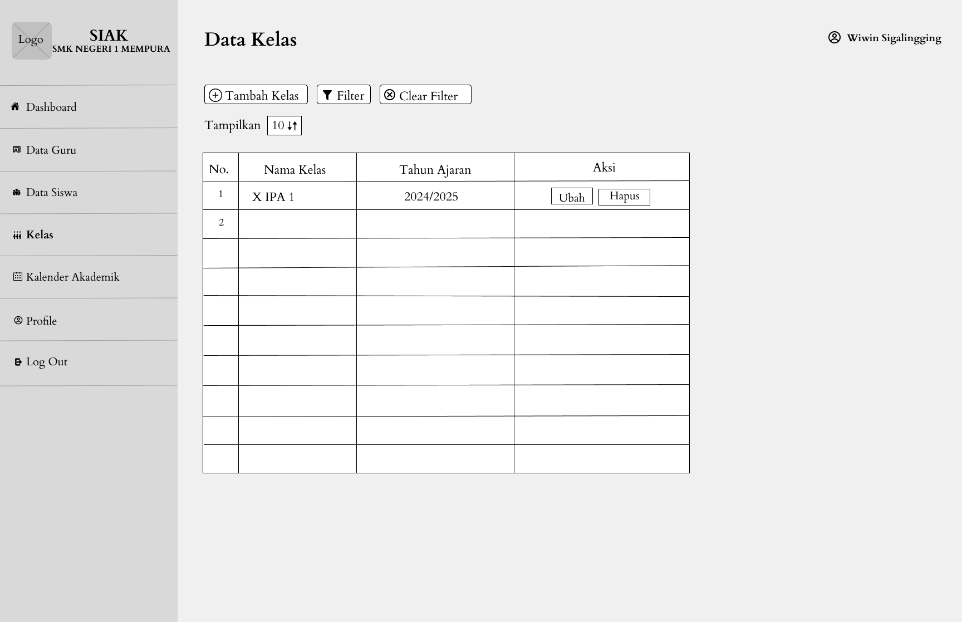
Antarmuka data guru ini digunakan oleh admin untuk memudahkan dalam mengelola data guru. Pada antarmuka ini menampilkan NIP, nama guru, jenis kelamin, tempat tanggal lahir, alamat, mata pelajaran dan jabatan. Selain itu, pada antarmuka data guru juga menyediakan fitur untuk menambah, mengubah, dan menghapus data guru. Antarmuka data guru dapat dilihat pada Gambar 3.7.



Gambar 3.7 Antarmuka Data Guru

1. Antarmuka Kelas

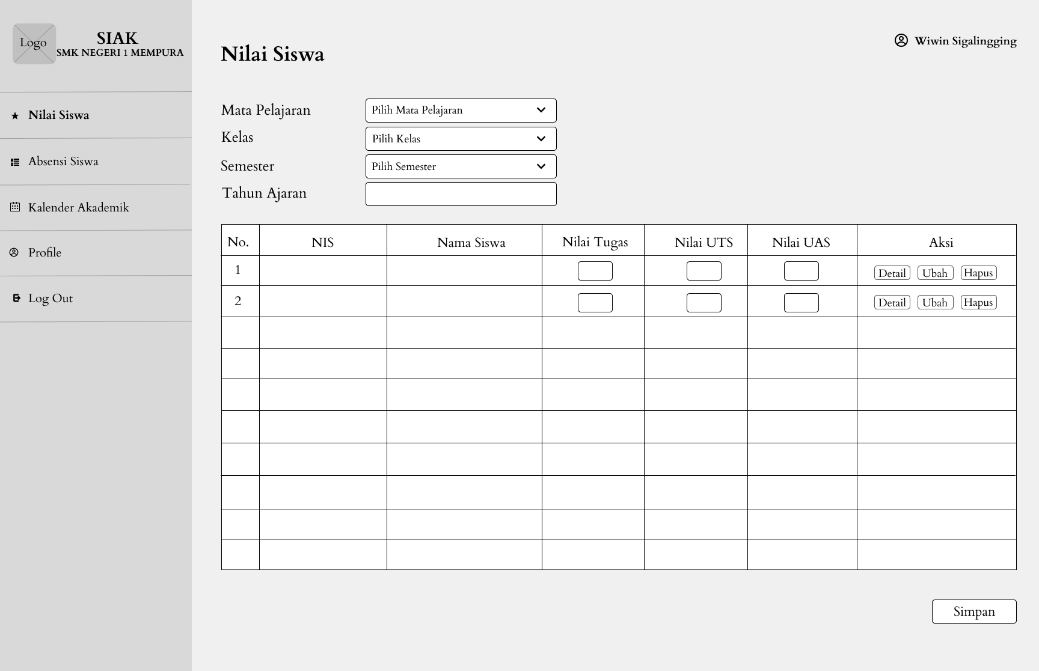
Antarmuka berikut ini merupakan antarmuka yang menampilkan data kelas. Pada antarmuka ini, admin dapat mengelola data kelas, seperti nama kelas dan tahun ajaran. Antarmuka ini juga menyediakan fitur tambah, ubah dan hapus. Antarmuka kelas dapat dilihat pada Gambar 3.8.



Gambar 3.8 Antarmuka Kelas

1. Antarmuka Nilai Siswa

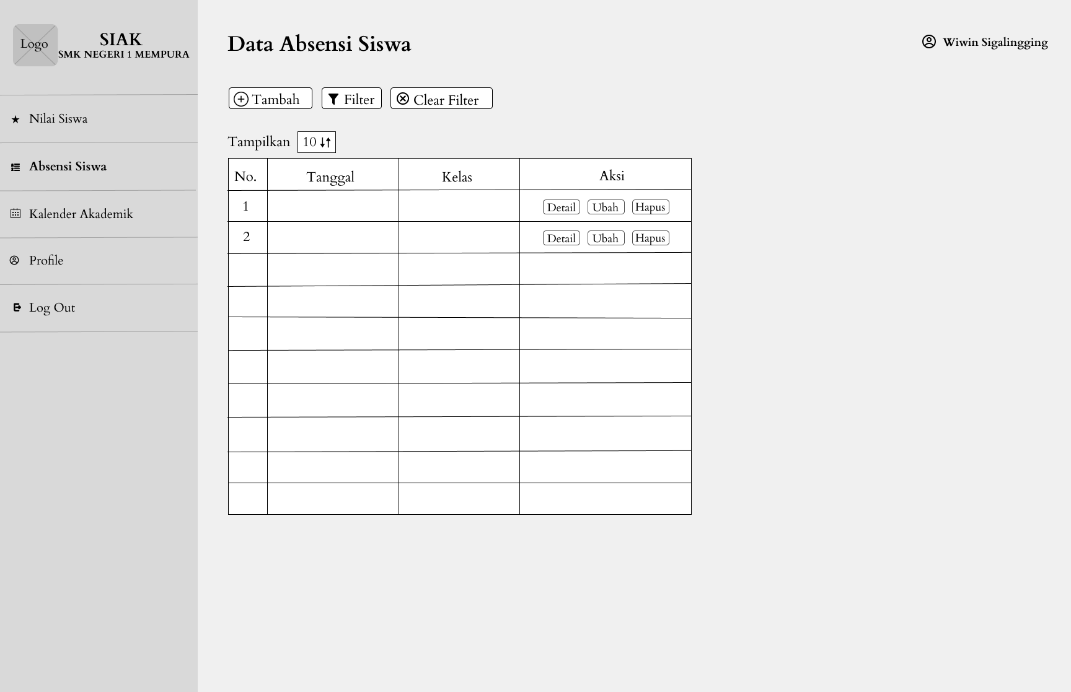
Antarmuka nilai siswa merupakan antarmuka yang digunakan untuk mengelola nilai siswa, seperti nilai tugas, nilai uts dan nilai uas. Antarmuka ini akan dikelola guru yang menyediakan fitur tambah, ubah dan hapus. Antarmuka nilai siswa dapat dilihat pada Gambar 3.9.



