

RANCANG BANGUN SISTEM INFORMASI AKADEMIK SISWA SMK NEGERI 2 MAKASSAR DENGAN REACT JS

LAPORAN TUGAS AKHIR

Oleh:

Tegar Alfalakh Thamrin

105219025



**FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
UNIVERSITAS PERTAMINA
MEI 2024**

RANCANG BANGUN SISTEM INFORMASI AKADEMIK SISWA SMK NEGERI 2 MAKASSAR DENGAN REACT JS

LAPORAN TUGAS AKHIR

Oleh:

Tegar Alfalakh Thamrin

105219025



**FAKULTAS SAINS DAN ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
UNIVERSITAS PERTAMINA
MEI 2024**



LEMBAR PENGESAHAN

Judul Tugas Akhir : Rancang Bangun Sistem Informasi Akademik
Siswa SMK Negeri 2 Makassar dengan React JS

Nama Mahasiswa : Tegar Alfalakh Thamrin
Nomor Induk Mahasiswa : 105219025
Program Studi : Ilmu Komputer
Fakultas : Sains dan Ilmu Komputer
Tanggal Lulus Sidang Tugas Akhir : 17 Mei 2024

Jakarta, 17 Agustus 2024

MENGESAHKAN

Pembimbing I

Pembimbing II

Ade Irawan, Ph.D
NIP.116130

Meredita Susanty, M.Sc
NIP.116020

MENGETAHUI,

Ketua Program Studi

Ade Irawan, Ph.D

NIP. 116130

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa Tugas Akhir berjudul Rancang Bangun Sistem Informasi Akademik Siswa SMK Negeri 2 Makassar dengan React JS

ini adalah benar-benar merupakan hasil karya saya sendiri dan tidak mengandung materi yang ditulis oleh orang lain kecuali telah dikutip sebagai referensi yang sumbernya telah dituliskan secara jelas sesuai dengan kaidah penulisan karya ilmiah.

Apabila dikemudian hari ditemukan adanya kecurangan dalam karya ini, saya bersedia menerima sanksi dari Universitas Pertamina sesuai dengan peraturan yang berlaku.

Demi pengembangan ilmu pengetahuan, saya menyetujui untuk memberikan kepada Universitas Pertamina hak bebas royalti noneksklusif (*non-exclusive royalty-free right*) atas Tugas Akhir ini beserta perangkat yang ada. Dengan hak bebas royalti noneksklusif ini Universitas Pertamina berhak menyimpan, mengalih media/format-kan, 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.

Jakarta, 17 Agustus 2024

– Materai –

Yang membuat pernyataan,

Tegar Alfalakh Thamrin

ABSTRAK

Tegar Alfalakh Thamrin. 105219025. Rancang Bangun Sistem Informasi Akademik Siswa SMK Negeri 2 Makassar dengan React JS.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Kata kunci: pembelajaran mesin, komunikasi, jaringan, pemrosesan sinyal, energi

ABSTRACT

Tegar Alfalakh Thamrin. 105219025. Not an Ordinary Final Project.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Keywords: machine learning, communication, network, signal processing, energy

KATA PENGANTAR

Terima kasih sudah menggunakan template Tugas Akhir ini. Telah habis waktu sekian lamanya untuk membuat dan memperbaiki template ini. Oleh karena itu, tolong jangan hilangkan kredit pembuat dan yang memperkaya template ini, sebagai bentuk apresiasi kalian. Doakan saja semoga ini menjadi amal jariah kami semua. Aamiin.

Seperti yang sering tertulis di bagian Kata Pengantar, tidak ada gading yang tak retak, perbaikan dan saran dari pengguna template ini sangat diharapkan.

Jakarta, 17 Agustus 2045

Nama Penulis

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
BAB II TINJAUAN PUSTAKA	4
2.1 <i>Waterfall Development</i>	4
2.2 Sistem Informasi Akademik	5
2.3 Visual Studio Code	5
2.4 Javascript	5
2.5 React JS	6
2.6 JSX	6
2.7 MySQL	6
2.8 <i>Unified Modeling Language (UML)</i>	7
2.9 <i>Use-Case Diagram</i>	7
2.10 <i>Entity Relationship Diagram</i>	8
2.11 <i>Activity Diagram</i>	10
2.12 Penelitian Relevan	10
BAB III Konsep Perancangan	13
3.1 <i>Requirement Definition</i>	13

3.1.1	<i>Functional Requirement</i>	14
3.1.2	<i>Non-Functional Requirement</i>	15
3.1.3	<i>Use Case Diagram</i>	17
3.1.4	<i>Use Case Scenario</i>	18
3.1.5	<i>Activity Diagram</i>	25
3.1.6	<i>Entity Relationship Diagram</i>	35
3.2	<i>System and Software Design</i>	36
3.2.1	<i>Physical Data Model (PDM)</i>	36
3.2.2	<i>Class Diagram</i>	36
3.2.3	<i>High Fidelity</i>	37
3.3	<i>Implementation</i>	40
3.4	<i>Integration and System Testing</i>	40
3.4.1	<i>Integration and System Testing</i>	40
BAB IV	HASIL DAN PEMBAHASAN	47
4.1	<i>Communication</i>	47
4.2	<i>Planning</i>	47
4.3	<i>Modeling</i>	47
4.3.1	<i>Analysis</i>	47
4.3.2	<i>Design</i>	48
4.3.3	<i>Use Case Diagram</i>	48
4.3.4	<i>Activity Diagram</i>	48
4.3.5	<i>Entity Relationship Diagram</i>	48
4.3.6	<i>Physical Data Modeling</i>	49
4.3.7	<i>Mock-up Design</i>	49
4.3.8	<i>Login Design</i>	49
4.3.9	<i>Admin Design</i>	49
4.3.10	<i>Parent Design</i>	51
4.3.11	<i>Teacher Design</i>	51
4.4	<i>Construction</i>	51
4.4.1	<i>Coding</i>	51

BAB V KESIMPULAN DAN SARAN	55
5.1 Kesimpulan	55
5.2 Saran	55
DAFTAR PUSTAKA	56
LAMPIRAN A Judul Lampiran 1	58
1.1 Coding	58
1.2 Coba Masukkan Gambar	59

DAFTAR TABEL

Tabel 2.1	Simbol pada <i>Use-Case Diagram</i> (Dicoding, 2023)	8
Tabel 2.2	Notasi <i>Entity Relationship Diagram</i>	9
Tabel 2.3	Notasi <i>Activity Diagram</i> (Dicoding, 2021)	10
Tabel 3.1	<i>Functional Requirement</i> Sistem Informasi Akademik Siswa SMK Negeri 02 Makassar	15
Tabel 3.2	<i>Non-Functional Requirement</i> Sistem Informasi Akademik Siswa SMK Negeri 02 Makassar	16
Tabel 3.3	Keterangan <i>Use-Case Diagram</i> Sistem Informasi Akademik	17
Tabel 3.4	<i>Use Case Scenario</i> : Membuat Akun	18
Tabel 3.5	<i>Use Case Scenario</i> : Mengelola Kelas	19
Tabel 3.6	<i>Use Case Scenario</i> : Mengelola Tugas	20
Tabel 3.7	<i>Use Case Scenario</i> : Mengelola Nilai	21
Tabel 3.8	<i>Use Case Scenario</i> : Mengelola Kehadiran	22
Tabel 3.9	<i>Use Case Scenario</i> : Melihat Informasi Akademik	22
Tabel 3.10	<i>Use Case Scenario</i> : Melakukan Absensi Kehadiran	23
Tabel 3.11	<i>Use Case Scenario</i> : Mengunggah Tugas	24
Tabel 3.12	Skenario Pengujian <i>Black Box</i>	40

DAFTAR GAMBAR

Gambar 2.1	<i>Waterfall Model, (Sommerville, 2014)</i>	4
Gambar 3.1	Diagram Alir Perancangan	13
Gambar 3.2	<i>Use Case Diagram</i> Sistem Informasi Akademik Siswa	17
Gambar 3.3	Diagram Aktivitas Siswa Login	25
Gambar 3.4	Diagram Aktivitas Guru Edit Kelas	26
Gambar 3.5	Diagram Aktivitas Guru Edit Tugas	27
Gambar 3.6	Diagram Aktivitas Guru Hapus Kelas	28
Gambar 3.7	Diagram Aktivitas Guru Hapus Tugas	29
Gambar 3.8	Diagram Aktivitas Guru Tambah Kelas	30
Gambar 3.9	Diagram Aktivitas Guru Tambah Tugas	31
Gambar 3.10	Diagram Aktivitas Siswa Akses Dashboard	32
Gambar 3.11	Diagram Aktivitas Siswa Upload Tugas	33
Gambar 3.12	Diagram Aktivitas User Login	34
Gambar 3.13	<i>Entity Relational Diagram</i>	35
Gambar 3.14	<i>Physical data Model</i>	36
Gambar 3.15	<i>Class Diagram</i>	37
Gambar 3.16	Desain halaman <i>login</i>	37
Gambar 3.17	Desain halaman <i>admin</i>	38
Gambar 3.18	Desain halaman orang tua.	38
Gambar 3.19	Desain halaman guru.	39
Gambar 3.20	Desain halaman siswa.	39
Gambar 4.1	Tabel <i>Gantt Chart</i> Pelaksanaan Pengembangan (Pressman, 2010)	47
Gambar 4.2	<i>Use Case Diagram</i> Sistem Informasi Akademik Siswa	48
Gambar 4.3	Kode dalam keterangan <i>use case</i>	49
Gambar 4.4	<i>Activity Diagram Login</i>	50
Gambar 4.5	Desain halaman <i>login</i>	50
Gambar 4.6	Desain halaman <i>admin</i>	51

Gambar 4.7	Desain halaman orang tua.	52
Gambar 4.8	Desain halaman guru.	52
Gambar 4.9	Desain halaman siswa.	53
Gambar 1.1	Arsitektur Blok ResNet	59



BAB I

PENDAHULUAN

1.1 Latar Belakang

Pendidikan di Sekolah Menengah Kejuruan (SMK) merupakan salah satu tahap pendidikan yang sangat penting dalam membentuk karakter dan mengembangkan kemampuan siswa. Oleh karena itu, pemantauan terhadap perkembangan kemampuan akademik siswa menjadi sangat penting untuk meningkatkan kualitas pendidikan di SMK.

Pada zaman sekarang, teknologi informasi semakin berkembang dan memudahkan berbagai aspek kehidupan, termasuk dalam bidang pendidikan. Salah satu teknologi informasi yang dapat dimanfaatkan dalam pendidikan SMK adalah Sistem Informasi Akademik Siswa (SIAS). SIAS merupakan sebuah aplikasi berbasis website yang digunakan untuk mengelola data akademik siswa, seperti nilai, jadwal pelajaran, dan informasi pribadi.

Namun, masih banyak SMK yang belum memanfaatkan SIAS secara optimal dalam memantau perkembangan kemampuan akademik siswa bahkan masih ada beberapa SMK yang belum menerapkan SIAS di sekolahnya. Hal ini dapat menyebabkan kesulitan bagi guru dan orang tua siswa dalam memantau perkembangan siswa, sehingga sulit untuk memberikan bantuan dan pengembangan yang tepat pada siswa yang membutuhkan.

Sebuah sistem informasi akademik yang baik harus dapat memberikan informasi yang akurat dan terkini mengenai perkembangan kemampuan akademik siswa kepada guru, orang tua, dan siswa itu sendiri. Dengan informasi tersebut, guru dapat memberikan perhatian lebih pada siswa yang membutuhkan bantuan, orang tua dapat memantau kemajuan akademik anaknya, dan siswa dapat memperbaiki diri dan meningkatkan kemampuan akademiknya.

Oleh karena itu, penelitian yang membahas tentang sistem informasi akademik siswa SMK yang digunakan untuk monitoring perkembangan kemampuan akademik siswa sangat penting dilakukan. Penelitian ini dapat memberikan gambaran mengenai penggunaan SIAS dalam monitoring kemampuan akademik siswa SMK dan memberikan rekomendasi terkait pengembangan SIAS yang lebih baik lagi dalam memantau perkembangan kemampuan akademik siswa. Dengan demikian, kualitas pendidikan di SMK dapat meningkat dan siswa dapat lebih siap untuk menghadapi masa depan.

1.2 Rumusan Masalah

1. Bagaimana merancang dan mengembangkan aplikasi berbasis web untuk memantau perkembangan kemampuan akademik siswa SMK, sehingga dapat diakses oleh orang tua siswa?
2. Bagaimana memastikan keamanan dan privasi data siswa dalam sistem informasi akademik siswa berbasis web yang diakses oleh orang tua siswa?
3. Bagaimana menentukan jenis data yang harus diakses oleh orang tua siswa dalam sistem infor-

masi akademik siswa berbasis web agar dapat memonitoring perkembangan kemampuan akademik anak atau siswa?

4. Bagaimana cara menyajikan informasi tentang perkembangan kemampuan akademik siswa secara efektif dan efisien dalam sistem informasi akademik siswa berbasis web yang diakses oleh orang tua siswa?

5. Bagaimana cara mengatasi masalah teknis dan non-teknis yang mungkin muncul dalam pengembangan dan penggunaan sistem informasi akademik siswa berbasis web yang diakses oleh orang tua siswa?

1.3 Batasan Masalah

1. Aplikasi ini dirancang dan dikembangkan sebagai aplikasi web. 2. Aplikasi ini khusus dikembangkan untuk SMK Negeri 2 Makassar.

1.4 Tujuan Penelitian

1. Merancang dan mengembangkan aplikasi berbasis web untuk memantau perkembangan kemampuan akademik siswa SMK, sehingga dapat diakses oleh orang tua siswa 2. Memastikan keamanan dan privasi data siswa dalam sistem informasi akademik siswa berbasis web yang diakses oleh orang tua siswa 3. Menentukan jenis data yang harus diakses oleh orang tua siswa dalam sistem informasi akademik siswa berbasis web agar dapat memonitoring perkembangan kemampuan akademik anak atau siswa 4. Menyajikan informasi tentang perkembangan kemampuan akademik siswa secara efektif dan efisien dalam sistem informasi akademik siswa berbasis web yang diakses oleh orang tua siswa 5. Mengatasi masalah teknis dan non-teknis yang mungkin muncul dalam pengembangan dan penggunaan sistem informasi akademik siswa berbasis web yang diakses oleh orang tua siswa

1.5 Manfaat Penelitian

1. Meningkatkan kualitas pendidikan di SMK dengan memperbaiki monitoring perkembangan kemampuan akademik siswa.

2. Memudahkan orang tua siswa dalam memantau perkembangan kemampuan akademik anak atau siswa tanpa harus datang ke sekolah.

3. Meningkatkan transparansi informasi tentang kemampuan akademik siswa, sehingga orang tua siswa dapat memberikan dukungan yang lebih efektif bagi anak atau siswa.

4. Memperkuat keamanan dan privasi data siswa dalam sistem informasi akademik siswa berbasis web.

5. Menjadi acuan bagi pengembangan sistem informasi akademik siswa berbasis web di SMK lainnya untuk meningkatkan monitoring kemampuan akademik siswa.

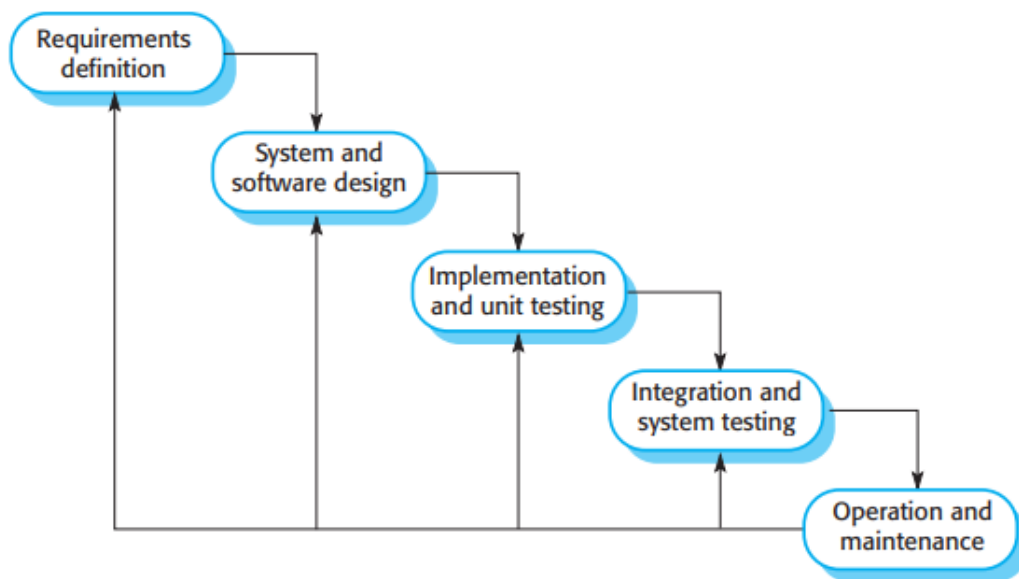
width=!,height=!,

BAB II

TINJAUAN PUSTAKA

2.1 *Waterfall Development*

Metode *Waterfall* merupakan metode pengembangan sistem secara sistematis artinya dari fase satu ke fase lainnya dilakukan secara terurut. Dalam proses implementasinya sebuah langkah harus diselesaikan terlebih dahulu mulai dari langkah pertama kemudian dilanjutkan tahap kedua dan seterusnya.



