



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN APLIKASI MANAJEMEN DATA
OPERASIONAL BERBASIS WEB DENGAN NEXT JS**

TUGAS AKHIR

Restu Adil Salatin Syah

0110220106

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

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



Nama : Restu Adil Salatin Syah

NIM : 0110220161

Depok, Agustus 2024

STT - NF
Tanda Tangan

A handwritten signature in black ink, appearing to read 'Restu Adil Salatin Syah', written in a cursive style.

Restu Adil Salatin Syah

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Restu Adil Salatin Syah

NIM : 0110220161

Program Studi : Teknik Informatika

Judul Skripsi : Rancang Bangun Aplikasi Manajemen Data Operasional Berbasis Web dengan *Next JS*

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

DEWAN PENGUJI

Pembimbing

Penguji



(Reza Maulana, S.Kom, M.kom.)



(Dr. Lukman Rosyidi, S.T, M.M., M.T)

Ditetapkan di : Depok

Tanggal : 12 Agustus 2024

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena dengan anugerah dan kasih sayang-nya, penulis berhasil menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilaksanakan sebagai salah satu syarat untuk meraih gelar Sarjana dalam Program Studi Teknik Informatika di Sekolah Tinggi Teknologi Terpadu Nurul Fikri. Penulis menyadari bahwa tanpa dukungan dan bimbingan dari berbagai pihak, mulai dari masa perkuliahan hingga penyusunan skripsi ini, penyelesaian tugas ini akan menjadi tantangan yang berat. Penulis ingin menyampaikan terima kasih kepada:

1. Allah SWT serta semua anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
2. Bapak Dr. Lukman Rosyidi selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
3. Ibu Tiffany Nabarian., S.Kom., M.T. selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Bapak Reza Maulana, S.Kom, M.kom selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama berkulia di Sekolah Tinggi Teknologi Terpadu Nurul Fikri dan juga selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
5. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.
6. Bapak Dede Sutisna selaku Manager Pondok Bambu II.

Dalam penulisan ilmiah ini, tentu masih terdapat kekurangan yang mungkin disebabkan oleh keterbatasan pengetahuan dan kemampuan penulis. Meskipun demikian, penulis menerima dengan rendah hati kritik dan saran dari pembaca apabila terdapat kekurangan.

Sebagai penutup, penulis berharap agar Allah SWT memberikan balasan yang berlipat ganda kepada semua pihak yang telah membantu. Semoga skripsi ini dapat memberikan manfaat dalam pengembangan ilmu pengetahuan.

Depok, Juni 2024

Restu Adil Salatin Syah



STT - NF

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

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

Nama : Restu Adil Salatin Syah
NIM : 0110220161
Program Studi : Teknik Informatika
Jenis karya : Tugas Akhir

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

Rancang Bangun Aplikasi Manajemen Data Berbasis Web dengan *Next JS* beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

STT - NF

Dibuat di : Depok
Pada tanggal : 12 Agustus 2024

Yang Menyatakan



(Restu Adil Salatin Syah)

ABSTRAK

Nama : Restu Adil Salatin Syah
NIM : 0110220161
Program Studi : Teknik Informatika
Judul : Rancang Bangun Aplikasi Manajemen Data Operasional Berbasis Web dengan *Next JS*

Penelitian ini bertujuan untuk mengembangkan aplikasi pendataan operasional berbasis web bagi PT. Swadharma Sarana Informatika dengan menggunakan metode *Rapid Application Development (RAD)*. Aplikasi ini dirancang untuk meningkatkan efisiensi dan efektivitas dalam mengelola data operasional perusahaan. Metode RAD dipilih karena memungkinkan pengembangan yang cepat dan iteratif, sehingga dapat menyesuaikan kebutuhan pengguna dengan lebih baik dan cepat.

Pada tahap awal, dilakukan studi pustaka untuk mengumpulkan informasi terkait teori dan teknologi yang relevan. Tahap perencanaan meliputi identifikasi masalah, penetapan tujuan penelitian, serta pembuatan *Flowchart* dan desain sistem. *Next.js* digunakan sebagai kerangka kerja frontend karena kemampuannya dalam menyediakan pengalaman pengguna yang responsif dan performa yang tinggi. *Firebase* dipilih sebagai *Backend as a Service (BaaS)* untuk menyediakan fitur *Backend* seperti basis data, autentikasi pengguna, dan penyimpanan file.