Gambar 3.9 Antarmuka Nilai Siswa

1. Antarmuka Absensi Siswa

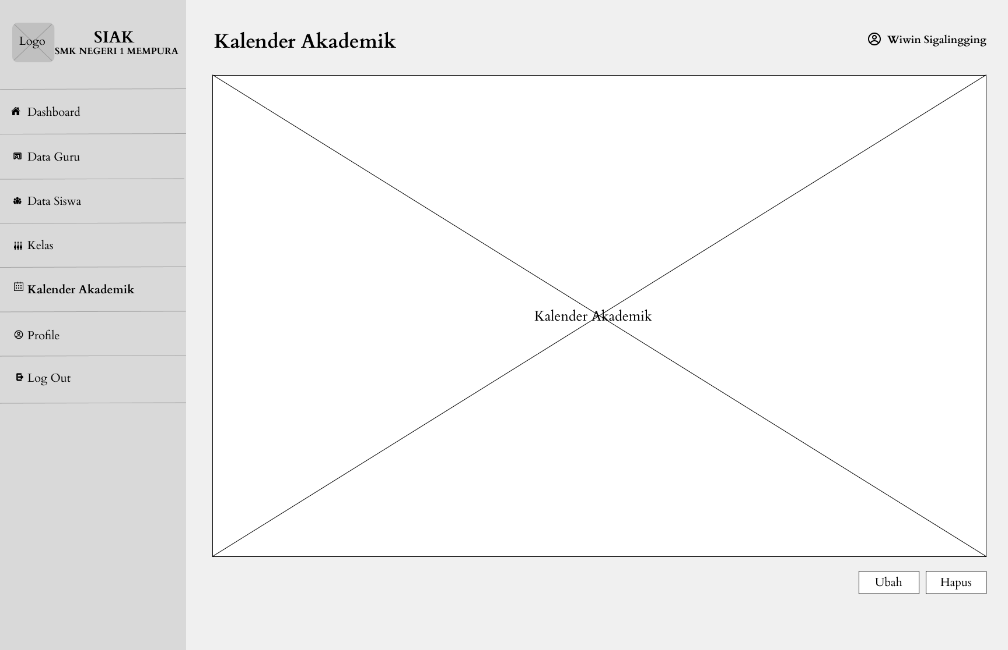
Antarmuka absensi siswa merupakan antarmuka yang digunakan untuk mengelola absensi siswa. Pada tampilan ini guru dapat menambah, mengubah dan menghapus data absensi siswa. Antarmuka absensi siswa dapat dilihat pada Gambar 3.10.



Gambar 3.10 Absensi Siswa

1. Antarmuka Kalender Akademik

Antarmuka kalender akademik merupakan antarmuka yang digunakan untuk mengelola kalender akademik. Pada tampilan ini admin dapat mengupload, mengubah atau menghapus kalender akademik. Antarmuka kalender akademik dapat dilihat pada Gambar 3.11.



Gambar 3.11 Kalender Akademik

### ***Customer Evaluation***

Evaluasi perancangan dilakukan melalui sesi wawancara dengan ibu Richa Sulastri, disertai dengan menampilkan desain *prototype*. Pada tahapan ini akan dilakukan setelah sidang proposal.

### ***Review* & *Updation***

Setelah melakukan evaluasi perancangan terhadap *user,* maka tahap selanjutnya adalah tahap *review* dan *updation.* Pada tahapan ini akan ditampilkan fitur-fitur *prototype* yang telah diperbaiki dan disetujui oleh *user.* Kegiatan ini akan dilakukan setelah sidang proposal.

## Development & Testing

### ***Development***

Pada tahap pengembangan, sistem mulai dibangun berdasarkan *prototype* yang sudah disetujui pengguna. Sistem ini mencakup fitur-fitur seperti pengelolaan data siswa, data guru, nilai, absensi, kelas, dan kalender akademik. Proses ini akan dilaksanakan setelah sidang proposal.

### ***Testing***

Pengujian pada penelitian ini menggunakan *black-box testing* dan pengujian sistem yang dirancang juga menggunakan pengujian *User Acceptance Testing (*UAT), berikut perancangan dari pengujian sistem;

1. Black-box Testing

Pengujian ini berfokus pada pemberian kondisi pada sistem, kemudian terdapat harapan hasil dari inputan. Hasil dari pengujian dibandingkan dengan harapan hasil untuk analisa. Berikut perancangan pengujian menggunakan *decision table testing*.

Tabel 3.11 Black-box Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Fitur** | **Skenario** | **Hasil yang diharapkan** | **Kesimpulan** |
| Melakukan *login* | Menginputkan email dan *password* yang benar | Menampilkan *dashboard* | [ ] Berhasil  [ ] Tidak Berhasil |
| Menginputkan email atau *password* yang salah | Menampilkan pesan dan peringatan email dan *password* salah | [ ] Berhasil  [ ] Tidak Berhasil |
| Mengelola data siswa | Memilih halaman data siswa | Menampilkan halaman daftar data siswa | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* tambah data siswa | Menampilkan *form input* data siswa | [ ] Berhasil  [ ] Tidak Berhasil |
| Menyimpan data *form* kosong | Menampilkan pesan *error form validation* | [ ] Berhasil  [ ] Tidak Berhasil |
| Menyimpan data siswa dengan benar | Menyimpan data dalam *database* dan menampilkan daftar siswa terbaru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* ubah data siswa | Menampilkan *pop-up form* ubah data siswa yang dipilih | [ ] Berhasil  [ ] Tidak Berhasil |
| Melakukan perubahan terhadap data siswa | Menyimpan data dalam *database* dan menampilkan data siswa terbaru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* hapus data siswa | Menampilkan konfirmasi penghapusan data siswa yang dipilih | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* detail data siswa | Menampilkan *form* detail data siswa | [ ] Berhasil  [ ] Tidak Berhasil |
| Mengelola data guru | Memilih halaman data guru | Menampilkan halaman daftar data guru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* tambah data guru | Menampilkan *form input* data guru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menyimpan data *form* kosong | Menampilkan pesan *error form validation* | [ ] Berhasil  [ ] Tidak Berhasil |
| Menyimpan data guru dengan benar | Menyimpan data dalam *database* dan menampilkan daftar guru terbaru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* ubah data guru | Menampilkan *pop-up form* ubah data guru yang dipilih | [ ] Berhasil  [ ] Tidak Berhasil |
| Melakukan perubahan terhadap data guru | Menyimpan data dalam *database* dan menampilkan data guru terbaru | [ ] Berhasil  [ ] Tidak Berhasil |
| Mengelola nilai siswa | Memilih halaman nilai siswa | Menampilkan halaman daftar nilai siswa | [ ] Berhasil  [ ] Tidak Berhasil |
| Menginputkan nilai siswa pada halaman nilai siswa dan menyimpan perubahan | Menyimpan data dalam *database* dan menampilkan nilai siswa terbaru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* ubah nilai siswa dan menyimpan perubahan | Menyimpan data dalam *database* dan menampilkan nilai siswa terbaru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* hapus nilai siswa | Menampilkan konfirmasi penghapusan nilai siswa yang dipilih | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* detail nilai sisswa | Menampilkan *form* detail nilai siswa | [ ] Berhasil  [ ] Tidak Berhasil |
| Mengelola absensi siswa | Memilih halaman absensi siswa | Menampilkan halaman daftar absensi siswa | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* tambah absensi siswa | Menampilkan *form input* absensi siswa | [ ] Berhasil  [ ] Tidak Berhasil |
| Menyimpan absensi *form* kosong | Menampilkan pesan *error form validation* | [ ] Berhasil  [ ] Tidak Berhasil |
| Menyimpan absensi siswa dengan benar | Menyimpan data dalam *database* dan menampilkan daftar absensi siswa terbaru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* ubah absensi siswa | Menampilkan *form* ubah absensi siswa yang dipilih | [ ] Berhasil  [ ] Tidak Berhasil |
| Melakukan perubahan terhadap absensi siswa | Menyimpan absensi dalam *database* dan menampilkan absensi siswa terbaru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* hapus absensi siswa | Menampilkan konfirmasi penghapusan absensi siswa yang dipilih | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* detail absensi siswa | Menampilkan *form* detail absensi siswa | [ ] Berhasil  [ ] Tidak Berhasil |
| Mengelola kalender akademik | Memilih halaman kalender akademik | Menampilkan halaman kalender akademik | [ ] Berhasil  [ ] Tidak Berhasil |
| *Upload* file pdf berisi kalender akademik | Menampilkan *form upload* kalender akademik | [ ] Berhasil  [ ] Tidak Berhasil |
| *Upload* file dengan format yang berbeda | Menampilkan pesan keterangan file berbeda | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* hapus kalender akademik | Menampilkan konfirmasi penghapusan absensi siswa yang dipilih | [ ] Berhasil  [ ] Tidak Berhasil |
| Mengelola kelas | Memilih halaman kelas | Menampilkan halaman kelas | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* tambah kelas | Menampilkan *form input* tambah kelas | [ ] Berhasil  [ ] Tidak Berhasil |
| Menyimpan halaman kelas kosong | Menampilkan pesan *error form validation* | [ ] Berhasil  [ ] Tidak Berhasil |
| Menyimpan kelas dengan benar | Menyimpan data dalam *database* dan menampilkan daftar kelas siswa terbaru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* ubah kelas | Menampilkan *form* ubah kelas yang dipilih | [ ] Berhasil  [ ] Tidak Berhasil |
| Melakukan perubahan terhadap kelas | Menyimpan kelas dalam *database* dan menampilkan kelas terbaru | [ ] Berhasil  [ ] Tidak Berhasil |
| Menekan *button* hapus kelas | Menampilkan konfirmasi penghapusan kelas yang dipilih | [ ] Berhasil  [ ] Tidak Berhasil |

1. User Acceptance Testing (UAT)

Pengujian sistem yang dirancang juga menggunakan pengujian *User Acceptance Test* (UAT). Pengujian ini dilakukan dengan memberikan pertanyaan berupa kuesioner kepada pengguna. Kuesioner ini akan diberikan kepada Ibu Richa Sulastri selaku pengguna sistem. Berikut pengujian menggunakan UAT dapat dilihat pada Tabel 3.10.