Gambar 2.1. *Waterfall Model*, (Sommerville, 2014)

1. *Requirements definition* Merupakan analisis kebutuhan perangkat lunak yang nantinya digunakan sebagai spesifikasi sistem yang akan dikembangkan proses analisis bisa dilakukan dengan beberapa cara misalnya wawancara dan kuesioner.
2. *System and software design* Data kebutuhan serta spesifikasi dari tahap sebelumnya akan dibuat menjadi *design* sistem yang akan membantu menentukan arsitektur sistem yang akan digunakan.
3. *Implementation and unit testing* Hasil *design* dari tahapan sebelumnya akan diimplementasikan sebagai suatu set program kemudian akan dilakukan testing terhadap masing-masing unit untuk memastikan apakah program sudah berjalan dengan benar.
4. *Integration and system testing* Pada tahapan semua unit program yang telah dibuat akan diintegrasikan satu sama lain menjadi suatu sistem utuh lalu kemudian dilakukan pengujian untuk memastikan sistem berjalan sesuai kebutuhan.

5. *Operation and maintenance* Pada tahapan ini aplikasi sudah mulai digunakan, kemudian juga pada tahapan ini akan dilakukan perbaikan apabila terdapat error baru yang tidak ditemukan pada proses pengembangan, selain itu dapat dilakukan pengembangan sistem baru seperti fitur dan lain-lain (Faisal, 2019).

2.2 Sistem Informasi Akademik

Sistem informasi akademik merupakan salah satu kebutuhan utama dalam suatu instansi pendidikan. Sistem ini dirancang untuk mempermudah proses dan pengelolaan data akademik, mengubahnya menjadi informasi yang bermanfaat. Sistem Informasi Akademik merupakan suatu sistem yang mengolah data dan melaksanakan proses kegiatan akademik yang melibatkan interaksi antara siswa, guru, administrasi akademik, penilaian, dan data atribut lainnya. (Wijaya, 2017) dan (Fitriana and Bakri, 2019)

2.3 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah program editor yang memungkinkan Anda menulis dan mengelola kode program. Aplikasi ini dikembangkan oleh Microsoft dan dapat digunakan pada sistem operasi Windows, Linux, dan macOS secara gratis dan sumber terbuka (Taniguchi and Nakasho, 2021). Tujuan utamanya adalah menyediakan alat yang efektif bagi pengembang perangkat lunak untuk melakukan tugas-tugas yang terkait dengan pengembangan kode.

Salah satu ciri khas yang membedakan VS Code adalah komunitas pengembang yang besar dan aktif. Komunitas ini telah menciptakan berbagai ekstensi yang dapat diintegrasikan ke dalam editor untuk menambahkan fungsionalitas tambahan sesuai kebutuhan. Sebagai contoh, ada ekstensi Divinator yang memungkinkan akses cepat ke ringkasan kode mutakhir dari dalam lingkungan pengembangan Husni et al. (2022).

2.4 Javascript

JavaScript adalah sebuah bahasa pemrograman yang memiliki penggunaan yang luas dalam pengembangan web. Lebih khusus, JavaScript digunakan sebagai bahasa scripting untuk membuat halaman web yang interaktif dan juga aplikasi web. Keunikan JavaScript terletak pada kemampuannya untuk berkolaborasi dengan HTML dan CSS dalam menciptakan antarmuka pengguna yang dinamis dan responsif. (Lotfy and Pyatt, 2018)

JavaScript memiliki sintaksis yang mirip dengan bahasa pemrograman lain seperti C++ dan Java, tetapi JavaScript juga memiliki beberapa fitur unik, seperti closures dan prototipe, yang membuatnya lebih fleksibel saat membuat perangkat lunak. JavaScript memungkinkan penggunaan berbagai jenis data, termasuk string (teks), angka, boolean (benar atau salah), dan objek (struktur data kompleks). Anda dapat menggunakan kata kunci "var", "let", atau "const" untuk mendeklarasikan variabel, masing-masing dengan tujuan yang berbeda. (Manjunath.R, 2020)

Fungsi adalah elemen kunci dalam JavaScript, memungkinkan kita untuk melakukan tugas atau perhitungan tertentu. Fungsi dapat didefinisikan menggunakan kata kunci 'function'. Mereka dapat menerima parameter, yaitu input yang diberikan pada fungsi, dan juga dapat mengembalikan nilai setelah melakukan operasi.

Salah satu kemampuan penting JavaScript adalah dalam manipulasi DOM (*Document Object Model*), yang merupakan representasi struktur halaman web dalam bentuk pohon, dan JavaScript dapat digunakan untuk memanipulasi elemen-elemen di dalam DOM. Ini memungkinkan konten dan tampilan halaman diubah secara dinamis sesuai dengan interaksi pengguna atau kejadian lainnya. Selain itu, ada banyak pustaka (*libraries*) dan kerangka kerja (*frameworks*) yang tersedia untuk JavaScript. React, Angular, dan jQuery adalah beberapa contoh yang populer. Pustaka-pustaka ini menyediakan kumpulan alat dan fungsi siap pakai yang dapat mempercepat proses pengembangan dan memberikan kemampuan tambahan kepada pengembang (Manjunath.R, 2020).

2.5 React JS

React, biasa dikenal sebagai React atau React.js, merupakan sebuah library JavaScript dengan sifat open source yang digunakan untuk mempermudah dalam membangun tampilan antarmuka pengguna (UI) menjadi lebih responsive dan interaktif. React biasa digunakan untuk mengembangkan tampilan pada Single Page Application (SPA) dan mobile application (Khuat, 2018).

React diciptakan oleh seorang insinyur perangkat lunak dari Facebook yang bernama Jordan Walke pada tahun 2011 dan secara resmi digunakan oleh Facebook pada bagian news feed mereka. Selain itu, Instagram juga memilih React untuk digunakan pada sistem mereka. Sejak saat itu, React telah berkembang menjadi salah satu library JavaScript yang banyak digunakan hingga saat ini (Satyal, 2020).

React hanya merepresentasikan bagian view dari konsep MVC (*Model-View-Controller*) (Bhalla et al., 2020). Artinya, React hanya digunakan untuk membuat tampilan antarmukanya saja tanpa memiliki kemampuan untuk berhubungan dengan basis data.

2.6 JSX

React.js memiliki aturan tersendiri dalam menuliskan kode pada setiap komponennya, yaitu dengan menggunakan JSX yang memungkinkan developer untuk menuliskan sintaks HTML di dalam JavaScript. Dengan mengkombinasikan dua hal tersebut antara HTML dan JavaScript, penulisan kode menjadi lebih sederhana dan ringkas dalam pendeskripsian tampilan antarmuka.

JSX sendiri merupakan ekstensi sintaks untuk ECMAScript tanpa ada definisi semantik tertentu. Pada dasarnya React.js bisa digunakan tanpa JSX, namun tanpa JSX penulisan kode menjadi lebih panjang dan sulit. Sedangkan dengan menggunakan JSX dapat memudahkan pengembang dalam penulisan kode dan membantu saat melakukan error debugging (Khuat, 2018).

2.7 MySQL

MySQL merupakan *software system management database (Database Management System-DBMS)* yang bersifat *Open Source* (gratis) dan sangat populer dikalangan pemrograman web, sehingga dapat digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelola datanya. Hal ini dikarenakan MySQL dapat digunakan cepat secara kinerja *query*, dan mencakupi untuk kebutuhan *database* perusahaan skala menengah kecil (Sidik, 2006).

2.8 *Unified Modeling Language (UML)*

UML merupakan bahasa visual yang digunakan untuk memodelkan dan mengkomunikasikan sistem melalui diagram dan teks pendukung (Syarif and Nugraha, 2020). Untuk menjelaskan bagaimana sebuah sistem perangkat lunak bekerja, Unified Modeling Language (UML) membantu kita melihat komponen-komponennya. UML juga dapat digunakan untuk merancang model dalam proyek data mining, yang meningkatkan pemahaman kita tentang sistem dari awal hingga penerapan.

Selain itu, UML juga berguna dalam melakukan analisis alur kendali, yang membantu kita memahami bagaimana program berorientasi objek dan terstruktur bekerja. Dengan UML, kita dapat dengan mudah melihat bagaimana objek-objek dalam program saling berinteraksi. UML juga digunakan dalam banyak metodologi rekayasa perangkat lunak, membantu kita memahami, melihat secara visual, dan menganalisis sistem dengan lebih baik (R., 2015).

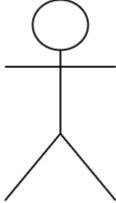
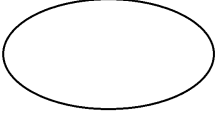



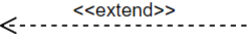
2.9 *Use-Case Diagram*

Menurut Kurniawan (2018), UC (Use Case) diagram dianggap sebagai alat yang efektif dalam menggambarkan interaksi antara pengguna (aktor) dan sistem yang sedang dikembangkan. Diagram ini memberikan gambaran yang jelas tentang konteks dari sebuah sistem dan membantu dalam menetapkan batasan sistem dengan baik.

Ada dua elemen penting yang harus dijelaskan dalam UC diagram, yaitu aktor dan UC itu sendiri. Aktor merujuk pada semua entitas yang berinteraksi langsung dengan sistem, baik itu individu (yang dinyatakan dengan peran yang dimainkan dan bukan dengan nama atau identitas pribadi) atau sistem komputer lainnya. Dalam UC diagram, aktor digambarkan sebagai gambar manusia sederhana (stick-man) dengan nama entitas yang mencerminkan peran atau sistem yang diwakilinya. Aktor dapat berperan sebagai aktor primer yang memulai sebuah UC, atau aktor sekunder yang membantu menjalankan sebuah UC (Dicoding, 2023).

Dalam UC diagram, UC sendiri diwakili sebagai simbol elips dengan kata kerja aktif yang menunjukkan pekerjaan yang dilakukan oleh aktor. Setiap aktor dapat berinteraksi dengan sistem dalam berbagai jenis UC. Sebaliknya, setiap UC dapat dijalankan oleh lebih dari satu aktor, tergantung pada interaksi yang terjadi antara sistem dan pengguna. UC diagram dianggap sebagai alat komunikasi yang kuat dan efektif untuk menggambarkan interaksi antara pengguna dan sistem. Hal ini sangat membantu dalam memahami konteks sistem yang sedang dibangun. Ini juga memperjelas ruang lingkup dan batasan sistem.

Tabel 2.1. Simbol pada *Use-Case Diagram* (Dicoding, 2023)

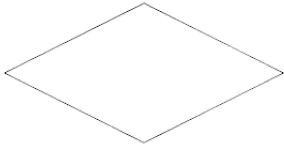
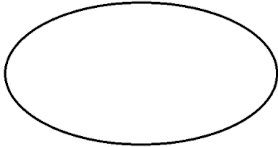
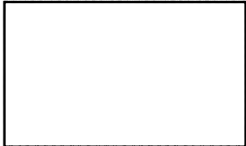

No.	Notasi	Keterangan
1	 <p>Actor</p>	Aktor berfungsi sebagai peran orang atau sistem yang turut berperan aktif dalam use case
2		Interaksi antara aktor dan sistem
3		Penghubung antara aktor dan sistem
4		Spesialisasi antara aktor dan sistem
5		Sebuah use-case yang merupakan fungsionalitas atau memerlukan use case lain untuk bekerja
6		Sebuah use-case yang bekerja jika use-case yang lain terpenuhi

2.10 Entity Relationship Diagram

ERD (*Entity Relationship Diagram*) adalah suatu gambar atau diagram yang menunjukkan bagaimana informasi dibuat, disimpan, dan digunakan dalam suatu sistem bisnis. ERD berfungsi sebagai alat

untuk memodelkan hubungan antar entitas dalam basis data relasional. Namun, jika sistem menggunakan OODBMS (*Object-Oriented Database Management System*) sebagai penyimpanan basis data, perancangan basis data tidak perlu menggunakan ERD (Pamungkas, 2020).







Tabel 2.2. Notasi *Entity Relationship Diagram*

No.	Notasi	Keterangan
1		Relasi atau hubungan antar entitas
2		Atribut
3		Himpunan entitas
4		Penghubung antar entitas

2.11 Activity Diagram

Diagram aktivitas, atau yang dikenal sebagai activity diagram dalam bahasa Inggris, adalah suatu representasi grafis yang dapat menggambarkan proses-proses yang terjadi dalam suatu sistem. Alur proses dari sistem tersebut diilustrasikan secara vertikal. Diagram aktivitas ini merupakan perkembangan dari Use Case dan mencakup rangkaian aktivitas yang terjadi (Dicoding, 2021).

Tabel 2.3. Notasi *Activity Diagram* (Dicoding, 2021)

No.	Simbol	Keterangan
1		Sebuah diagram aktivitas memiliki sebuah status awal
2		Aktivitas yang dilakukan sistem
3		Percabangan dimana ada pilihan aktivitas yang lebih dari satu
4		Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu
5		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6		Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.12 Penelitian Relevan

Dalam perancangan ini, diambil beberapa referensi dari perancangan serupa yang telah dilakukan sebelumnya dengan topik yang sama. Tujuan pengambilan beberapa referensi ini adalah untuk mengetahui apakah sistem yang akan dirancang memungkinkan atau tidak memungkinkan untuk dilakukan dengan requirement yang telah direncanakan.

Penelitian terkait yang dilakukan Lutviana, Irfan Arfianto, Taufik Fadhil Rohman, R. Bagus Bam-

bang Sumantri, dan Riska Suryani pada tahun 2023 dengan judul "Perancangan Sistem Informasi Akademik Sekolah Dasar Dengan Metode Waterfall Berbasis Website" membahas tentang perancangan Sistem Informasi Akademik Sekolah Dasar berbasis website dengan menggunakan metode Waterfall. Perancangan ini berfokus untuk meningkatkan efektivitas pengelolaan data akademik dengan mengurangi kesalahan manual melalui sistem berbasis web. Perancangan ini telah menghasilkan pembuatan ERD dan DFD, implementasi dengan PHP dan MySQL, serta pengujian menggunakan Black-box Testing. Perancangan SIAKAD yang dirancang dapat membantu sekolah dasar dalam pengelolaan data akademik dan mengurangi kesalahan sebelum penggunaan sistem. Saran untuk pengembangan lebih lanjut termasuk penambahan fitur absensi, tugas, kolom diskusi, dan lainnya.

Penelitian terkait yang dilakukan Nadya Hanifah Kamal dan Aris Gunaryati pada tahun 2023 dengan judul "Implementasi Pengembangan Web Menggunakan Teknologi MERN Stack pada Sistem Informasi Akademik Siswa Berbasis Web" membahas tentang penggunaan teknologi MERN (MongoDB, Express, React, Node) Stack untuk pengembangan web berbasis JavaScript. Pengembangan ini berfokus pada efektivitas MERN Stack dalam menghasilkan sistem web yang real-time dan efisien dan Hasil dari pengujian metode PUT yang dilakukan pada 20 user menunjukkan kinerja yang valid dan efisien. Pengembangan ini menghasilkan kesimpulan bahwa teknologi MERN Stack sangat berguna untuk pengembangan web karena sistemnya yang real-time, memudahkan, dan menghemat waktu.

Penelitian terkait yang dilakukan Tri Haryati pada tahun 2019 dengan judul "Metode Waterfall pada Sistem Informasi Akademik SMPIT Boarding School Thariq Bin Ziyad Cikarang" membahas tentang penggunaan metode waterfall pada pengembangan Sistem Informasi Akademik untuk SMPIT Boarding School Thariq Bin Ziyad Cikarang berbasis website dengan menggunakan bahasa pemrograman PHP. Pengembangan ini berfokus untuk meningkatkan efisiensi pengelolaan data akademik dengan menggantikan sistem manual berbasis excel menjadi sistem berbasis web. Pengembangan ini dapat mendukung kinerja sekolah dalam pengolahan data akademik, mempercepat proses input data, dan meningkatkan pelayanan terhadap pengguna sistem, khususnya guru dan siswa.

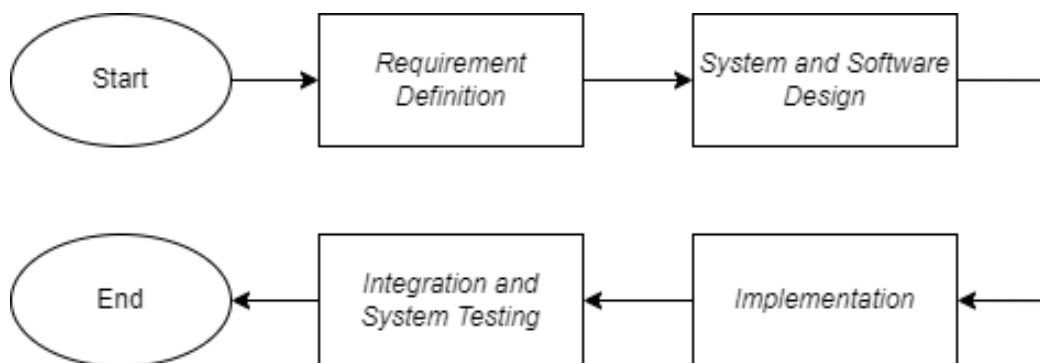
Penelitian terkait yang dilakukan Yoga Handoko Agustin, Ridwan Setiawan, Ditdit Putuwenda, dan Dendi Ramdani pada tahun 2022 dengan judul "Perancangan Sistem Informasi Akademik Berbasis Web di SMK YAPB 1 Garut" membahas tentang perancangan Sistem Informasi Akademik Berbasis Web di SMK YAPB 1 Garut dengan menggunakan metodologi Rational Unified Process dan modelan Unified Modeling Language yang dimana perancangannya mencakup data guru, siswa, kelas, jadwal, dan nilai. Pengembangan ini berfokus untuk meningkatkan efisiensi pengelolaan data akademik dan menyediakan informasi dengan cepat dan akurat dan diuji menggunakan Black Box Testing. Pengembangan ini membantu SMK YAPB 1 Garut dalam pengelolaan data akademik dan memberikan informasi yang cepat dan akurat kepada pengajar, siswa, dan orang tua/wali siswa.

width=!,height=!,

BAB III

Konsep Perancangan

Pengembangan Sistem Informasi Akademik Siswa SMK Negeri 02 Makassar menggunakan metode *waterfall* yang telah disesuaikan dengan kebutuhan, metode ini dipilih karena langkah dan tahapannya terstruktur sehingga akan lebih mudah untuk berfokus saat proses pengerjaan aplikasi, pada metode *waterfall* ketika proses *requirement* sudah selesai dilakukan maka proses pengerjaan tidak melibatkan *client* lagi, sehingga kemungkinan adanya perubahan spesifikasi aplikasi akan sangat kecil terjadi, hal ini akan sangat membantu untuk penelitian yang memiliki keterbatasan waktu dan hanya dikembangkan oleh satu orang saja. Untuk tahapan pengembangan aplikasi akan dilakukan hingga tahap *Integration and system testing*.



Gambar 3.1. Diagram Alir Perancangan

Dalam pengembangan *website* Sistem Informasi Akademik Siswa diterapkan menggunakan metode pengembangan *software Waterfall* milik Roger S. Pressman dan gambar di atas telah diolah kembali agar sesuai dengan kebutuhan. Tahapan pengembangan *software* dengan menggunakan metode ini yaitu *Communication, Planning, Modelling, Construction, dan Deployment*.

3.1 Requirement Definition

Dalam tahap *Requirement Definition*, penulis melakukan penelusuran informasi dan pengumpulan data mengenai sistem yang akan dirancang. Tujuan dari penelusuran dan pengumpulan data ini adalah untuk merencanakan fitur-fitur apa saja yang akan diimplementasikan dalam aplikasi yang sedang dirancang.

Untuk mengumpulkan informasi ini, penulis melakukan pengamatan terhadap aplikasi serupa, seperti Sistem Informasi Akademik Siswa CERIA. Selain itu, penulis juga melakukan wawancara dengan narasumber dari SMK Negeri 2 Makassar, khususnya dengan Kepala Jurusan Teknik Komputer dan Jaringan.

Wawancara dilakukan secara tatap muka melalui konferensi video menggunakan aplikasi Zoom. Pertanyaan yang diajukan telah ditetapkan sebelumnya oleh tim pengembang, dan narasumber juga

memberikan beberapa pertanyaan tambahan agar penulis dapat memenuhi permintaan yang diinginkan oleh pihak sekolah yang terkait.

Dari hasil wawancara didapatkan beberapa hal penting yaitu :

1. Pembuatan sistem informasi akademik ini dapat mempermudah orang tua dalam mengakses informasi akademik anak atau siswa.
2. Admin dapat membuat pengguna baru dengan role guru, orang tua, dan siswa.
3. Guru dapat membuat kelas dengan detail seperti nama kelas, deskripsi kelas, tahun aktif, dan status kelas.
4. Guru dapat membuat tugas pada kelas dengan detail seperti nama tugas, deskripsi, dan tenggat waktu.
5. Guru dapat membuat kehadiran pada kelas.
6. Orang tua dapat menampilkan dashboard yang berisi daftar kelas, daftar tugas, dan kehadiran siswa.
7. Siswa dapat menampilkan dashboard yang berisi daftar kelas, daftar tugas, dan kehadiran.
8. Siswa dapat melakukan absensi.
9. Siswa dapat melakukan *upload* tugas.

3.1.1 *Functional Requirement*

langkah selanjutnya yaitu menyusun *functional requirement* dari hasil wawancara yang sudah dilakukan dengan kepala jurusan teknik komputer jaringan SMK Negeri 02 Makassar untuk sistem yang akan dibangun.

Tabel 3.1. *Functional Requirement* Sistem Informasi Akademik Siswa SMK Negeri 02 Makassar

ID	Keterangan
FR-01	Sistem mampu mengelola data pengguna yang terdiri dari nama, nomor induk, email, nomor telepon, jenis kelamin, dan alamat.
FR-02	Sistem mampu mengelola data akademik yang terdiri dari nilai, jadwal, kelas, dan tugas.
FR-03	Sistem mampu membuat pengguna baru oleh admin SIAS berdasarkan peran (Admin, Guru, Siswa, Orang Tua).
FR-04	Sistem mampu menampilkan dashboard khusus yang menampilkan informasi relevan sesuai peran pengguna yang login.
FR-05	Sistem mampu membuat kelas dan menambahkan siswa ke dalam kelas.
FR-06	Sistem mampu membuat tugas, memfasilitasi siswa untuk mengumpulkan tugas, dan memfasilitasi guru untuk memberikan nilai.
FR-07	Sistem mampu mencatat dan mengelola data kehadiran siswa per pertemuan atau mata pelajaran.
FR-08	Sistem mampu mengirimkan notifikasi kepada pengguna terkait pembaruan penting (contoh: tugas baru, nilai masuk, pengumuman).

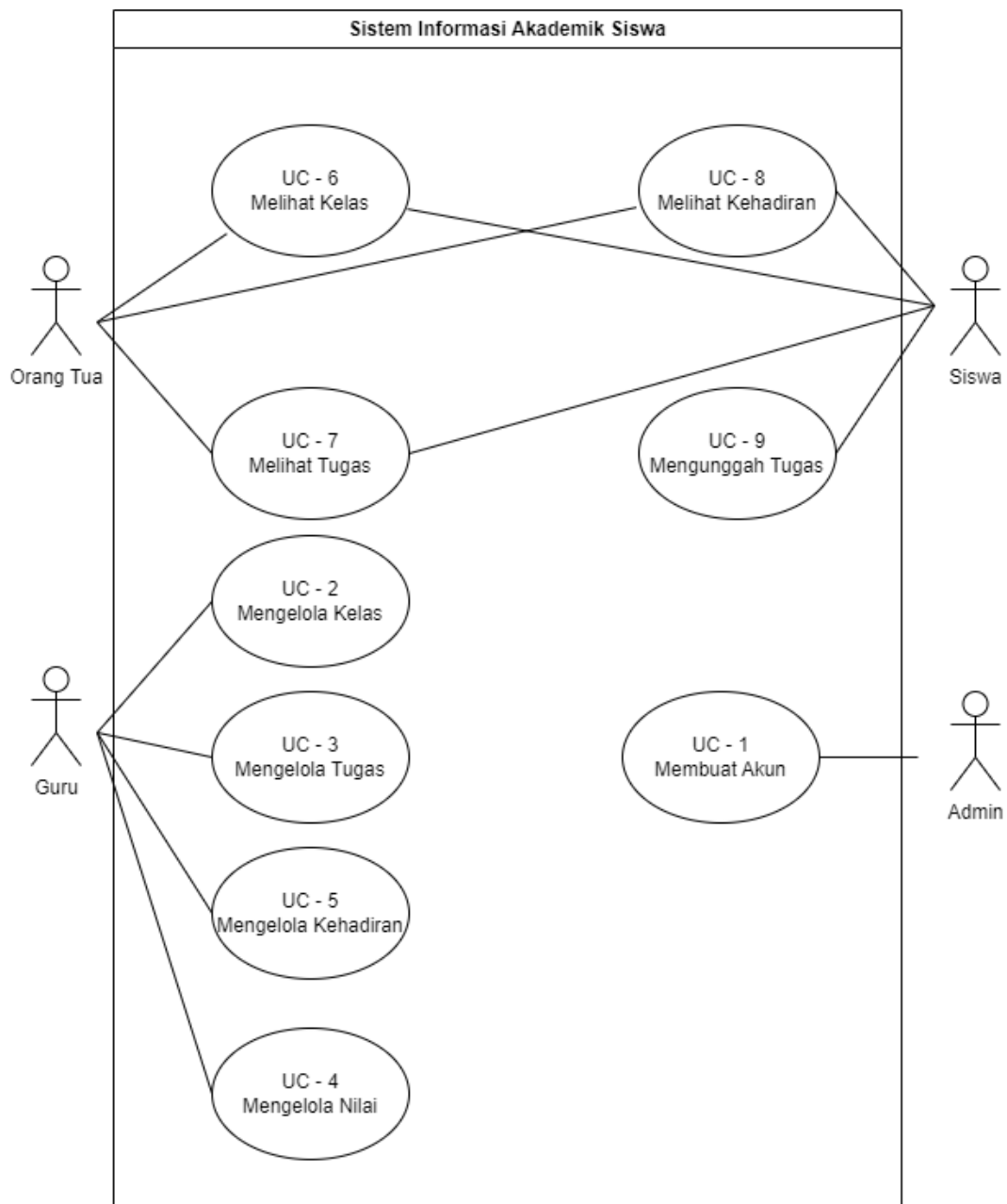
3.1.2 *Non-Functional Requirement*

Kebutuhan non-fungsional mendefinisikan atribut kualitas dan batasan operasional dari sistem yang akan dibangun.

Tabel 3.2. *Non-Functional Requirement* Sistem Informasi Akademik Siswa SMK Negeri 02 Makassar

ID	Kategori	Keterangan
NFR-01	<i>Usability</i>	Sistem dilengkapi antarmuka pengguna yang responsif, yang mampu menyesuaikan tampilan sesuai dengan perangkat yang digunakan (smartphone, tablet, komputer) untuk mengakses sistem informasi akademik siswa.
NFR-02	<i>Usability</i>	Sistem harus mudah dipahami dan digunakan oleh semua peran pengguna (Admin, Guru, Siswa, Orang Tua) dengan navigasi yang intuitif.
NFR-03	<i>Performance</i>	Waktu muat halaman utama (dashboard) setelah pengguna login tidak boleh melebihi 3 detik pada koneksi internet standar.
NFR-04	<i>Security</i>	Sistem harus melindungi data pribadi dan akademik siswa dari akses tidak sah melalui mekanisme otentikasi dan otorisasi berbasis peran.
NFR-05	<i>Security</i>	Semua transmisi data sensitif, terutama kata sandi, harus dienkripsi saat transit (HTTPS) dan saat disimpan di database (hashing).
NFR-06	<i>Reliability</i>	Sistem harus memiliki tingkat ketersediaan (uptime) minimal 99% selama jam operasional sekolah.

3.1.3 Use Case Diagram



Gambar 3.2. Use Case Diagram Sistem Informasi Akademik Siswa

Tabel 3.3. Keterangan Use-Case Diagram Sistem Informasi Akademik

ID.	Use Case	Keterangan
UC-1	Membuat Akun	Admin dapat menambah akun pengguna baru
UC-2	Mengelola Kelas	Guru dapat menambah, menghapus, dan memperbaharui kelas.
UC-3	Mengelola Tugas	Guru dapat menambah, menghapus, dan memperbaharui tugas.

ID.	Use Case	Keterangan
UC-4	Mengelola Nilai	Guru dapat menambah, menghapus, dan memperbaharui nilai.
UC-5	Mengelola Kehadiran	Guru dapat menambah, menghapus, dan memperbaharui kehadiran.
UC-6	Melihat Kelas	Orang tua dan siswa dapat melihat daftar kelas.
UC-7	Melihat Tugas	Orang tua dan siswa dapat melihat daftar tugas.
UC-8 A	Melihat Kehadiran	Orang tua dan siswa dapat melihat kehadiran.
UC-8 B	Melihat Kehadiran	Siswa dapat melakukan absensi kehadiran.
UC-9	Mengunggah Tugas	Siswa dapat mengunggah tugas.

3.1.4 Use Case Scenario

Tabel 3.4. Use Case Scenario: Membuat Akun

Use Case ID	UC-1
Nama Use Case	Membuat Akun Pengguna
Aktor	Admin
Deskripsi	Admin membuat akun pengguna baru (Guru, Siswa, atau Orang Tua) melalui panel administrasi.
Pre-kondisi	Admin telah login ke dalam sistem.
Skenario Normal - Membuat Akun Baru	
Aksi Aktor	Respons Sistem
1. Memilih menu "Manajemen Pengguna".	2. Menampilkan halaman daftar pengguna.
3. Menekan tombol "Tambah Pengguna Baru".	4. Menampilkan form registrasi pengguna.
5. Mengisi data form dengan data valid dan menekan "Simpan".	6. Memvalidasi data, menyimpan ke database, dan menampilkan notifikasi "Pengguna baru berhasil ditambahkan".
Kondisi Akhir: Pengguna baru berhasil dibuat dan tampil dalam daftar.	
Skenario Tidak Normal - Email Sudah Terdaftar	
Aksi Aktor	Respons Sistem
1. Mengikuti langkah 1-4 pada Skenario Normal.	

2. Mengisi form dengan email yang sudah ada di database, lalu menekan "Simpan".	3. Sistem menolak penyimpanan data dan menampilkan pesan error: "Email sudah terdaftar."
<i>Kondisi Akhir: Akun pengguna baru gagal dibuat.</i>	

Tabel 3.5. *Use Case Scenario: Mengelola Kelas*

<i>Use Case ID</i>	UC-2
<i>Nama Use Case</i>	Mengelola Kelas
<i>Aktor</i>	Guru
<i>Deskripsi</i>	Guru dapat membuat kelas baru, mengubah detail kelas, dan menghapus kelas yang sudah tidak aktif.
<i>Pre-kondisi</i>	Guru telah login ke dalam sistem.
Skenario Menambah Kelas	
Aksi Aktor (Guru)	Respons Sistem
1. Memilih menu "Kelas".	2. Menampilkan daftar kelas yang diajar.
3. Menekan tombol "Tambah Kelas".	4. Menampilkan form untuk membuat kelas baru.
5. Mengisi detail kelas dan menekan "Simpan".	6. Menyimpan data kelas baru ke database dan menampilkan notifikasi "Kelas berhasil dibuat".
<i>Kondisi Akhir: Data kelas baru berhasil disimpan.</i>	
Skenario Mengubah Kelas	
Aksi Aktor (Guru)	Respons Sistem
1. Memilih salah satu kelas dari daftar, lalu menekan tombol "Edit".	2. Menampilkan form dengan data kelas yang sudah ada.
3. Mengubah informasi kelas dan menekan "Simpan Perubahan".	4. Memperbarui data kelas di database dan menampilkan notifikasi "Kelas berhasil diperbarui".
<i>Kondisi Akhir: Data kelas berhasil diperbarui.</i>	
Skenario Menghapus Kelas	
Aksi Aktor (Guru)	Respons Sistem
1. Memilih salah satu kelas dari daftar, lalu menekan tombol "Hapus".	2. Menampilkan jendela konfirmasi "Apakah Anda yakin ingin menghapus kelas ini?".