Tabel 3.12 Bobot Jawaban Kuesioner UAT

|  |  |
| --- | --- |
| **Keterangan** | **Bobot** |
| Sangat Setuju | 5 |
| Setuju | 4 |
| Netral | 3 |
| Kurang Setuju | 2 |
| Sangat Tidak Setuju | 1 |

###### **Admin**

Beberapa pertanyaan yang disampaikan untuk memastikan apakah sudah sesuai dengan kebutuhan dari klien.

Tabel 3.13 Usability Testing (Admin)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kode** | **Pertanyaan** | **1** | **2** | **3** | **4** | **5** |
| **Learnability** | | | | | | |
| P1 | Sistem ini mudah untuk digunakan |  |  |  |  |  |
| P2 | Saya dapat menggunakan sistem ini tanpa bantuan |  |  |  |  |  |
| P3 | Saya tidak mengalami kendala saat menggunakan sistem ini |  |  |  |  |  |
| **Efficiency** | | | | | | |
| P4 | Saya dapat menyelesaikan tugas (seperti menambah, mengedit, atau menghapus data) dengan cepat menggunakan sistem ini. |  |  |  |  |  |
| P5 | Sistem ini memungkinkan saya untuk menemukan data yang diperlukan tanpa membuang banyak waktu. |  |  |  |  |  |
| **Memorability** | | | | | | |
| P6 | Sistem ini mudah dipahami |  |  |  |  |  |
| P7 | Saya dapat dengan mudah mengingat bagaimana cara menggunakan sistem ini |  |  |  |  |  |
| **Errors** | | | | | | |
| P8 | Tidak terdapat kesalahan pada tiap halaman |  |  |  |  |  |
| P9 | Terdapat pesan yang jelas ketika ada kesalahan |  |  |  |  |  |
| **Satisfication** | | | | | | |
| P10 | Sistem ini nyaman untuk digunakan |  |  |  |  |  |
| P11 | Sistem ini membuat pekerjaan saya lebih cepat dan efisien dibandingkan sebelumnya. |  |  |  |  |  |
| P12 | Sistem ini dapat menampilkan statistik sekolah |  |  |  |  |  |

###### **Guru**

Beberapa pertanyaan yang disampaikan untuk memastikan apakah sudah sesuai dengan kebutuhan dari klien.

Tabel 3.14 Usability Testing (Guru)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kode** | **Pertanyaan** | **1** | **2** | **3** | **4** | **5** |
| **Learnability** | | | | | | |
| P1 | Sistem ini mudah untuk digunakan |  |  |  |  |  |
| P2 | Saya dapat menggunakan sistem ini tanpa bantuan |  |  |  |  |  |
| P3 | Saya tidak mengalami kendala saat menggunakan sistem ini |  |  |  |  |  |
| **Efficiency** | | | | | | |
| P4 | Sistem ini memudahkan saya dalam menginput, mengubah dan menghapus nilai dan absensi siswa |  |  |  |  |  |
| P5 | Sistem ini memungkinkan saya menyelesaikan tugas dengan cepat |  |  |  |  |  |
| **Memorability** | | | | | | |
| P6 | Sistem ini mudah dipahami |  |  |  |  |  |
| P7 | Saya dapat dengan mudah mengingat bagaimana cara menggunakan sistem ini |  |  |  |  |  |
| **Errors** | | | | | | |
| P8 | Tidak terdapat kesalahan pada tiap halaman |  |  |  |  |  |
| P9 | Terdapat pesan yang jelas ketika ada kesalahan |  |  |  |  |  |
| **Satisfication** | | | | | | |
| P10 | Sistem ini nyaman untuk digunakan |  |  |  |  |  |
| P11 | Sistem ini membuat pekerjaan saya lebih cepat dan efisien dibandingkan sebelumnya. |  |  |  |  |  |
| P12 | Saya merasa puas dengan fitur-fitur sistem ini |  |  |  |  |  |
| P13 | Sistem ini membantu saya dalam mengelola nilai dan absensi dengan baik |  |  |  |  |  |

###### **Siswa**

Beberapa pertanyaan yang disampaikan untuk memastikan apakah sudah sesuai dengan kebutuhan dari klien.

Tabel 3.15 Usability Testing (Siswa)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kode** | **Pertanyaan** | **1** | **2** | **3** | **4** | **5** |
| **Learnability** | | | | | | |
| P1 | Sistem ini mudah untuk digunakan |  |  |  |  |  |
| P2 | Saya dapat menggunakan sistem ini tanpa bantuan |  |  |  |  |  |
| P3 | Saya tidak mengalami kendala saat menggunakan sistem ini |  |  |  |  |  |
| **Efficiency** | | | | | | |
| P4 | Saya dapat melihat nilai dan absensi secara berkala |  |  |  |  |  |
| P5 | Sistem ini memungkinkan saya untuk melihat kalender akademik untuk mengetahui tanggal penting dalam sekolah |  |  |  |  |  |
| **Memorability** | | | | | | |
| P6 | Sistem ini mudah dipahami |  |  |  |  |  |
| P7 | Saya dapat dengan mudah mengingat bagaimana cara menggunakan sistem ini |  |  |  |  |  |
| **Errors** | | | | | | |
| P8 | Tidak terdapat kesalahan pada tiap halaman |  |  |  |  |  |
| P9 | Terdapat pesan yang jelas ketika ada kesalahan |  |  |  |  |  |
| **Satisfication** | | | | | | |
| P10 | Sistem ini nyaman untuk digunakan |  |  |  |  |  |

## *Maintenance*

Tahap pemeliharaan dilakukan setelah sistem selesai dirancang dan diimplementasikan. Pada tahap ini, sistem akan dipantau secara berkala untuk memastikan kinerja tetap optimal dan mendeteksi adanya *bug* atau masalah yang mungkin muncul. Jika ditemukan kendala, perbaikan akan dilakukan untuk menjaga agar sistem tetap berjalan sesuai dengan kebutuhan pengguna.

# JADWAL DAN ANGGARAN BIAYA

## Jadwal

Proses pengerjaan proposal akhir memiliki jadwal yang dirancang oleh peneliti. Penjadwalan penelitian ini bertujuan agar penggunaan lebih efektif dan sesuai dengan metodologi yang digunakan. Jadwal pengerjaan dapat dilihat pada Tabel 4.1.

Tabel 4.1 Jadwal

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tahapan** | **November** | | | | **Desember** | | | | **Januari** | | | | **Februari** | | | | **Maret** | | | | **April** | | | | **Mei** | | | | **Juni** | | | |
| 1 | Bimbingan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Pembuatan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Seminar Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Pembuatan Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Pengujian dan Analisa |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Penulisan Laporan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Sidang  Proyek Akhir |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Perkiraan Biaya

Penelitian ini membutuhkan biaya untuk mendukung penyelesaian. Perkiraan anggaran biaya yang dikeluarkan dalam pengerjaan penelitian ini dapat dilihat pada Tabel 4.2.

Tabel 4.2 Perkiraan Biaya

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Barang** | **Jumlah** | **Harga** | **Keterangan** |
| 1 | Laptop Asus | 1 | Rp0,00 | Milik Pribadi |
| 2 | Internet | 1 | Rp250.000,00 | Untuk 3 bulan |
| 3 | Hosting | 3 | Rp100.000,00 | Untuk 3 bulan |
| 4 | Biaya Pengembangan | 480 jam | Rp6.240.000,00 | Untuk 4 bulan |
| Total Biaya | | | Rp6.590.000,00 |  |

# DAFTAR PUSTAKA

Abdillah, M. T., Kurniastuti, I., Susanto, A., & Yudianto, F. (2023). Implementasi Black box Testing dan Usability Testing pada Website Sekolah MI Miftahul Ulum Warugunung Surabaya. *Jurnal Ilmu Komputer dan Desain Komunikasi Visual*, *8*(1).

Ambriani, D., & Nurhidayat, A. I. (2020). RANCANG BANGUN REPOSITORY PUBLIKASI ILMIAH DOSEN BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL. Dalam *Jurnal Manajemen Informatika* (Vol. 10, Nomor 01).

Arizal, A., Puteri, A. N., Zakiyabarsi, F., & Priambodo, D. F. (2022). Metode Prototype pada Sistem Informasi Manajemen Tugas Akhir Mahasiswa Berbasis Website. *Jurnal Teknologi Informasi dan Komunikasi (TIKomSiN)*, *10*(1). https://doi.org/10.30646/tikomsin.v10i1.606

Desi Aprianto, & Kiki Rizki Nova Wardani. (2021). *PENGGUNAAN METODE PROTOTYPING DALAM PENGEMBANGAN SISTEM INFORMASI PERPUSTAKAAN DIGITAL*.

Desma Aipina, & Harry Witriyono. (2022). PEMANFAATAN FRAMEWORK LARAVEL DAN FRAMEWORK BOOTSTRAP PADA PEMBANGUNAN APLIKASI PENJUALAN HIJAB BERBASIS WEB. *Jurnal Media Infotama*, *18*(1).

Dwi Wijaya, Y., & Wardah Astuti, M. (2021). PENGUJIAN BLACKBOX SISTEM INFORMASI PENILAIAN KINERJA KARYAWAN PT INKA (PERSERO) BERBASIS EQUIVALENCE PARTITIONS. *Jurnal Digital Teknologi Informasi*, *4*, 2686–9706.

Erlang Bagus Sadewa, Wijiyanto, & Nurohman. (2024). SISTEM INFORMASI AKADEMIK BERBASIS WEB PADA SMK AL-ISLAM SURAKARTA. Dalam *Journal of Information System Management (JOISM) e-ISSN* (Vol. 6, Nomor 1).

Ery Hartati. (2022). SISTEM INFORMASI TRANSAKSI GUDANG BERBASIS WEBSITE PADA CV. ASYURA. *Jurnal Ilmu Komputer*, *3*(1), 2774–6151.

Fried Sinlae, Eko Irwando, Zaky Maulana, & Vicky Eka Syahputra. (2024). *Penggunaan Framework Laravel dalam Membangun Aplikasi Website Berbasis PHP*. https://doi.org/10.38035/jsmd.v2i2

Muhamad Solahudin. (2021). Rancang Bangun Sistem Informasi Akademik Sekolah (SIAS) Berbasis Website. *Journal of Computer and Information Technology*, 2579–5317.

Nadia Fatin Harahap. (2022). *RANCANG BANGUN SISTEM INFORMASI AKADEMIK SEKOLAH BERBASIS WEB PADA SMK NEGERI 4 BANDA ACEH*.

Nistrina, K., & Sahidah, L. (2022). *UNIFIED MODELLING LANGUAGE (UML) UNTUK PERANCANGAN SISTEM INFORMASI PENERIMAAN SISWA BARU DI SMK MARGA INSAN KAMIL*.

Pratiwi, Y. A., Ginting, R. U., Situmorang, H., & Sitanggang, R. (2020). PERANCANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEB DI SMP RAHMAT ISLAMIYAH. *Jurnal Teknologi, Kesehatan dan Ilmu Sosial*, *2*.

Putri, F., Koko, A., & Yayuri, A. (2024). Rancang Bangun Sistem Informasi Inventaris menggunakan Metode Prototype pada SMK Muhamnmadiyah 3 Pekanbaru. *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI V*, *7*(1), 143–153.

Ronal, Yunita, & Yuliana. (2022). Desain Unified Modeling Language (UML) Dalam Perancangan Aplikasi Hauling Trip Di Industri Tambang Batubara. *Jurnal Teknik Informatika dan Sistem Informasi*, *9*(4). http://jurnal.mdp.ac.id

Santoso, L., & Amanullah, J. (2022). PENGEMBANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEBSITE MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT (RAD). *JURNAL ILMIAH ELEKTRONIKA DAN KOMPUTER,* *15*(2), 250–259. http://journal.stekom.ac.id/index.php/elkom€page250

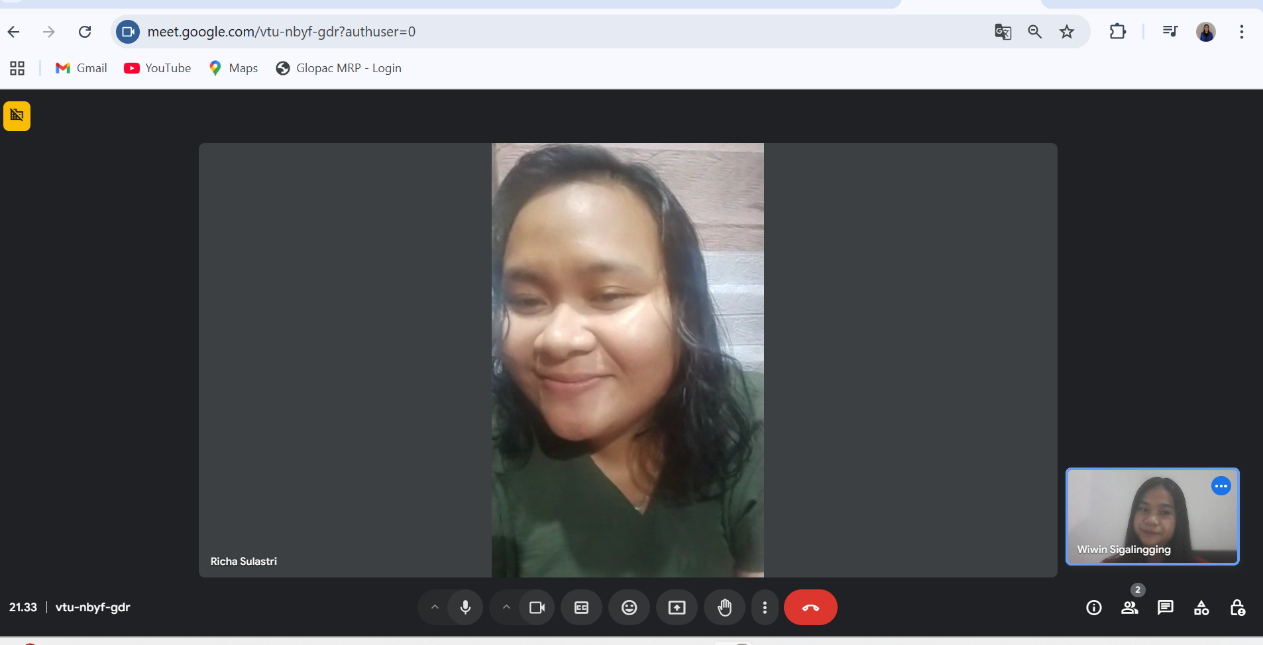
Sidharta, K., & Wibowo, T. (2020). *STUDI EFISIENSI SUMBER DAYA TERHADAP EFEKTIVITAS PENGGUNAAN DATABASE : STUDI KASUS SQL SERVER DAN MYSQL* (Vol. 1). http://journal.uib.ac.id/index.php/cbssit

Sutrisno, J., & Karnadi, V. (2021). APLIKASI PENDUKUNG PEMBELAJARAN BAHASA INGGRIS MENGGUNAKAN MEDIA LAGU BERBASIS ANDROID. *JURNAL COMASIE*, *04*(06).

Wini Muthia Kansha, Saherih, & Muchlis. (2023). Analisis Perbandingan Struktur dan Performa Framework Codeigniter dan Laravel dalam Pengembangan Web Application. *JURNAL TEKNIK INFORMATIKA STMIK ANTAR BANGSA*, 2442–3459.