3. Menekan tombol "Ya, Hapus".	4. Menghapus data kelas dari database dan menampilkan notifikasi "Kelas berhasil dihapus".
<i>Kondisi Akhir: Data kelas berhasil dihapus.</i>	
Skenario Tidak Normal - Gagal Menghapus	
Aksi Aktor (Guru)	Respons Sistem
1. Guru menekan "Ya, Hapus" pada kelas yang masih memiliki siswa aktif.	2. Sistem menolak penghapusan dan menampilkan pesan error: "Kelas tidak dapat dihapus karena masih ada data siswa terkait."
<i>Kondisi Akhir: Data kelas tidak terhapus.</i>	

Tabel 3.6. *Use Case Scenario*: Mengelola Tugas

<i>Use Case ID</i>	UC-3
<i>Nama Use Case</i>	Mengelola Tugas
<i>Aktor</i>	Guru
<i>Deskripsi</i>	Guru dapat membuat, mengubah, dan menghapus tugas pada kelas yang diajar.
<i>Pre-kondisi</i>	Guru sudah login dan berada di halaman detail kelas.
Skenario Menambah Tugas	
Aksi Aktor (Guru)	Respons Sistem
1. Menekan tombol "Tambah Tugas".	2. Menampilkan form untuk menambah tugas baru.
3. Mengisi form (judul, deskripsi, tenggat waktu) dan menekan "Simpan".	4. Memvalidasi data, menyimpan ke database, dan menampilkan notifikasi "Tugas berhasil dibuat".
<i>Kondisi Akhir: Data tugas baru berhasil disimpan.</i>	
Skenario Mengubah Tugas	
Aksi Aktor (Guru)	Respons Sistem
1. Memilih tugas dari daftar dan menekan tombol "Edit".	2. Menampilkan form edit dengan data tugas yang sudah ada.
2. Mengubah data pada form dan menekan "Simpan Perubahan".	3. Memvalidasi data, memperbarui di database, dan menampilkan notifikasi "Tugas berhasil diperbarui".
<i>Kondisi Akhir: Data tugas berhasil diperbarui.</i>	
Skenario Menghapus Tugas	
Aksi Aktor (Guru)	Respons Sistem

1. Memilih tugas dari daftar dan menekan tombol "Hapus".	2. Menampilkan jendela konfirmasi penghapusan.
3. Menekan tombol konfirmasi "Ya, Hapus".	4. Menghapus data tugas dari database dan menampilkan notifikasi "Tugas berhasil dihapus".
<i>Kondisi Akhir: Data tugas berhasil dihapus.</i>	

Tabel 3.7. *Use Case Scenario*: Mengelola Nilai

<i>Use Case ID</i>	UC-4
<i>Nama Use Case</i>	Mengelola Nilai
<i>Aktor</i>	Guru
<i>Deskripsi</i>	Guru memasukkan dan mengubah nilai siswa untuk tugas tertentu.
<i>Pre-kondisi</i>	Guru sudah login, dan data kelas, siswa, serta tugas telah dibuat.
Skenario Normal - Memberi Nilai Siswa	
Aksi Aktor (Guru)	Respons Sistem
1. Memilih kelas, lalu memilih menu "Tugas" atau "Nilai".	2. Menampilkan daftar tugas yang ada di kelas tersebut.
3. Memilih salah satu tugas untuk dinilai.	4. Menampilkan daftar siswa yang telah mengumpulkan tugas beserta kolom untuk input nilai.
5. Memasukkan nilai untuk seorang siswa.	6. (Sistem melakukan validasi otomatis, misal nilai harus antara 0-100).
7. Menekan tombol "Simpan Nilai".	8. Menyimpan data nilai ke database dan menampilkan notifikasi "Nilai berhasil disimpan".
<i>Kondisi Akhir: Data nilai siswa berhasil disimpan.</i>	
Skenario Tidak Normal - Input Nilai Tidak Valid	
Aksi Aktor (Guru)	Respons Sistem
1. Mengikuti langkah 1-4 pada Skenario Normal.	
2. Memasukkan nilai di luar rentang (misal: 110 atau "A").	3. Sistem menampilkan pesan error "Input nilai tidak valid. Harap masukkan angka antara 0-100."
<i>Kondisi Akhir: Data nilai tidak tersimpan.</i>	

Tabel 3.8. *Use Case Scenario*: Mengelola Kehadiran

<i>Use Case ID</i>	UC-5
<i>Nama Use Case</i>	Mengelola Data Kehadiran Siswa
<i>Aktor</i>	Guru
<i>Deskripsi</i>	Guru mencatat status kehadiran (Hadir, Izin, Sakit, Alfa) untuk setiap siswa pada pertemuan tertentu.
<i>Pre-kondisi</i>	1. Guru telah login.
	2. Kelas dan daftar siswa di dalamnya telah terdefinisi.
Skenario Normal - Mencatat Kehadiran	
Aksi Aktor (Guru)	Respons Sistem
1. Memilih kelas, lalu memilih menu "Kehadiran".	2. Menampilkan daftar pertemuan atau tanggal untuk pencatatan kehadiran.
3. Menekan tombol "Buat Sesi Kehadiran Baru".	4. Menampilkan daftar siswa di kelas tersebut dengan pilihan status kehadiran (default: Hadir).
5. Mengubah status untuk siswa yang tidak hadir (misal: Izin, Sakit, Alfa).	6. Sistem secara visual membedakan status.
7. Menekan tombol "Simpan Kehadiran".	8. Menyimpan data kehadiran semua siswa untuk sesi tersebut ke database dan menampilkan notifikasi "Data kehadiran berhasil disimpan".
<i>Kondisi Akhir: Data kehadiran berhasil disimpan.</i>	

Tabel 3.9. *Use Case Scenario*: Melihat Informasi Akademik

<i>Use Case ID</i>	UC-6, UC-7, UC-8A
<i>Nama Use Case</i>	Melihat Informasi Kelas, Tugas, dan Kehadiran
<i>Aktor</i>	Siswa, Orang Tua
<i>Deskripsi</i>	Siswa atau Orang Tua mengakses dan melihat informasi detail mengenai kelas, daftar tugas, dan rekap kehadiran.
<i>Pre-kondisi</i>	Aktor telah login ke dalam sistem.
Skenario Normal	
Aksi Aktor (Siswa/Orang Tua)	Respons Sistem
1. Memilih menu "Kelas" dari dashboard.	2. Menampilkan daftar semua kelas yang diikuti.

3. Mengklik salah satu nama kelas.	4. Menampilkan halaman detail kelas, yang berisi deskripsi dan menu/tab lain (Tugas, Kehadiran).
4. Memilih tab "Tugas".	5. Menampilkan daftar semua tugas untuk kelas tersebut, beserta tenggat waktu dan status.
5. Memilih tab "Kehadiran".	6. Menampilkan rekapitulasi kehadiran siswa (persentase dan daftar per pertemuan).
<i>Kondisi Akhir: Aktor berhasil melihat informasi akademik yang relevan.</i>	

Tabel 3.10. *Use Case Scenario*: Melakukan Absensi Kehadiran

<i>Use Case ID</i>	UC-8B
<i>Nama Use Case</i>	Melakukan Absensi Mandiri
<i>Aktor</i>	Siswa
<i>Deskripsi</i>	Siswa mencatatkan kehadirannya secara mandiri untuk sesi pertemuan yang sedang berlangsung.
<i>Pre-kondisi</i>	0. Siswa telah login. 1. Guru telah memulai sesi absensi untuk pertemuan saat ini. 2. Sesi absensi masih dalam periode waktu yang aktif.
Skenario Normal	
Aksi Aktor (Siswa)	Respons Sistem
1. Membuka halaman dashboard atau halaman kelas.	2. Menampilkan notifikasi atau tombol "Lakukan Absensi Sekarang".
3. Menekan tombol "Lakukan Absensi Sekarang".	4. Mencatat status siswa sebagai "Hadir" untuk sesi tersebut.
	5. Menampilkan pesan konfirmasi "Absensi berhasil direkam pada [waktu]". Tombol absensi menghilang.
<i>Kondisi Akhir: Status kehadiran siswa berhasil tercatat.</i>	
Skenario Tidak Normal - Sesi Sudah Berakhir	
Aksi Aktor (Siswa)	Respons Sistem
1. Siswa mencoba mengakses fitur absensi setelah sesi ditutup oleh guru.	2. Sistem menolak aksi dan menampilkan pesan: "Sesi absensi untuk pertemuan ini sudah berakhir."
<i>Kondisi Akhir: Absensi gagal, status kehadiran siswa tidak berubah.</i>	

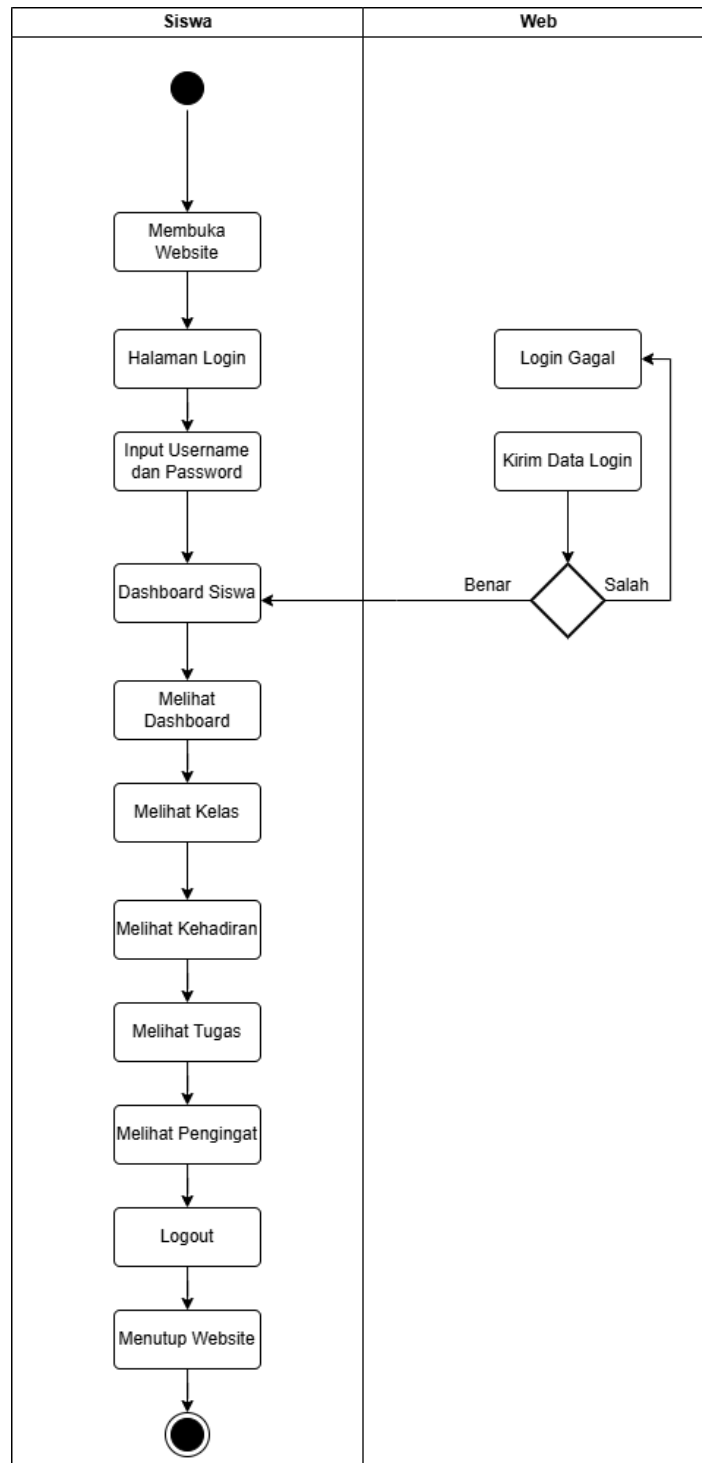
Tabel 3.11. *Use Case Scenario*: Mengunggah Tugas

<i>Use Case ID</i>	UC-9
<i>Nama Use Case</i>	Mengunggah File Tugas
<i>Aktor</i>	Siswa
<i>Deskripsi</i>	Siswa mengirimkan file sebagai jawaban atau hasil pengerjaan tugas.
<i>Pre-kondisi</i>	1. Siswa telah login.
	2. Tenggat waktu pengumpulan tugas belum terlewat.
Skenario Normal - Mengunggah Tugas Tepat Waktu	
Aksi Aktor (Siswa)	Respons Sistem
1. Memilih kelas, lalu memilih tugas yang akan dikerjakan.	2. Menampilkan halaman detail tugas dan tombol "Unggah Tugas".
3. Menekan tombol "Unggah Tugas".	4. Membuka jendela dialog untuk memilih file.
5. Memilih file yang valid (misal: .pdf) dan menekan "Open".	6. Menampilkan nama file yang dipilih pada antarmuka.
7. Menekan tombol "Kirim".	8. Mengunggah file ke server, menyimpan referensi file, memperbarui status tugas menjadi "Terkumpul", dan menampilkan notifikasi "Tugas berhasil diunggah".
<i>Kondisi Akhir: Tugas berhasil diunggah.</i>	
Skenario Tidak Normal - Ukuran File Terlalu Besar	
Aksi Aktor (Siswa)	Respons Sistem
1. Mengikuti langkah 1-5 pada Skenario Normal, namun memilih file yang ukurannya melebihi batas.	
2. Menekan tombol "Kirim".	3. Sistem menolak unggahan dan menampilkan pesan error: "Ukuran file melebihi batas maksimal (misal: 8 MB)."
<i>Kondisi Akhir: Unggah gagal, status tugas tidak berubah.</i>	
Skenario Tidak Normal - Melebihi Tenggat Waktu	
Aksi Aktor (Siswa)	Respons Sistem
1. Mencoba mengakses halaman unggah tugas setelah tenggat waktu berakhir.	2. Sistem menampilkan pesan "Tenggat waktu untuk tugas ini telah berakhir" dan menonaktifkan tombol unggah.

Kondisi Akhir: Siswa tidak dapat mengunggah tugas.

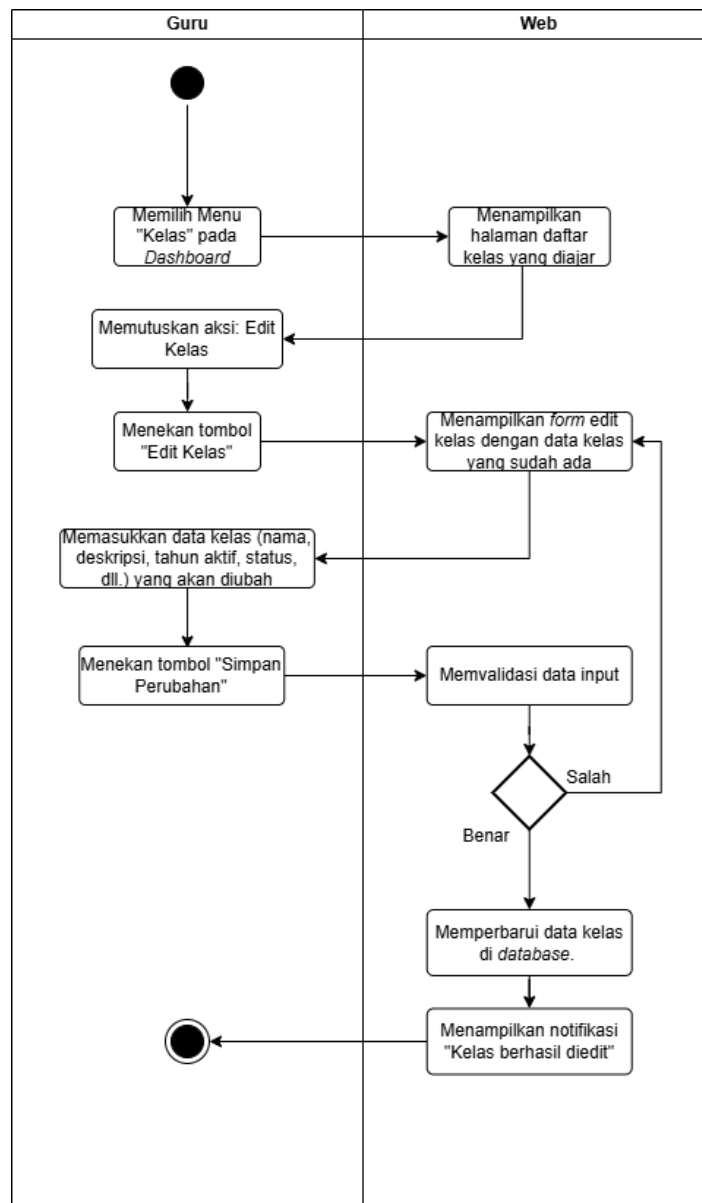
3.1.5 Activity Diagram

Activity Siswa Login



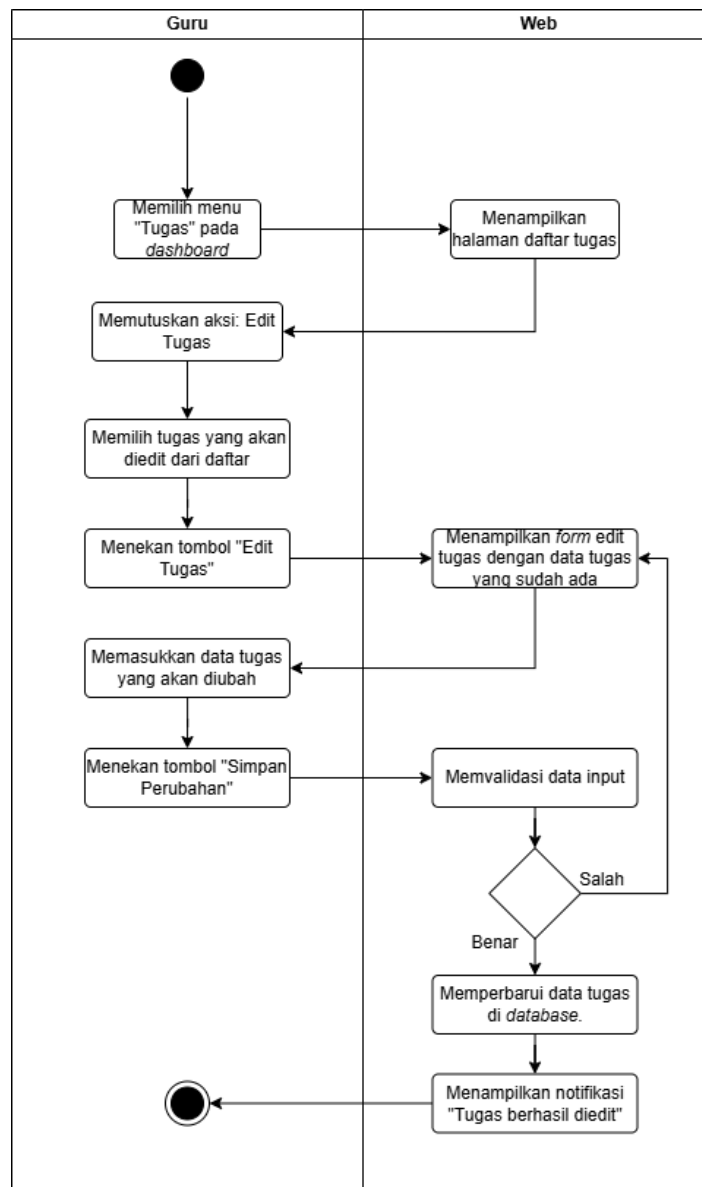
Gambar 3.3. Diagram Aktivitas Siswa Login