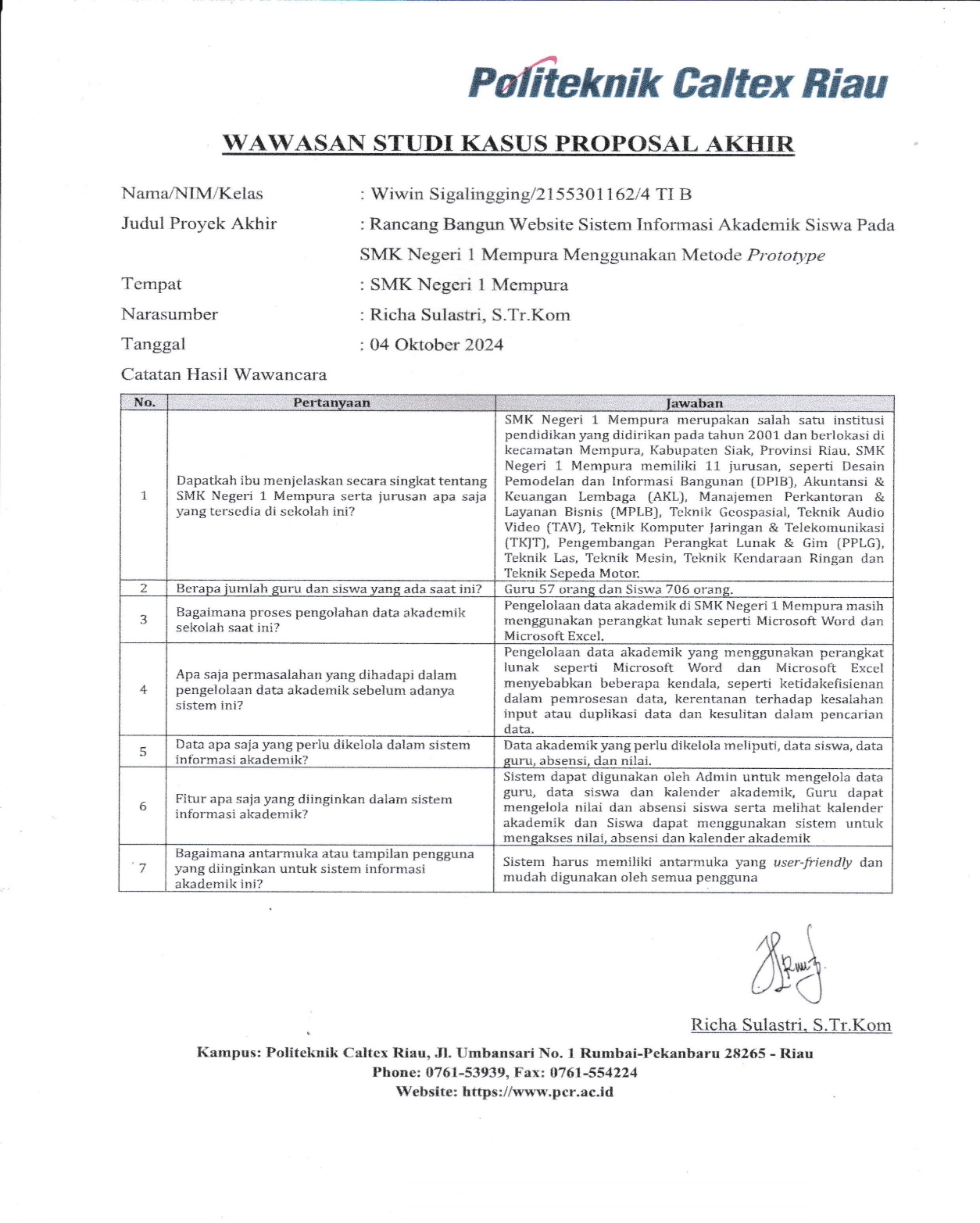
# 

# LAMPIRAN A DOKUMENTASI WAWANCARA



# LAMPIRAN B PENGUMPULAN KEBUTUHAN *USER*

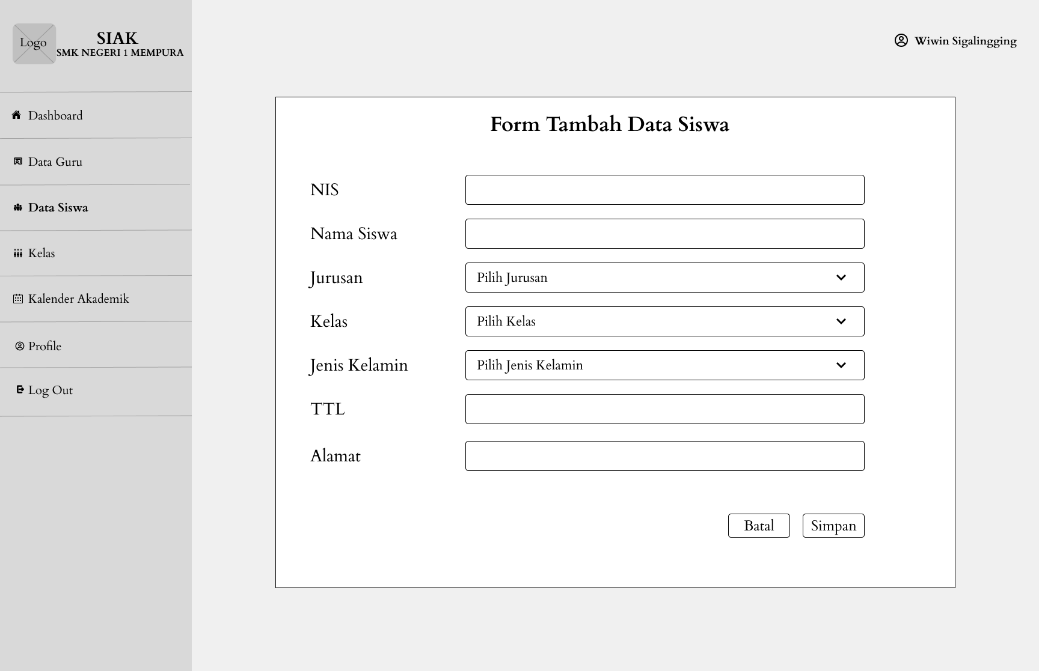
**Hasil Wawancara**

****

# LAMPIRAN C *WIREFRAME*

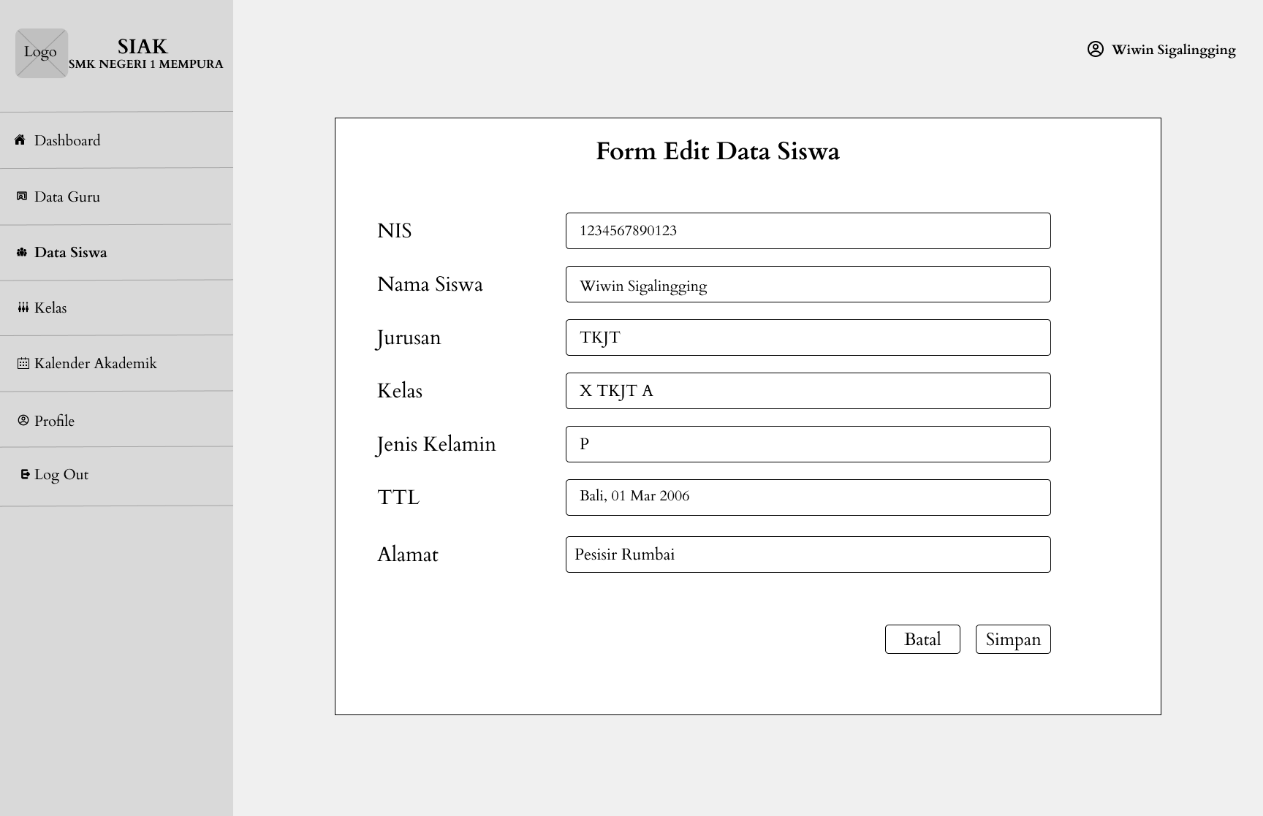
1. Antarmuka Tambah Data Siswa (Admin)

Antarmuka tambah data siswa digunakan untuk memudahkan admin dalam menambah data siswa.