Activity Guru Edit Kelas



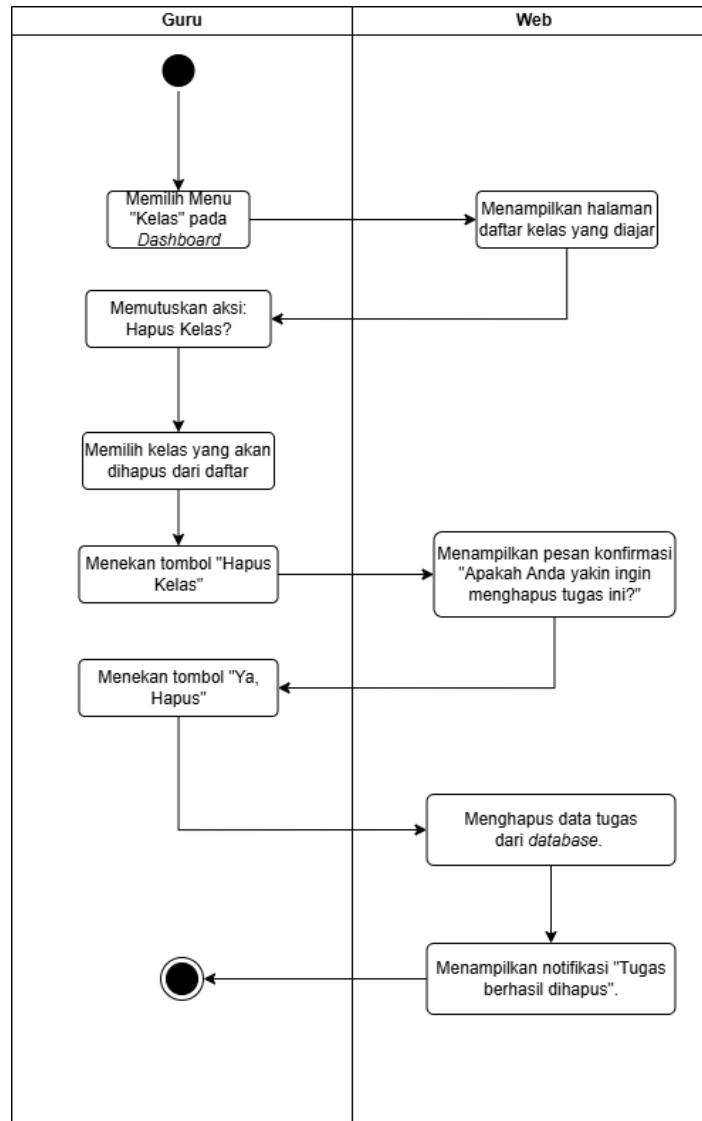
Gambar 3.4. Diagram Aktivitas Guru Edit Kelas

Activity Guru Edit Tugas



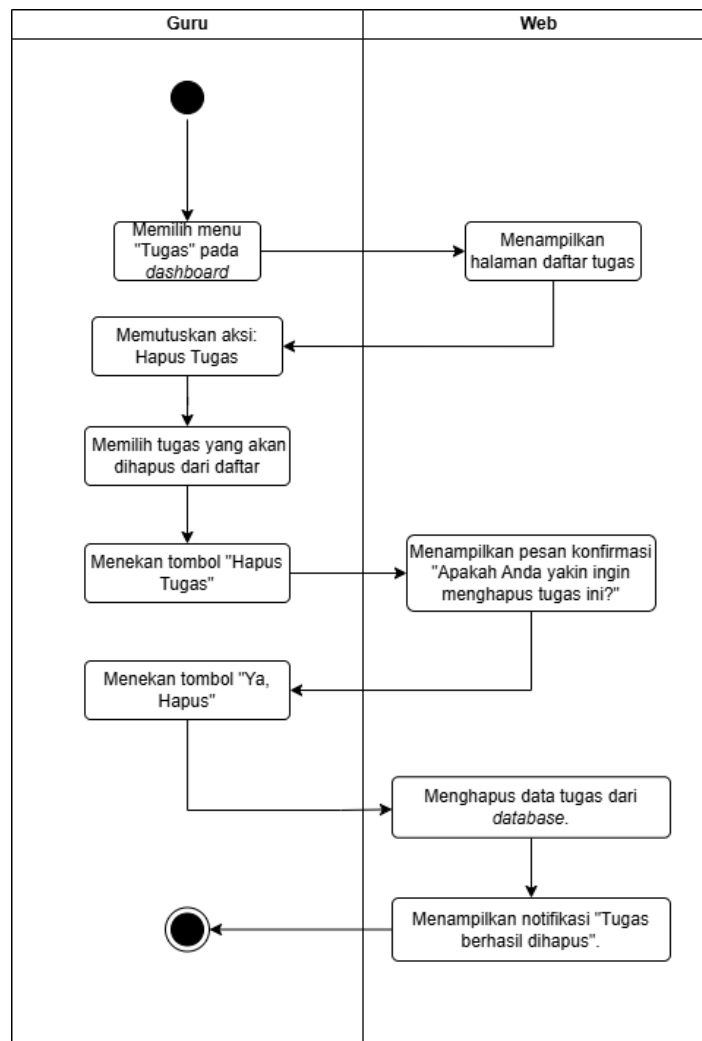
Gambar 3.5. Diagram Aktivitas Guru Edit Tugas

Activity Guru Hapus Kelas



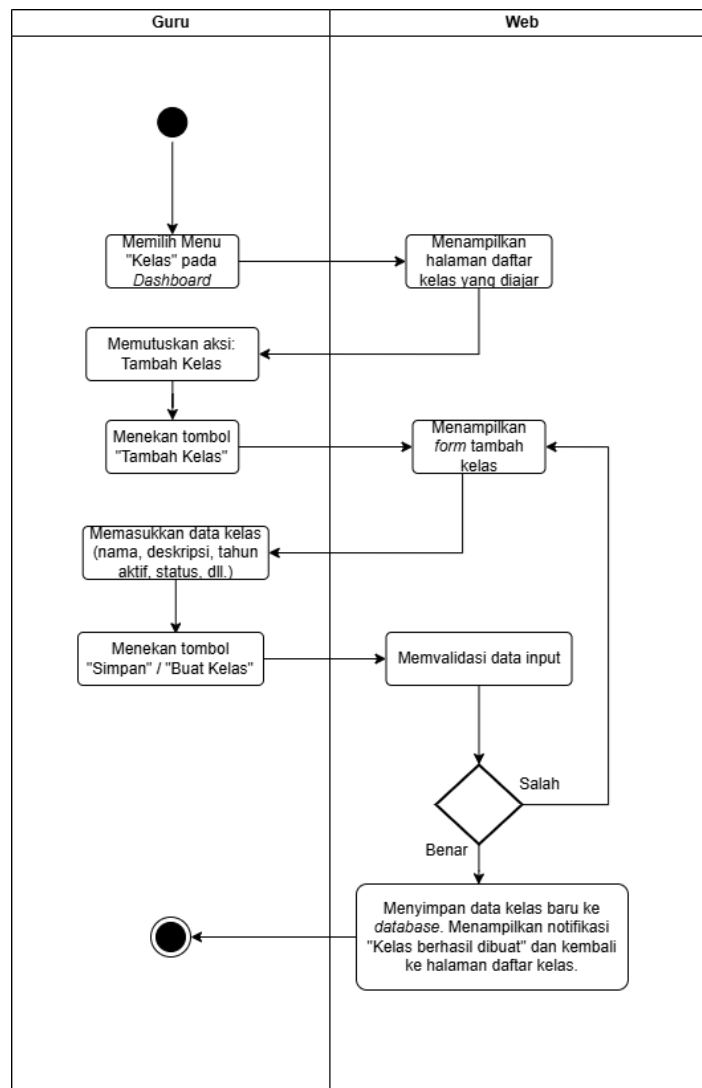
Gambar 3.6. Diagram Aktivitas Guru Hapus Kelas

Activity Guru Hapus Tugas



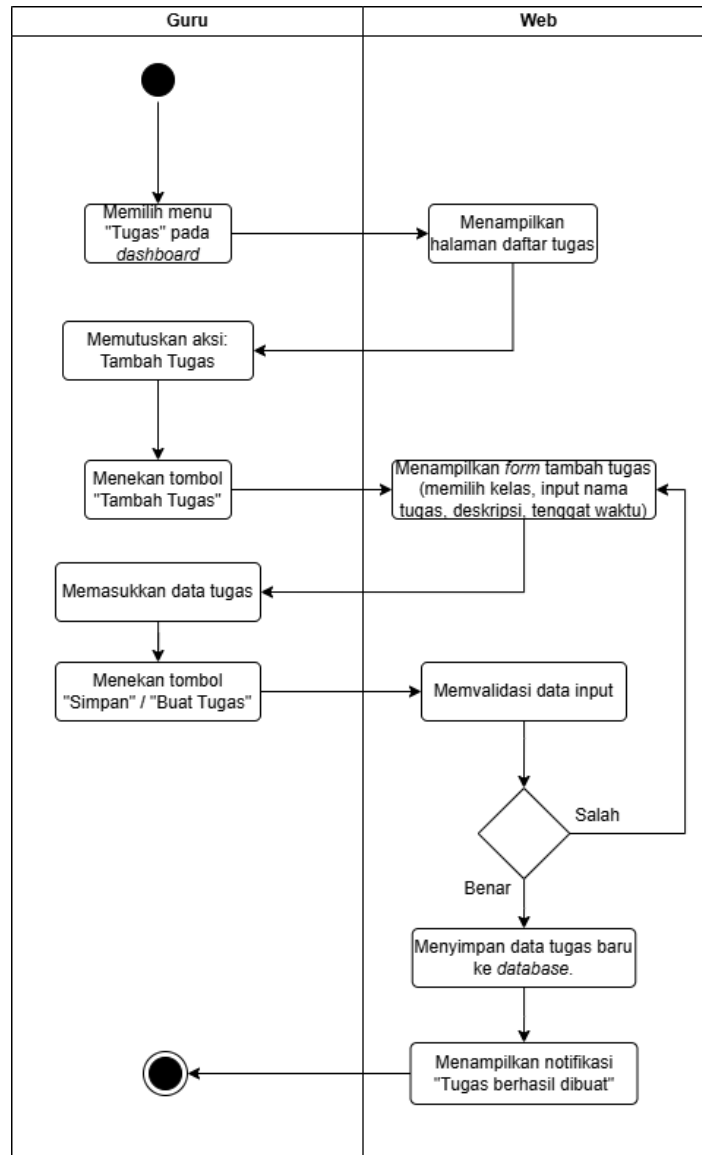
Gambar 3.7. Diagram Aktivitas Guru Hapus Tugas

Activity Guru Tambah Kelas



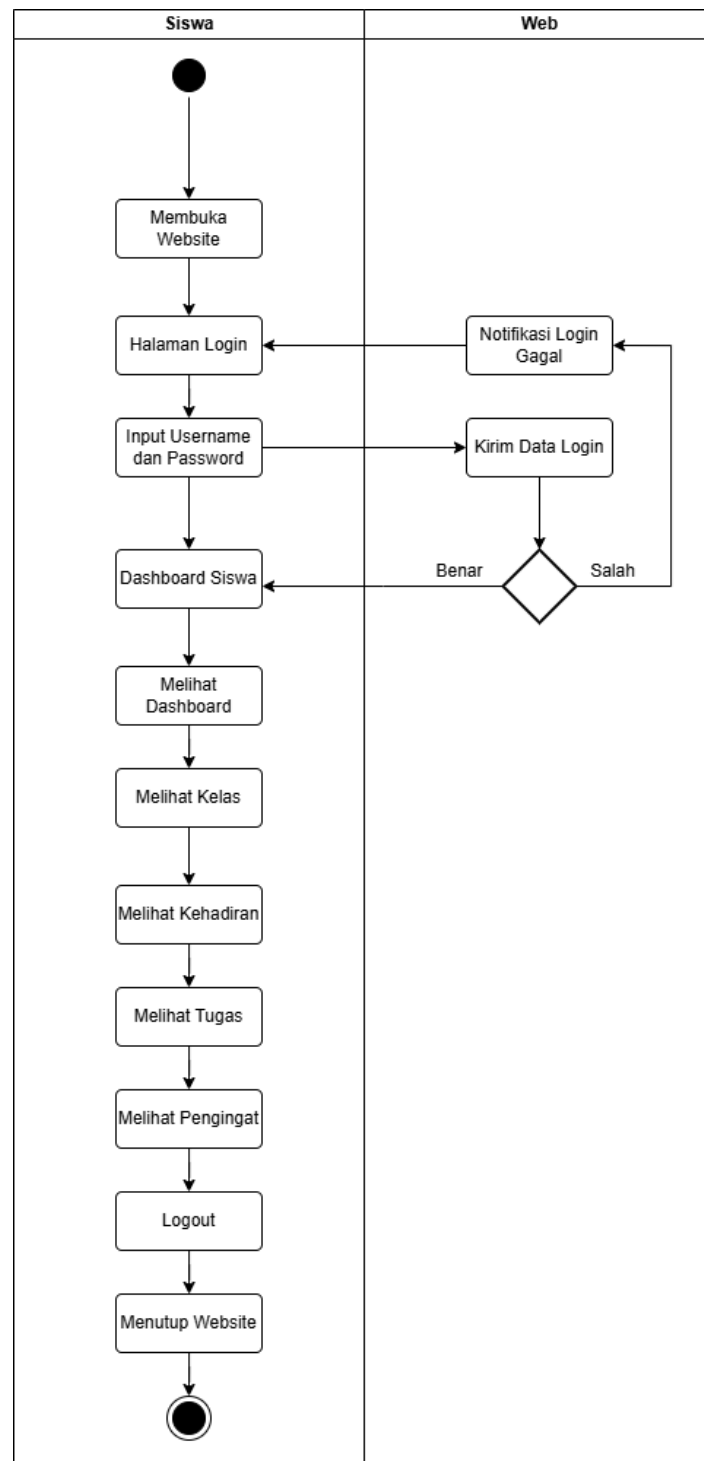
Gambar 3.8. Diagram Aktivitas Guru Tambah Kelas

Activity Guru Tambah Tugas



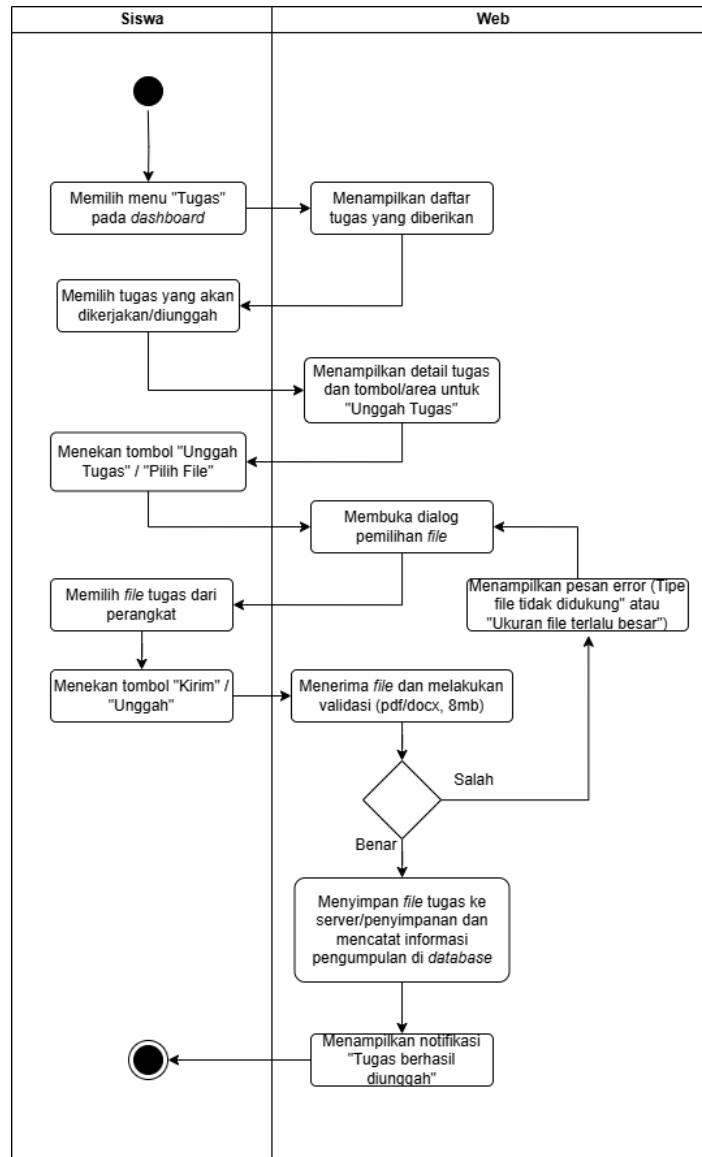
Gambar 3.9. Diagram Aktivitas Guru Tambah Tugas

Activity Siswa Akses Dashboard



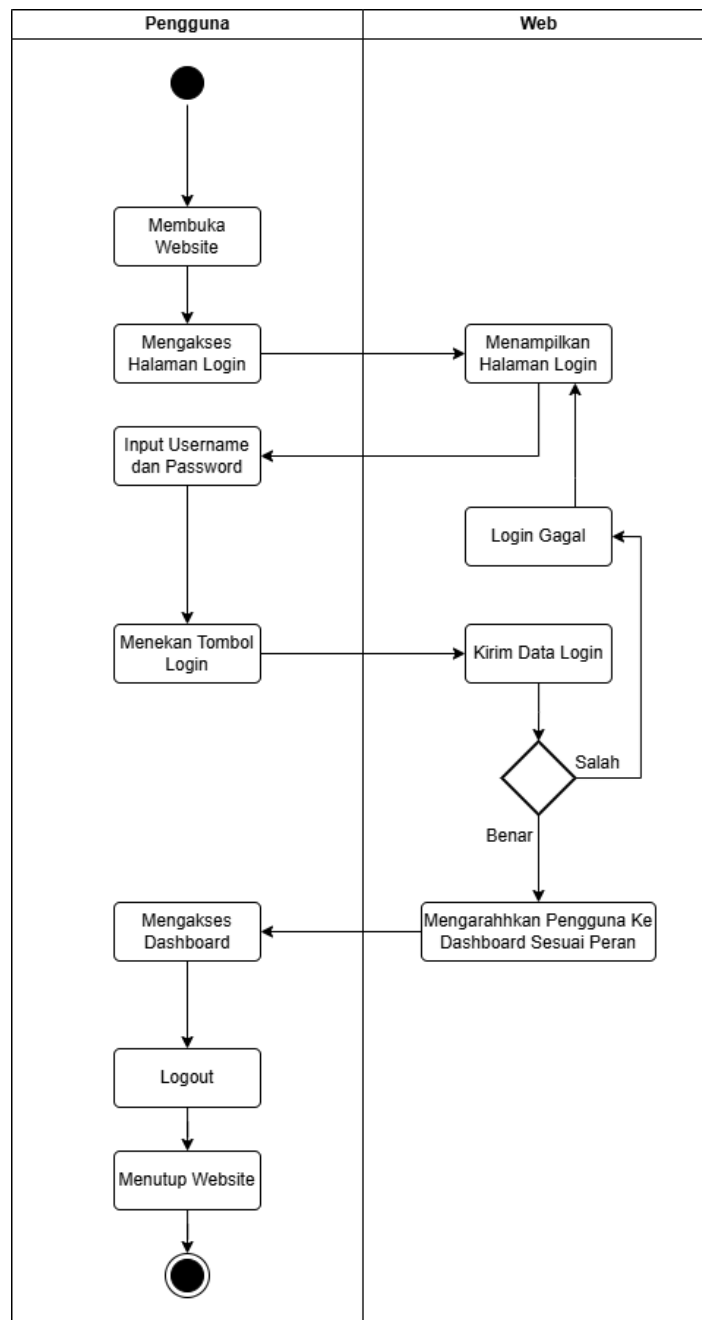
Gambar 3.10. Diagram Aktivitas Siswa Akses Dashboard

Activity Siswa Upload Tugas



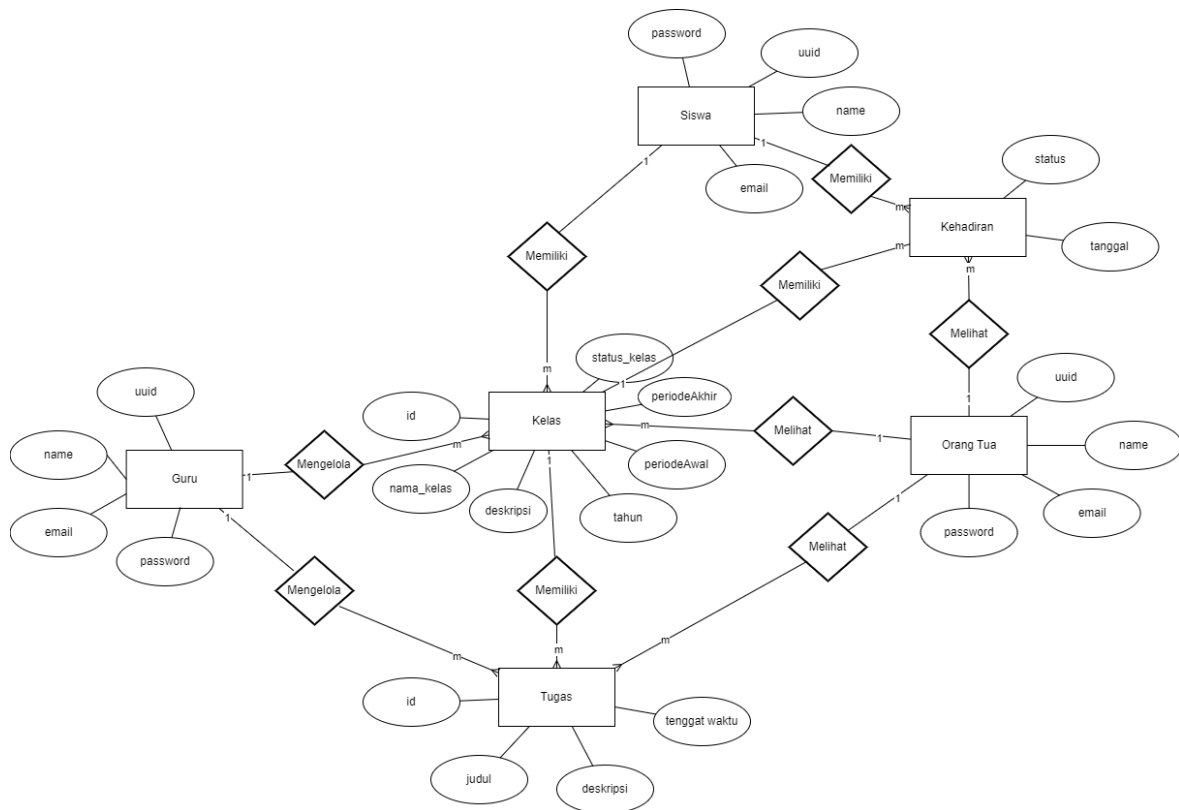
Gambar 3.11. Diagram Aktivitas Siswa Upload Tugas

Activity User Login



Gambar 3.12. Diagram Aktivitas User Login

3.1.6 Entity Relationship Diagram



Gambar 3.13. Entity Relational Diagram

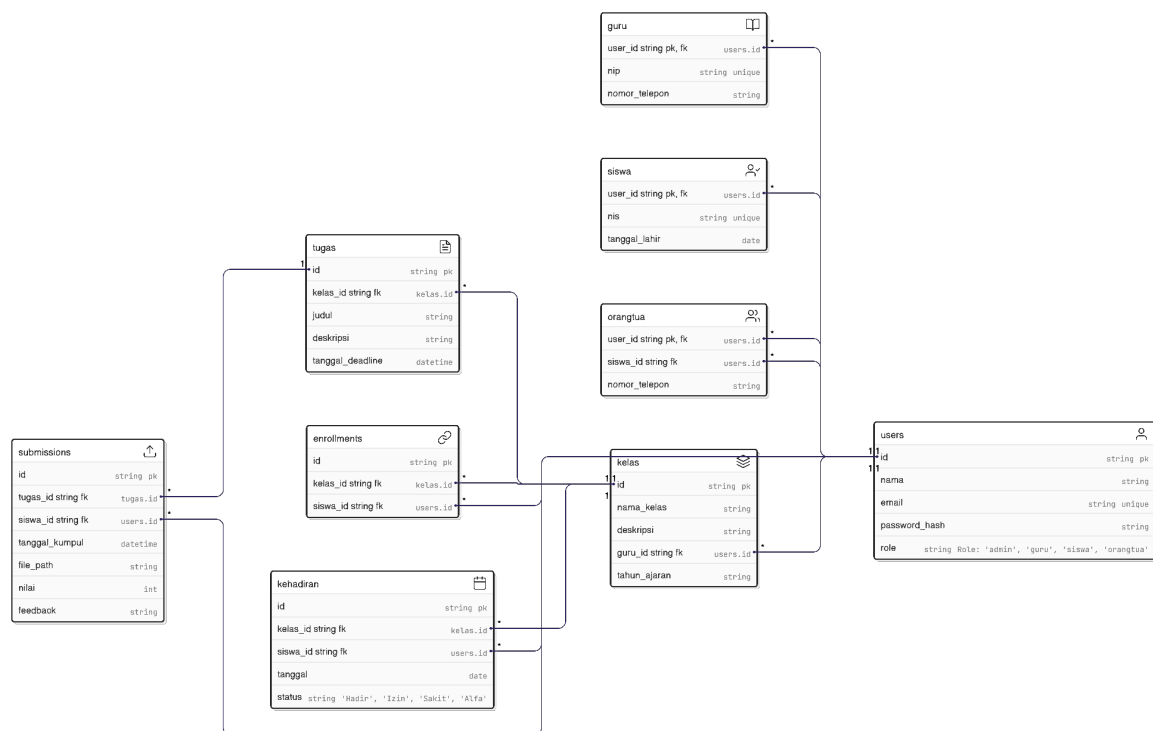
1. Siswa memiliki hubungan *one to many* dengan kelas, artinya satu murid bisa memiliki satu atau banyak kelas.
2. Siswa memiliki hubungan *one to many* dengan kehadiran, artinya satu murid bisa memiliki satu atau banyak kehadiran.
3. Orang tua memiliki hubungan *one to many* dengan kehadiran, artinya satu orang tua bisa melihat satu atau banyak kehadiran.
4. Orang tua memiliki hubungan *one to many* dengan kelas, artinya satu orang tua bisa melihat satu atau banyak kelas.
5. Orang tua memiliki hubungan *one to many* dengan tugas, artinya satu orang tua bisa melihat satu atau banyak tugas.
6. Guru memiliki hubungan *one to many* dengan kelas, artinya satu guru bisa mengelola satu atau banyak kelas.
7. Guru memiliki hubungan *one to many* dengan tugas, artinya satu guru bisa mengelola satu atau banyak tugas.
8. Kelas memiliki hubungan *one to many* dengan tugas, artinya kelas bisa memiliki satu atau banyak tugas.

9. Kehadiran memiliki hubungan *many to many* dengan kelas, artinya satu atau banyak kehadiran bisa memiliki satu atau banyak kelas.

3.2 System and Software Design

Dalam tahap ini, pengembangan akan berfokus pada merancang keseluruhan struktur sistem yang akan dibangun serta merinci desain aplikasi yang akan diimplementasikan dengan tujuan utama menghasilkan desain rinci yang akan menjadi dasar untuk memulai proses implementasi.

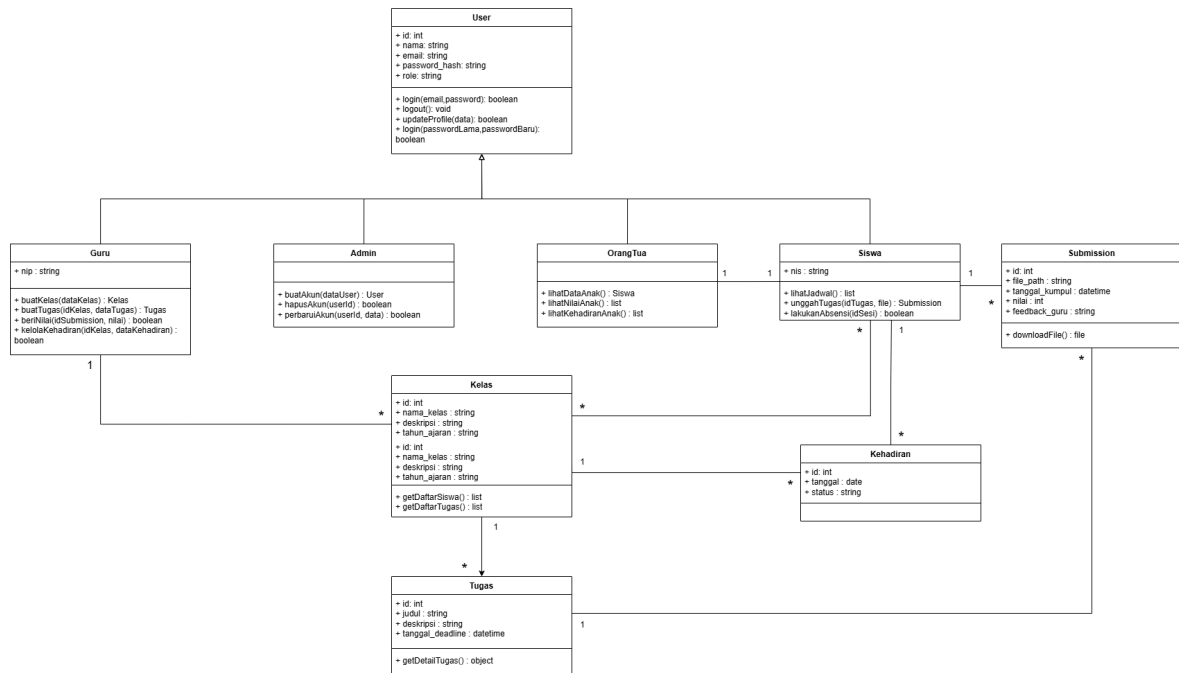
3.2.1 Physical Data Model (PDM)



Gambar 3.14. Physical data Model

3.2.2 Class Diagram

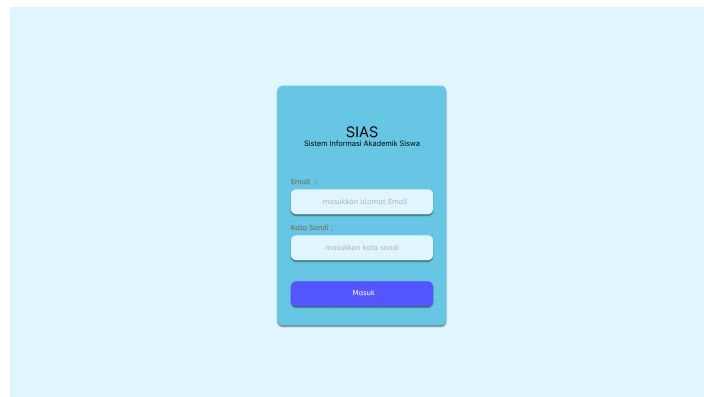
Class Diagram akan dikembangkan pada tahap selanjutnya untuk memodelkan struktur dan perilaku objek dalam sistem



Gambar 3.15. Class Diagram

3.2.3 High Fidelity

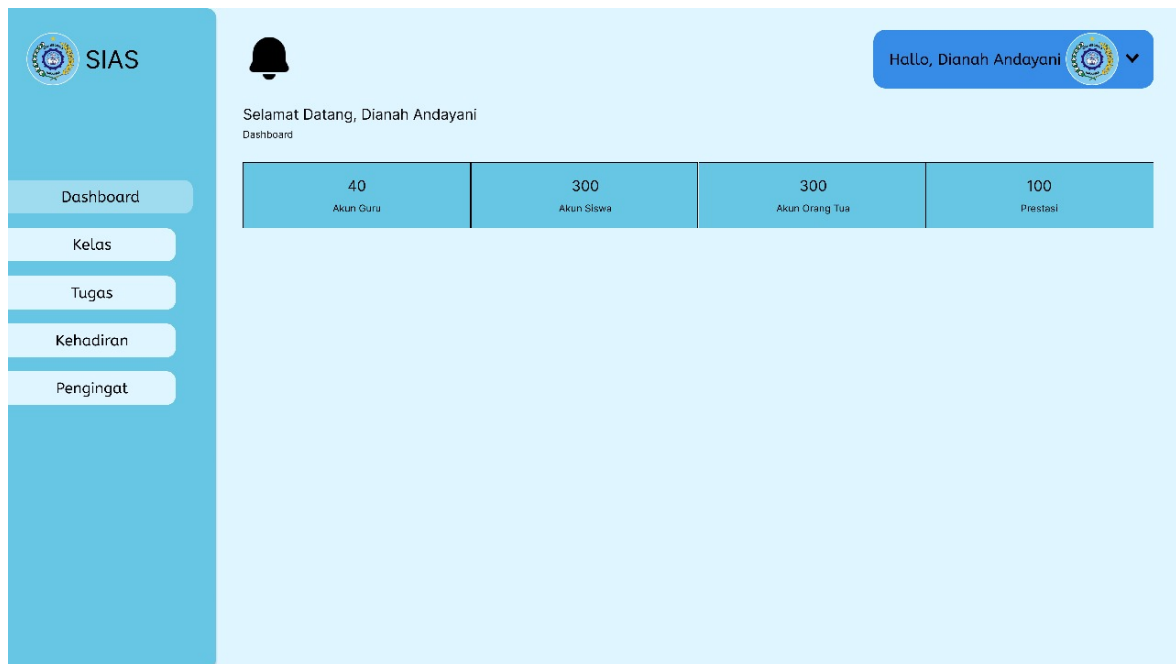
0. Login Design



Gambar 3.16. Desain halaman login.