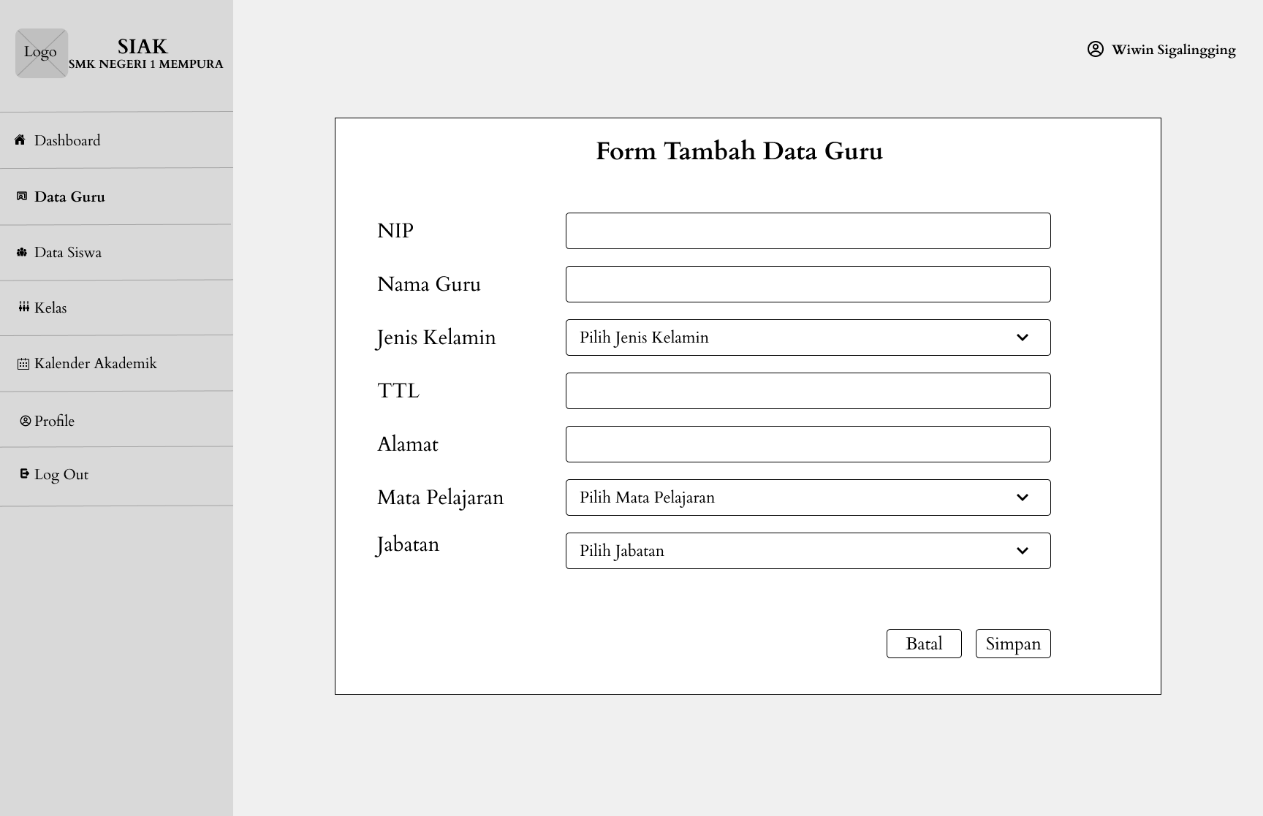
1. Antarmuka Ubah Data Siswa (Admin)

Antarmuka ubah data siswa digunakan untuk mengubah data siswa yang sudah ada.



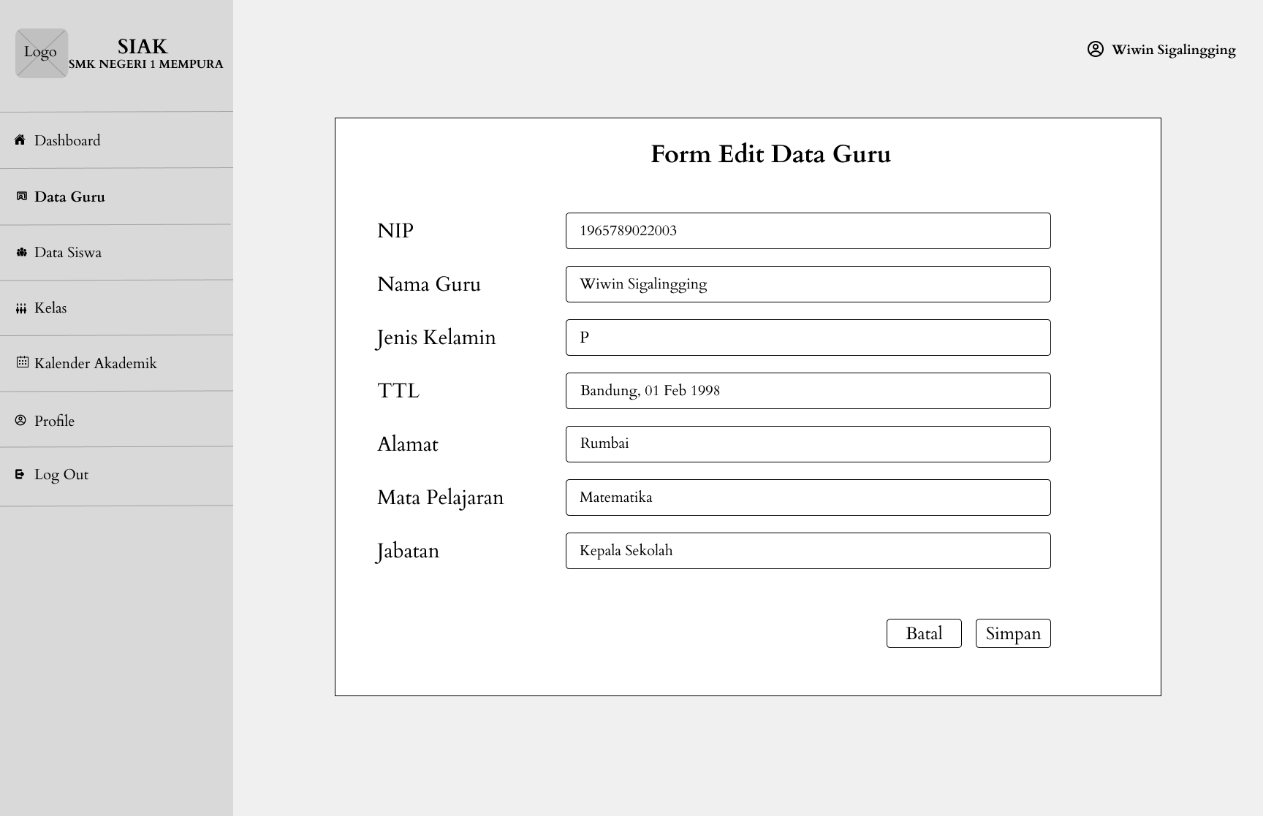
1. Antarmuka Tambah Data Guru (Admin)

Antarmuka tambah data guru digunakan untuk memudahkan admin dalam menambah data guru.



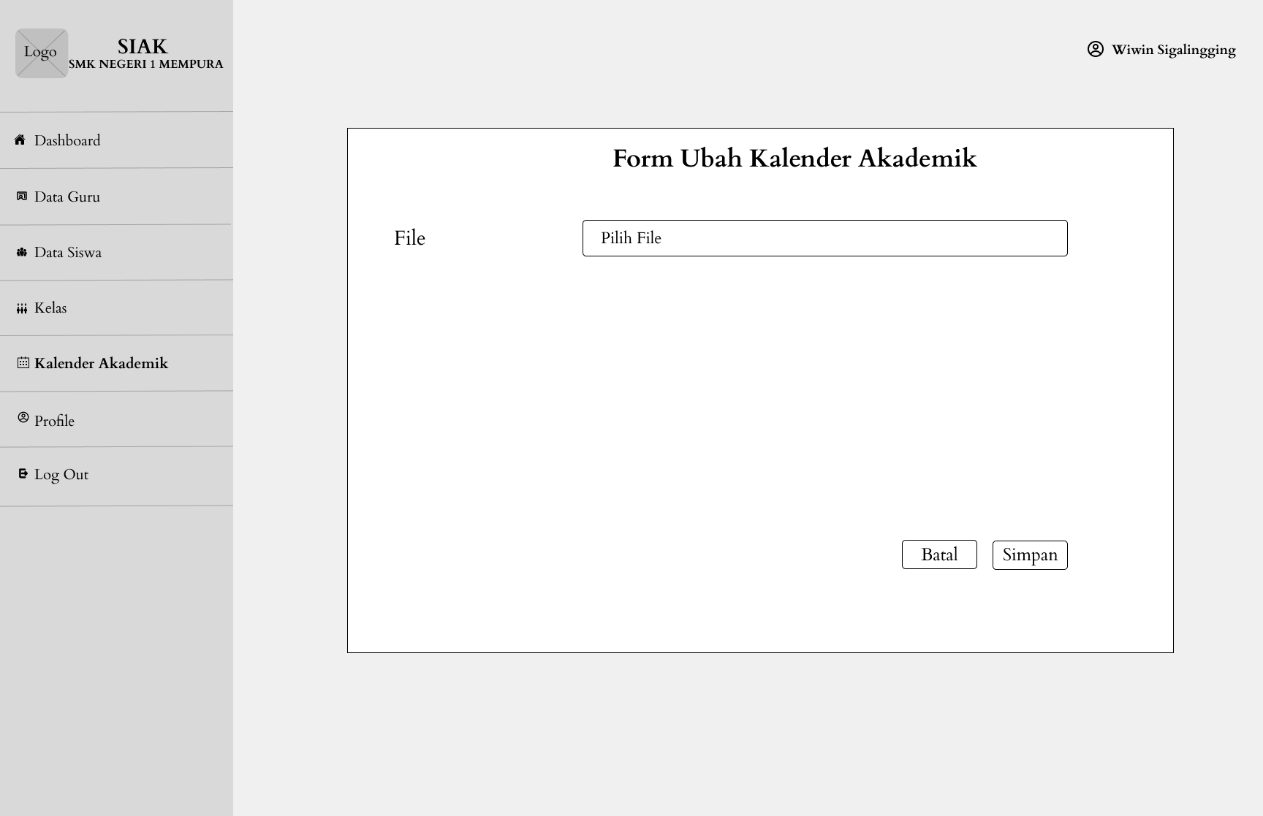
1. Antarmuka Ubah Data Guru (Admin)

Antarmuka ubah data guru digunakan untuk memudahkan admin dalam mengubah data guru yang sudah ada.



1. Antarmuka Ubah Kalender Akademik (Admin)

Antarmuka ubah kalender akademik digunakan untuk mengubah file kalender akademik yang akan digunakan.



1. Antarmuka Tambah Kelas (Admin)

Antarmuka tambah kelas digunakan admin untuk menambah data kelas.

A screenshot of a computer

Description automatically generated

1. Antarmuka Ubah Kelas (Admin)

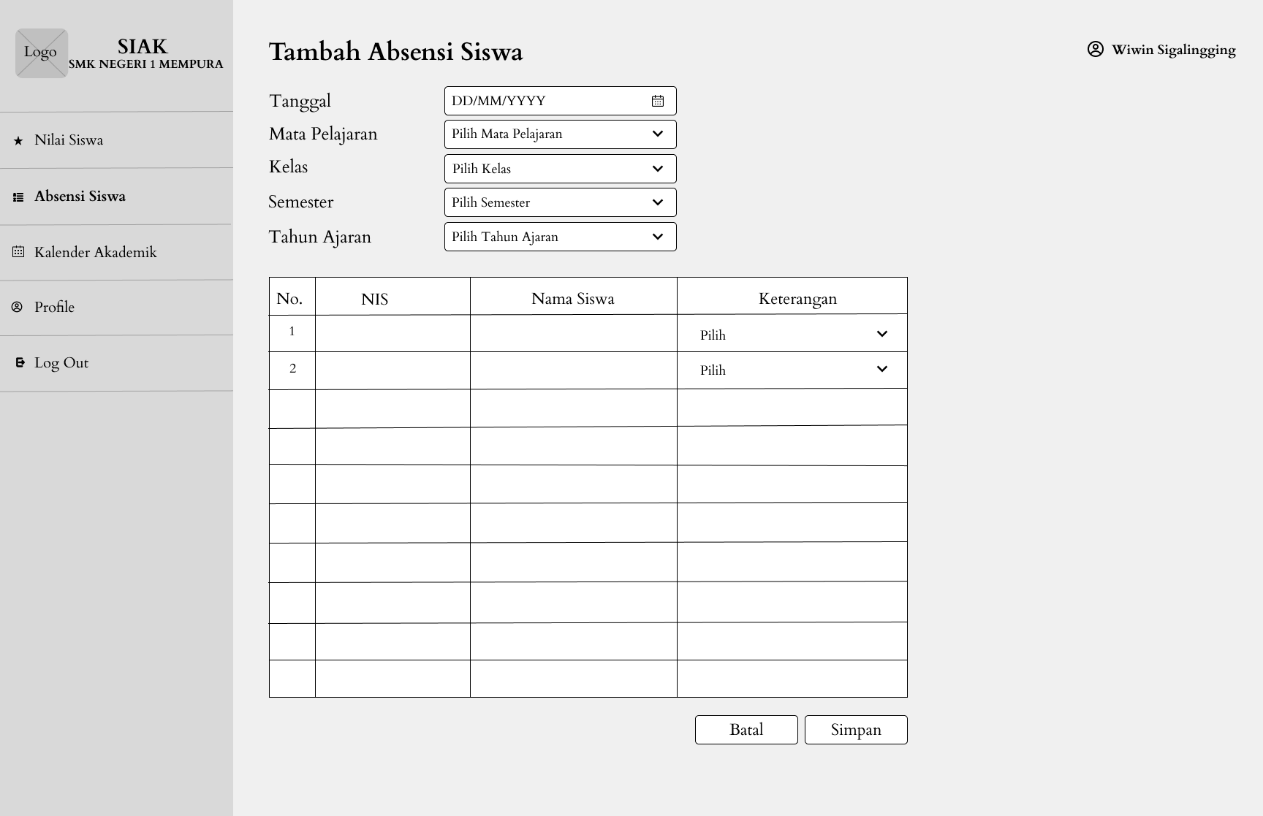
Antarmuka ubah kelas digunakan admin untuk mengubah data kelas siswa.

A screenshot of a computer

Description automatically generated

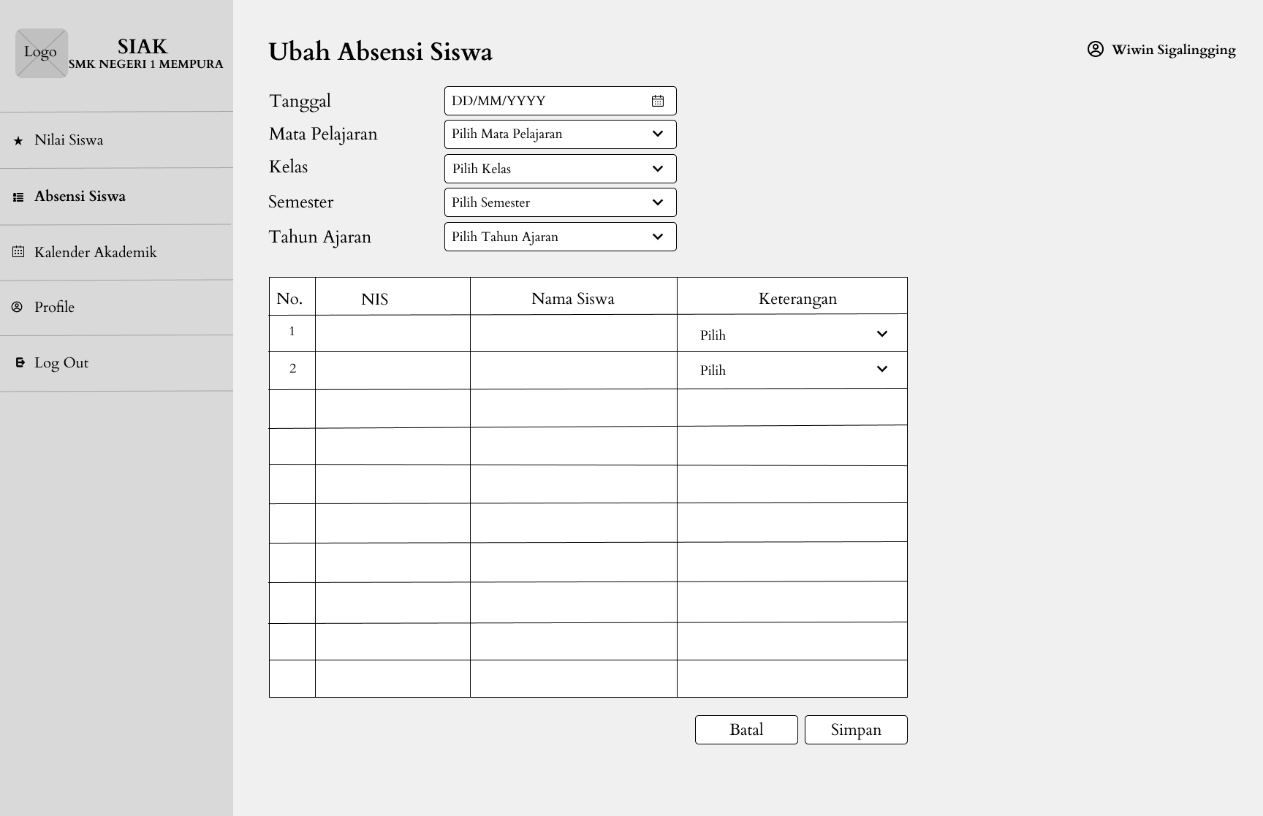
1. Antarmuka Tambah Absensi (Guru)

Antarmuka tambah absensi yang digunakan oleh guru untuk menambah absensi siswa pada setiap mata pelajaran berdasarkan tanggal, kelas, semester dan tahun ajaran.