Halaman ini digunakan untuk *login* agar dapat mengakses halaman *dashboard* sehingga dapat menggunakan fitur-fitur yang telah ada.

1. Admin Design



Gambar 3.17. Desain halaman *admin*.

Halaman ini merupakan halaman *admin* yang digunakan untuk mengelola akun pengguna.

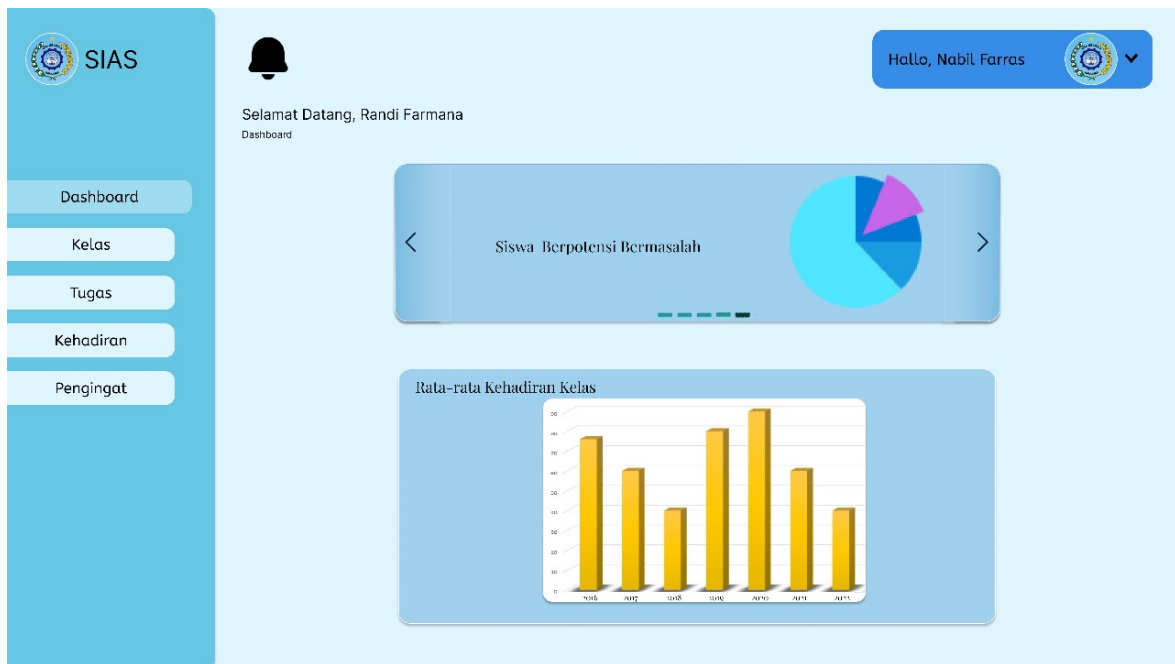
2. Parent Design



Gambar 3.18. Desain halaman orang tua.

Halaman ini merupakan halaman untuk orang tua atau wali siswa untuk melihat dan memantau kehadiran, tugas, dan kelas siswa.

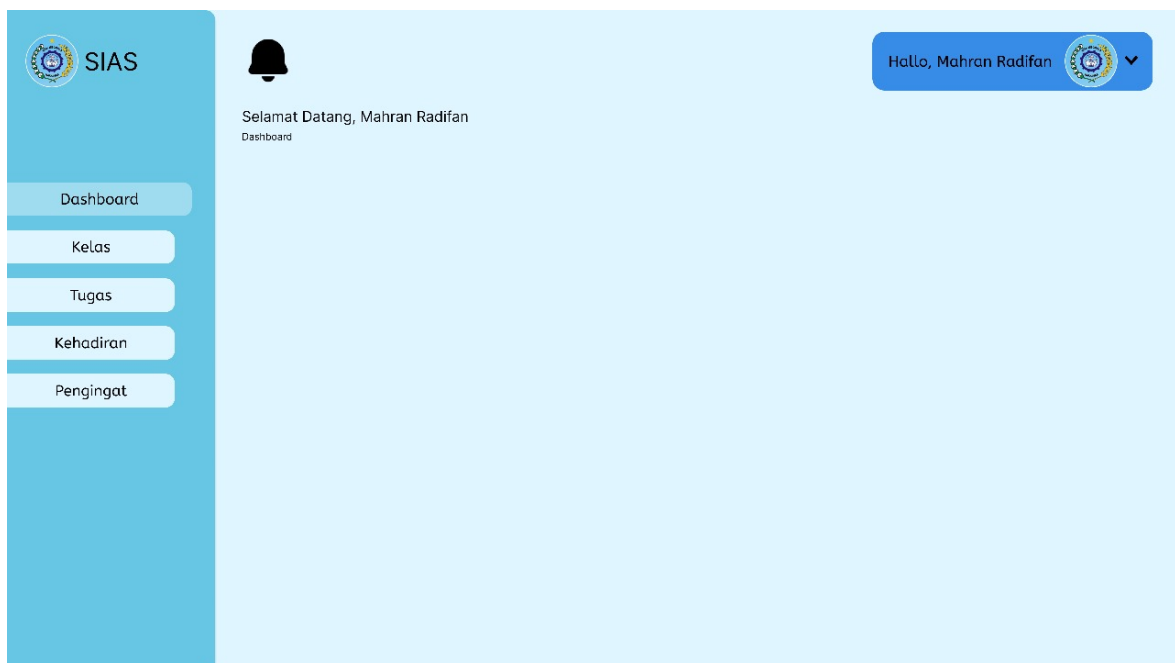
3. Teacher Design



Gambar 3.19. Desain halaman guru.

Halaman ini merupakan halaman untuk guru yang berfungsi untuk mengelola data siswa seperti data kelas, nilai, tugas, dan kehadiran.

4. Design Siswa



Gambar 3.20. Desain halaman siswa.

Halaman ini merupakan halaman untuk siswa sekolah untuk melihat kehadiran, tugas, dan nilai serta dapat mengunggah tugas.

3.3 Implementation

Setelah requirements serta design system dan aplikasi sudah didapatkan maka langkah selanjutnya yaitu pengimplementasian kode untuk membuat aplikasi,

3.4 Integration and System Testing

Setelah beberapa fitur aplikasi berhasil dibuat pada tahap Implementation and Unit testing langkah selanjutnya yaitu melakukan integrasi dan sistem testing.

Tujuan utama dari pengujian sistem adalah untuk memverifikasi bahwa sistem yang dibangun telah memenuhi semua kebutuhan fungsional (FR) dan non-fungsional (NFR) yang telah didefinisikan pada tahap analisis, serta untuk menemukan dan memperbaiki potensi kecacatan (defects) sebelum sistem diimplementasikan. Metode pengujian yang digunakan dalam penelitian ini adalah Black Box Testing. Pendekatan ini dipilih karena berfokus pada pengujian fungsionalitas sistem dari perspektif pengguna akhir, tanpa perlu mengetahui detail struktur kode internal. Pengujian dilakukan dengan memberikan input tertentu dan memvalidasi output yang dihasilkan oleh sistem, apakah sudah sesuai dengan hasil yang diharapkan.

3.4.1 Integration and System Testing

Setelah seluruh modul dan fitur aplikasi berhasil dikembangkan pada tahap implementasi, langkah selanjutnya adalah melakukan integrasi dan pengujian sistem secara menyeluruh. Pada tahap ini, semua komponen yang telah dibuat secara terpisah diintegrasikan menjadi satu kesatuan aplikasi yang utuh.

Tujuan utama dari pengujian sistem adalah untuk memverifikasi bahwa sistem yang dibangun telah memenuhi semua kebutuhan fungsional (FR) dan non-fungsional (NFR) yang telah didefinisikan pada tahap analisis, serta untuk menemukan dan memperbaiki potensi kecacatan (*defects*) sebelum sistem diimplementasikan. Metode pengujian yang digunakan dalam penelitian ini adalah *Black Box Testing*. Pendekatan ini dipilih karena berfokus pada pengujian fungsionalitas sistem dari perspektif pengguna akhir, tanpa perlu mengetahui detail struktur kode internal. Pengujian dilakukan dengan memberikan input tertentu dan memvalidasi output yang dihasilkan oleh sistem, apakah sudah sesuai dengan hasil yang diharapkan.

Skenario pengujian dikembangkan berdasarkan *use case* yang telah dirancang sebelumnya, mencakup pengujian alur normal (valid) dan alur tidak normal (invalid) untuk memastikan ketangguhan sistem. Rincian skenario pengujian disajikan pada Tabel 3.12.

Tabel 3.12. Skenario Pengujian *Black Box*

TC-ID	FR-ID	Skenario Pengujian	Langkah Pengujian	Data Uji	Hasil yang Diharapkan
-------	-------	--------------------	-------------------	----------	-----------------------

TC-01	FR-04	Pengujian login dengan kredensial yang valid.	<ul style="list-style-type: none"> • Membuka halaman login. • Memasukkan email dan kata sandi yang terdaftar. • Menekan tombol "Login". 	<ul style="list-style-type: none"> • Email: 'guru.contoh@smk.id' • Kata Sandi: 'Password-Valid123' 	Sistem berhasil melakukan autentikasi dan menampilkan halaman dashboard yang sesuai dengan peran (Guru).
TC-02	FR-04	Pengujian login dengan kata sandi yang salah.	<ul style="list-style-type: none"> • Membuka halaman login. • Memasukkan email yang benar dan kata sandi yang salah. • Menekan tombol "Login". 	<ul style="list-style-type: none"> • Email: 'guru.contoh@smk.id' • Kata Sandi: 'Password-Salah' 	Sistem menolak proses login dan menampilkan pesan kesalahan "Email atau kata sandi salah".

TC-03	FR-03	Pengujian Admin membuat akun Siswa baru dengan data yang valid.	<ul style="list-style-type: none"> • Admin melakukan login. • Mengakses menu Manajemen Pengguna. • Menekan tombol "Tambah Pengguna". • Mengisi semua kolom pada formulir dengan data yang valid. • Menekan tombol "Simpan". 	<ul style="list-style-type: none"> • Nama: 'Budi Hartono' • Email: 'budi.h@siswa.id' • Peran: 'Siswa' 	Sistem berhasil membuat akun baru. Akun tersebut muncul di daftar pengguna.
TC-04	FR-03	Pengujian Admin membuat akun dengan email yang sudah ada.	<ul style="list-style-type: none"> • Mengikuti langkah 1-4 pada TC-03. • Menggunakan email yang sudah terdaftar pada sistem. 	<ul style="list-style-type: none"> • Email: 'budi.h@siswa.id' (diasumsikan sudah ada) 	Sistem menolak pembuatan akun dan menampilkan pesan kesalahan "Email sudah digunakan".

TC-05	FR-05	Pengujian Guru membuat kelas baru.	<ul style="list-style-type: none"> • Guru melakukan login. • Mengakses menu "Kelas". • Menekan tombol "Tambah Kelas". • Mengisi nama kelas dan deskripsi. • Menekan tombol "Simpan". 	<ul style="list-style-type: none"> • Nama Kelas: 'X TKJ 1' • Deskripsi: 'Kelas 10 Jurusan TKJ' 	Kelas baru berhasil dibuat dan muncul pada daftar kelas yang diajar oleh guru tersebut.
TC-06	FR-06	Pengujian Guru membuat tugas baru dengan tenggat waktu yang valid.	<ul style="list-style-type: none"> • Guru masuk ke halaman detail kelas. • Menekan tombol "Tambah Tugas". • Mengisi judul, deskripsi, dan tenggat waktu di masa depan. • Menekan tombol "Simpan". 	<ul style="list-style-type: none"> • Judul: 'Tugas Algoritma Dasar' • Tenggat Waktu: Tanggal di masa depan (misal: 30 Oktober 2025) 	Tugas baru berhasil dibuat dan dapat dilihat oleh siswa di kelas tersebut.

TC-07	FR-06, UC-9	Pengujian Siswa mengunggah file tugas (skenario valid).	<ul style="list-style-type: none"> Siswa melakukan login. Masuk ke halaman detail tugas. Menekan tombol "Unggah Tugas". Memilih file dengan format dan ukuran yang diizinkan. Menekan tombol "Kirim". 	<ul style="list-style-type: none"> File: 'tugas_{budi}.pdf' <i>Ukuran : 2MB(batasmaks.8MB)</i> 	File berhasil diunggah. Status tugas pada antarmuka siswa berubah menjadi "Terkumpul".
TC-08	FR-06, UC-9	Pengujian Siswa mengunggah file dengan format yang tidak diizinkan.	<ul style="list-style-type: none"> Mengikuti langkah 1-3 pada TC-07. Memilih file dengan format yang tidak diizinkan (misal: .zip). 	<ul style="list-style-type: none"> File: 'tugas_{budi}.zip' 	Sistem menolak proses unggah dan menampilkan pesan kesalahan "Format file tidak didukung".
TC-09	FR-06, UC-9	Pengujian Siswa mengunggah tugas setelah melewati tenggat waktu.	<ul style="list-style-type: none"> Siswa melakukan login. Mencoba mengakses fitur unggah pada tugas yang sudah lewat tenggat waktu. 	<ul style="list-style-type: none"> Waktu saat ini > Tenggat waktu tugas 	Fitur unggah dinonaktifkan atau sistem menampilkan pesan "Tenggat waktu telah berakhir".

TC-10	FR-06, UC-4	Pengujian Guru memberikan nilai untuk tugas yang telah dikumpulkan siswa.	<ul style="list-style-type: none"> • Guru melakukan login. • Masuk ke halaman detail tugas. • Memilih siswa yang akan dinilai. • Memasukkan nilai pada kolom yang tersedia. • Menekan tombol "Simpan Nilai". 	<ul style="list-style-type: none"> • Nilai: '85' 	Nilai berhasil disimpan dan ditampilkan. Siswa dan Orang Tua dapat melihat nilai tersebut pada halaman terkait.
TC-11	FR-06, UC-4	Pengujian Guru mencoba memasukkan nilai yang tidak valid.	<ul style="list-style-type: none"> • Mengikuti langkah 1-3 pada TC-10. • Memasukkan nilai di luar rentang (misal: 6 100) atau format non-numerik. 	<ul style="list-style-type: none"> • Nilai: '105' atau 'Bagus' 	Sistem menolak input dan menampilkan pesan validasi "Masukkan nilai antara 0-100".

width=!,height=!,

BAB IV

HASIL DAN PEMBAHASAN

4.1 *Communication*

Dalam tahap *communication*, penulis telah melakukan penelusuran informasi dan pengumpulan data mengenai sistem yang akan dirancang.

4.2 *Planning*

Tabel *Gantt chart* berikut merupakan rencana pengembangan yang akan dilakukan dan berisikan tahapan utama beserta dengan sub tahapan.

Kegiatan	Januari	Februari	Maret	April	Mei	Juni
Communication						
<i>Requirement Gathering</i>						
Planning						
<i>Scheduling</i>						
Modeling						
<i>Analysis</i>						
<i>Designing</i>						
Construction						
<i>Coding</i>						
<i>Testing</i>						

Gambar 4.1. Tabel *Gantt Chart* Pelaksanaan Pengembangan (Pressman, 2010)

4.3 *Modeling*

Tahapan ini dilakukan setelah perencanaan pengembangan selesai dilakukan. Pada tahap ini pengembang akan mendesain kerangka sistem sesuai dengan data yang telah didapatkan pada tahap *communication*.

4.3.1 *Analysis*

Aplikasi serupa yang diamati adalah Ceria. Aplikasi CERIA merupakan Sistem Informasi Akademik yang berbasis website dengan menggunakan framework Laravel yang juga dapat memonitoring siswa di sekolah, didapatkan beberapa referensi fitur yang sekiranya dapat diimplementasikan dalam aplikasi yang akan dirancang ini, yaitu:

1. Aplikasi memiliki fitur kehadiran sehingga orang tua dan siswa dapat melihat kehadiran serta

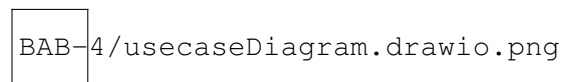
guru dapat menambah kehadiran

2. Aplikasi memiliki fitur tugas sehingga guru dapat menambahkan tugas, memasukkan nilai tugas dan siswa serta orang tua siswa dapat melihat tugas dan nilai tugas siswa.
3. Aplikasi memiliki dashboard yang dapat memudahkan akses aplikasi.

4.3.2 Design

Setelah data telah diperoleh dan dianalisis pada tahap *communication*, proses *design* dilakukan untuk merancang struktur sistem yang akan dibuat.

4.3.3 Use Case Diagram



Gambar 4.2. *Use Case Diagram* Sistem Informasi Akademik Siswa

Pada use case diagram tersebut, dapat dilihat terdapat empat aktor yang akan berinteraksi dengan sistem yaitu Admin, Guru, Orang Tua, dan Siswa. Keempat aktor tersebut memiliki batasan tersendiri dalam menggunakan fitur website seperti aktor Admin yang hanya dapat mengelola akun, aktor Guru tidak dapat mengelola akun tetapi dapat mengakses fitur lainnya seperti mengelola tugas, mengelola kelas, mengelola nilai, dan mengelola kehadiran. Sedangkan untuk aktor orang tua dan siswa hanya terdapat satu perbedaan yaitu pada *use case* mengunggah tugas. Penjelasan mengenai keterangan use case diagram diatas adalah sebagai berikut:

4.3.4 Activity Diagram

Activity diagram dibuat sebagai alat bantu untuk memahami mekanisme kerja sistem dengan mendetailkan proses-proses yang berlangsung dalam suatu aktivitas. Sebagai pengembangan dari *use case diagram*, *activity diagram* menjelaskan secara rinci langkah-langkah yang ditempuh untuk menuntaskan suatu aktivitas.