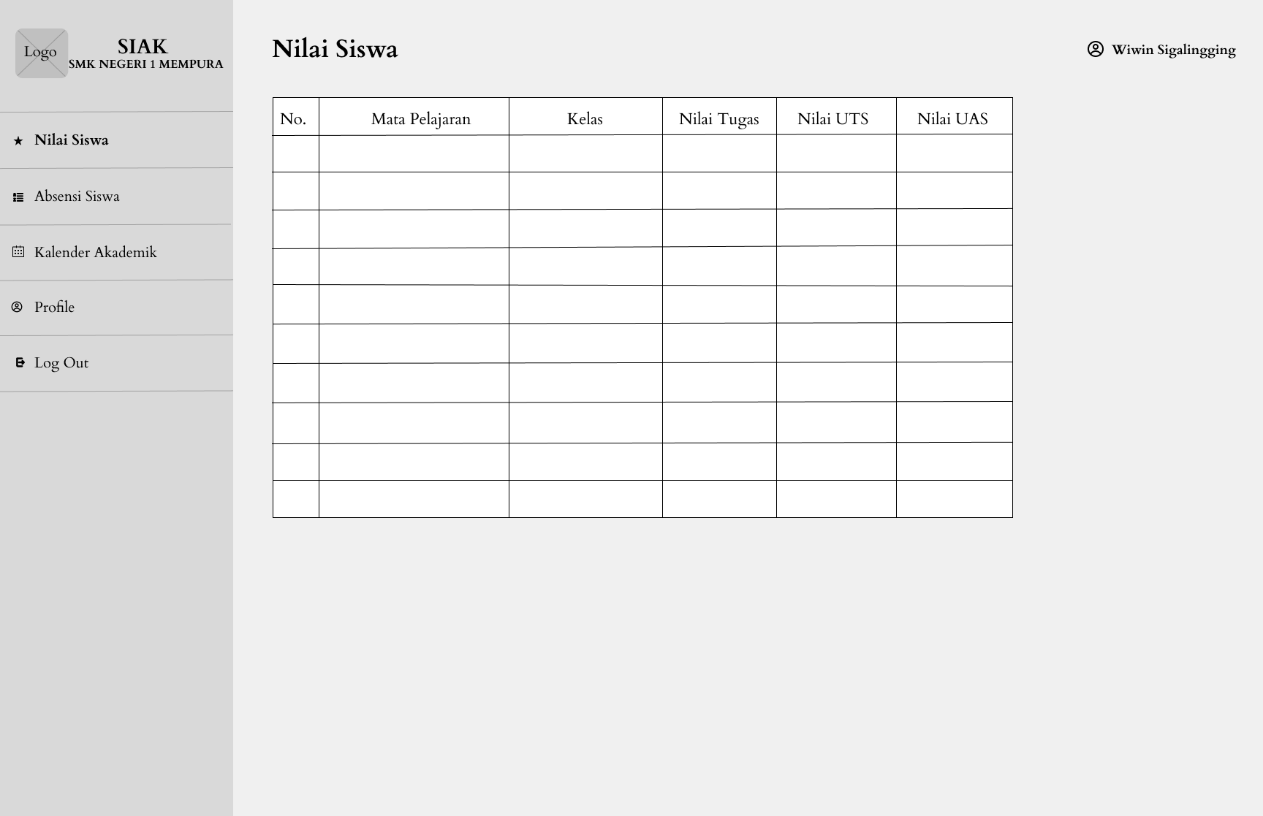
1. Antarmuka Ubah Absensi (Guru)

Antarmuka ubah absensi yang digunakan oleh guru untuk mengubah absensi siswa.



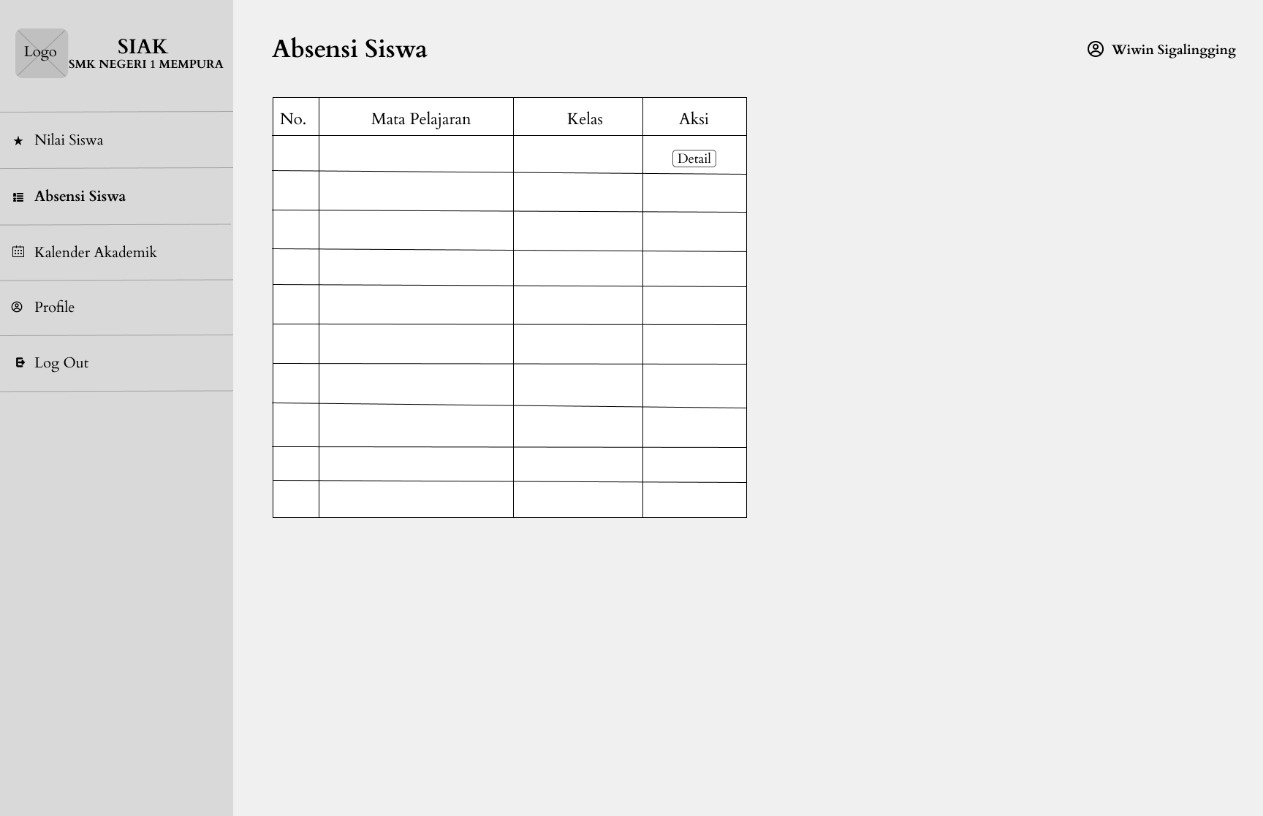
1. Antarmuka Nilai (Siswa)

Antarmuka nilai siswa merupakan tampilan ketika siswa login ke sistem. Pada antarmuka ini akan menampilkan tentang informasi nilai siswa.



1. Antarmuka Absensi (Siswa)

Antarmuka absensi siswa merupakan tampilan yang menampilkan tentang informasi absensi siswa.



1. Antarmuka *Dashboard* (Kepala Sekolah)

Antarmuka *dashboard* merupakan tampilan jika kepala sekolah berhasil login ke sistem. Tampilan ini akan menampilan informasi akademik dalam bentuk grafik.