Activity diagram di atas merupakan urutan aksi dari *use case* untuk *login*. Aktivitas ini dapat dilakukan oleh pengguna yang akunnya telah terdaftar di dalam *database*. Jika *login* berhasil, pengguna akan masuk ke dalam *dashboard*.

4.3.5 Entity Relationship Diagram

Pada tahap ini penulis akan merancang *Entity Relationship Diagram* setelah peneliti melengkapi beberapa *requirement* yang belum terpenuhi.

Kode	Use Case	Keterangan
UC-1	Login	Pengguna dapat melakukan <i>login</i> untuk masuk ke <i>dashboard</i> .
UC-2	Menampilkan Dashboard	Pengguna dapat melihat <i>dashboard</i> setelah melakukan <i>login</i> .
UC-3	Mengelola Akun	<i>Admin</i> dapat menambah, menghapus, dan memperbaharui akun pengguna.
UC-4	Mengelola Kelas	Guru dapat menambah, menghapus, dan memperbaharui kelas.
UC-5	Mengelola Tugas	Guru dapat menambah, menghapus, dan memperbaharui tugas.
UC-6	Mengelola Nilai	Guru dapat menambah, menghapus, dan memperbaharui nilai.
UC-7	Mengelola Kehadiran	Guru dapat menambah, menghapus, dan memperbaharui kehadiran.
UC-8	Melihat Kelas	Orang tua dan siswa dapat melihat daftar kelas.
UC-9	Melihat Tugas	Orang tua dan siswa dapat melihat daftar tugas.
UC-10	Melihat Kehadiran	Orang tua dan siswa dapat melihat daftar kehadiran.
UC-11	Mengunggah Tugas	Siswa dapat mengunggah tugas.

Gambar 4.3. Kode dalam keterangan *use case*

4.3.6 *Physical Data Modeling*

Pada tahap ini penulis akan merancang *Physical Data Modeling* setelah peneliti melengkapi beberapa *requirement* yang belum terpenuhi.

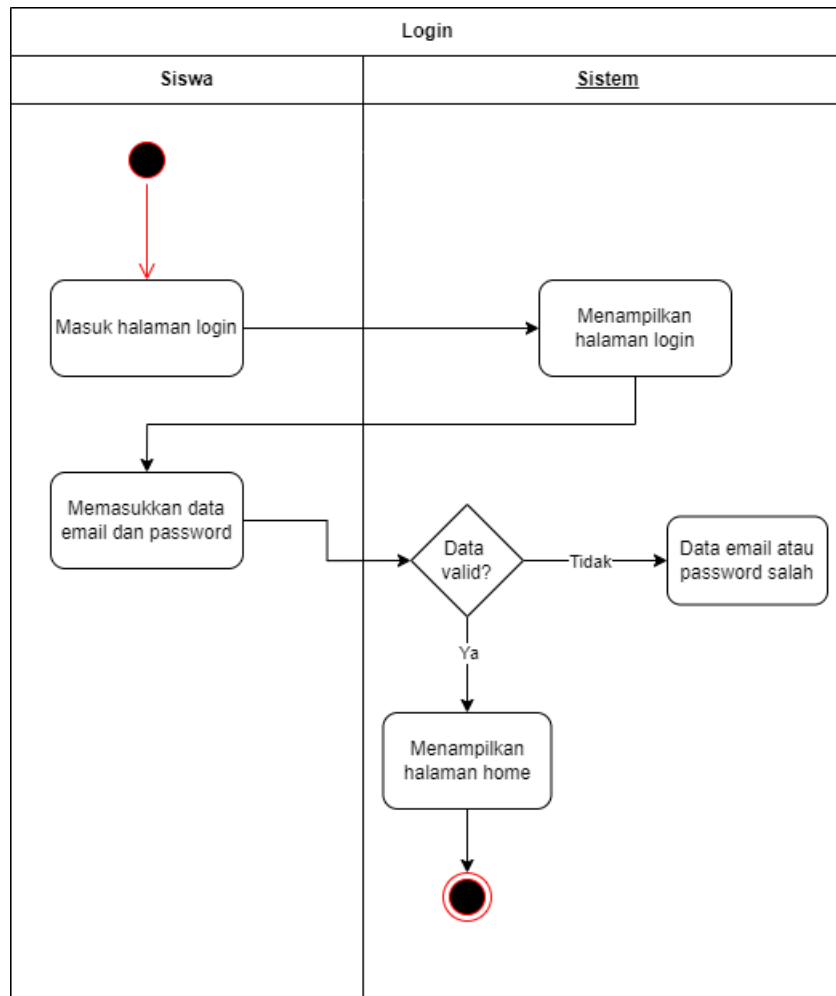
4.3.7 *Mock-up Design*

4.3.8 *Login Design*

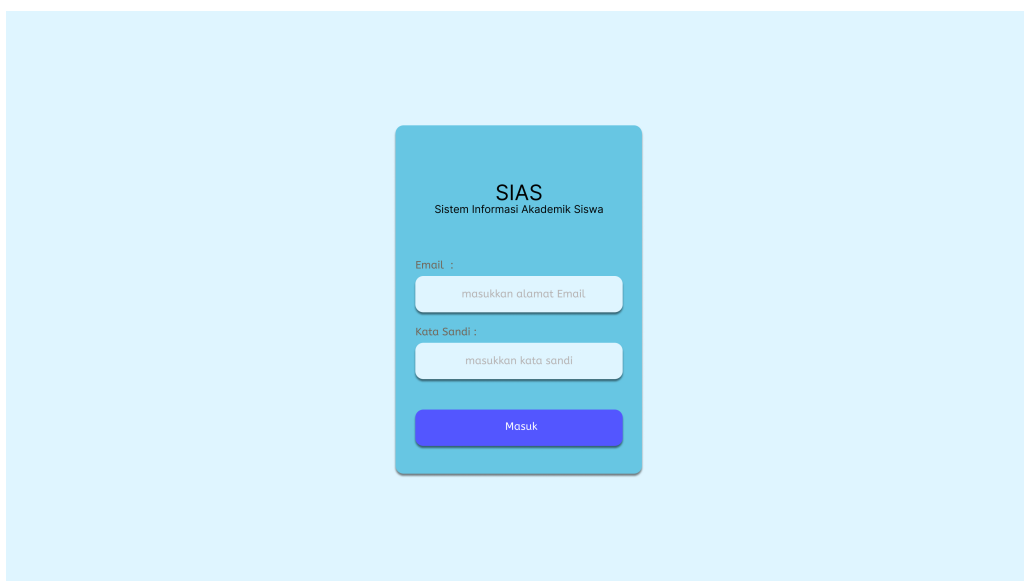
Halaman ini digunakan untuk *login* agar dapat mengakses halaman *dashboard* sehingga dapat menggunakan fitur-fitur yang telah ada.

4.3.9 *Admin Design*

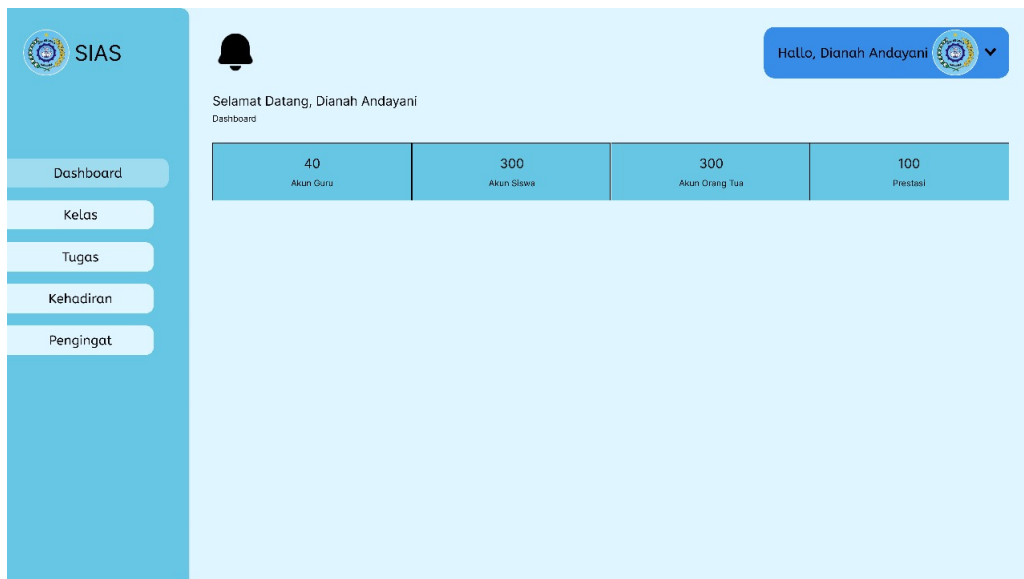
Halaman ini merupakan halaman *admin* yang digunakan untuk mengelola akun pengguna.



Gambar 4.4. Activity Diagram Login.



Gambar 4.5. Desain halaman login.



Gambar 4.6. Desain halaman *admin*.

4.3.10 *Parent Design*

Halaman ini merupakan halaman untuk orang tua atau wali siswa untuk melihat dan memantau kehadiran, tugas, dan kelas siswa.

4.3.11 *Teacher Design*

Halaman ini merupakan halaman untuk guru yang berfungsi untuk mengelola data siswa seperti data kelas, nilai, tugas, dan kehadiran.

Halaman ini merupakan halaman untuk siswa sekolah untuk melihat kehadiran, tugas, dan nilai serta dapat mengunggah tugas.

4.4 *Construction*

Pada tahap ini adalah tahap utama untuk mengimplementasikan diagram yang telah dirancang sebelumnya.

4.4.1 *Coding*

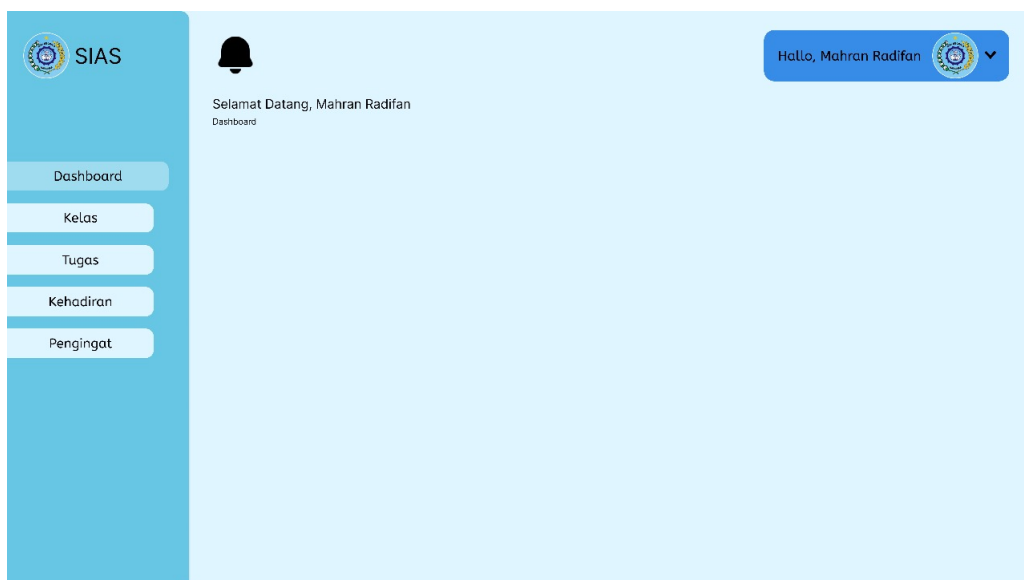
Pada tahap ini, dilakukan pemrograman untuk merealisasikan desain yang telah dibuat pada tahap *modeling*. Selama proses pemrograman ini akan digunakan kerangka kerja React JS dengan menggunakan bahasa pemrograman Node JS untuk back end. Untuk penyimpanan data akan disimpan secara lokal menggunakan *database* MySQL dengan *built-in tools* PHPMyAdmin pada XAMPP.



Gambar 4.7. Desain halaman orang tua.



Gambar 4.8. Desain halaman guru.



Gambar 4.9. Desain halaman siswa.

width=!,height=!,

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

5.2 Saran

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

width=!,height=!,

DAFTAR PUSTAKA

- Bhalla, A., Garg, S., and Singh, P. (2020). Present day web-development using reactjs. *International Research Journal of Engineering and Technology*, 07(05).
- Dicoding (2021). Apa itu activity diagram? beserta pengertian, tujuan, komponen. Diakses pada 20 November.
- Dicoding (2023). Contoh use case diagram. Diakses pada 20 November.
- Faisal, M. (2019). Rancang bangun sistem informasi housekeeping inventory dengan metode water-fall. *Jurnal Infortech*, 1(1). Universitas Bina Sarana Informatika.
- Fitriana, R. and Bakri, M. (2019). Perancangan arsitektur sistem informasi akademik menggunakan the open group arsitekture framework (togaf). *Jurnal Tekno Kompak*, 13:24.
- Husni, N. L., Sari, P. A. R., Dewi, T., Handayani, A. S., Sartika, D., and Mirza, A. (2022). Visual studio code for activity monitoring interface. In *Proceedings of the 5th FIRST T1 T2 2021 International Conference (FIRST-T1-T2 2021)*, pages 380–386. Atlantis Press.
- Khuat, T. (2018). Developing a frontend application using reactjs and redux. Master's thesis, Laurea University of Applied Sciences.
- Kurniawan, T. (2018). Pemodelan use case (uml): Evaluasi terhadap beberapa kesalahan dalam praktik. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 5.
- Lotfy, M. and Pyatt, K. (2018). The mean stack web application development platform: Tutorial presentation. *Journal of Computing Sciences in Colleges*, 34:99–101.
- Manjunath.R (2020). *C, C++, Java, Python, PHP, JavaScript and Linux For Beginners*.
- Pamungkas, D. (2020). Rancang bangun sistem informasi penjadwalan ujian tugas akhir (sijukir) berbasis web (studi kasus: Prodi informatika universitas teknologi yogyakarta).
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education.
- R., R. (2015). *Object oriented software design for association rule mining algorithms using sequence diagram*.
- Satyal, A. (2020). Designing and developing a website with reactjs. Master's thesis, Oulu University of Applied Sciences.
- Sidik, B. (2006). *Pemrograman Web dengan PHP*. Informatika, Bandung.
- Sommerville, I. (2014). *Software Engineering*. Pearson Education Limited, Edinburgh Gate, Harlow, Essex CM20 2JE, England, tenth edition. ISBN 10: 1-292-09613-6, ISBN 13: 978-1-292-09613-1.
- Syarif, M. and Nugraha, W. (2020). Pemodelan diagram uml sistem pembayaran tunai pada transaksi e-commerce. 4:64–70.
- Taniguchi, H. and Nakasho, K. (2021). Visual studio code extension and auto-completion for mizar language. pages 182–188.
- Wijaya, G. (2017). Perancangan data warehouse nilai mahasiswa dengan kimball nine-step methodology. *JURNAL INFORMATIKA*, 4(1):1–11.

width=!,height=!,

LAMPIRAN A

Judul Lampiran 1

1.1 Coding

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Cara Web Scraping

```
1
2  from bs4 import BeautifulSoup
3  import requests
4  from time import sleep
5  base_url = "https://www.oreilly.com/search/" + \
6            "?q=data&type=*&rows=10&page="
7
8
9  books = []
10 NUM_PAGES = 31
11
12 for page_num in range(1, NUM_PAGES + 1):
13     print("souping page", page_num, ",", len(books), " found")
14     url = base_url + str(page_num)
15     soup = BeautifulSoup(requests.get(url).text, 'html5lib')
16
17     for td in soup('td', 'thumbtext'): if not is_video(td):
18         books.append(book_info(td))
19
20     # give a break
21     sleep(30)
22
23 def get_year(book):
24     return int(book["date"].split()[1])
```

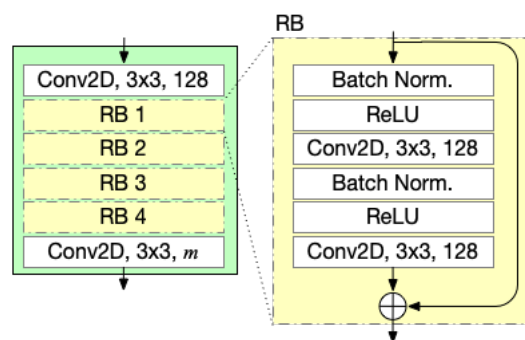
```

25
26 year_counts = Counter(get_year(book) for book in books
27                        if get_year(book) <= 2014)
28
29 import matplotlib.pyplot as plt
30 years = sorted(year_counts)
31 book_counts = [year_counts[year] for year in years]
32 plt.plot(years, book_counts)
33 plt.ylabel("# of data books")
34 plt.title("Data is Big!")
35 plt.show()
36

```

1.2 Coba Masukkan Gambar

Gambar 1.1 sama dengan di Bab 2.



Gambar 1.1. Arsitektur Blok ResNet