Proses pengembangan aplikasi dilakukan dengan pendekatan iteratif, melibatkan pemangku kepentingan dan pengguna akhir untuk mendapatkan umpan balik yang cepat dan relevan. Aplikasi yang dikembangkan diuji menggunakan metode *Black Box Testing* dan *User Acceptance Testing (UAT)* untuk memastikan fungsionalitas dan kegunaan aplikasi.

Hasil penelitian menunjukkan bahwa aplikasi pendataan operasional berbasis web yang dikembangkan berhasil memenuhi kebutuhan PT. Swadharma Sarana Informatika. Aplikasi ini memudahkan proses pengelolaan data operasional, meningkatkan efisiensi kerja, dan memberikan pengalaman pengguna yang baik. Evaluasi terhadap aplikasi juga menunjukkan bahwa aplikasi ini dapat diterapkan dengan baik dan memiliki potensi untuk dikembangkan lebih lanjut sesuai kebutuhan perusahaan.

Kata kunci : *Next JS, Firebase, Blackbox*

ABSTRACT

Name : Restu Adil Salatin Syah

NIM : 0110220161

Study Program : Teknik Informatika

Title : *Design Build Web-based Operational Data Management Applications with Next JS*

This research aims to develop web-based operational deployment applications for PT. Swadharma Sarana Informatika tools using the Rapid Application Development (RAD) method. This application is designed to improve efficiency and efficiency in managing company operational data. The RAD method is chosen because it allows for fast and iterative development, so it can adapt to the needs of the user better and faster.

In the early stages, a library study was carried out to gather information related to theory and relevant technology. The planning phase included the identification of problems, the establishment of research objectives, as well as the creation of Flowcharts and system design. Next.js is used as a frontend framework because of its ability to provide a responsive user experience and high performance. Firebase is selected as a Backend as a Service (BaaS) to provide Backend features such as database, user authentication, and file storage.

The application development process is done with an iterative approach, involving stakeholders and end users to get quick and relevant feedback. The applications developed are tested using the Black Box Testing and User Acceptance Testing (UAT) methods to ensure the functionality and usability of the applications.

The results of the research show that the web-based operational deposition application developed successfully meets the needs of Swadharma Sarana Informatika. The application facilitates the process of operational data management, improves work efficiency, and provides a good user experience. Evaluations of the application also show that the application can be implemented well and has the potential to be further developed according to the needs of the company.

Keywords: Next JS, Firebase, Blackbox

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	i
HALAMAN PENGESAHAN	ii
KATA PENGANTAR	iii
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan dan Manfaat Penelitian	2
1.4 Batasan Masalah	3
1.5 Sistematika Penulisan	3
BAB II KAJIAN LITERATUR	6
2.1 Aplikasi Berbasis Website	6
2.2 Front End	7
2.3 Back End	7
2.4 Tools dan Teknologi Pengembangan	8
2.5 Unified Modeling Language (UML)	11
2.6 Metode Pengembangan Aplikasi	16
2.7 Pengujian Aplikasi	18
2.8 Penelitian Terkait	20

BAB III METODOLOGI PENELITIAN	23
3.1 Tahapan Penelitian	23
3.2 Rancangan Penelitian	26
BAB IV IMPLEMENTASI DAN EVALUASI	30
4.1 Proses Perancangan Sistem	30
4.2 Flowchart sistem berjalan	31
4.3 Use Case Diagram	34
4.4 Activity Diagram	35
4.5 Sequence Diagram	41
4.6 Class Diagram	43
4.7 State Machine Diagram	45
4.8 Component Diagram	46
4.9 Development	47
BAB V KESIMPULAN DAN SARAN	63
DAFTAR PUSTAKA	65
LAMPIRAN	68

STT - NF

DAFTAR GAMBAR

Gambar 2.1 Use case diagram.....	12
Gambar 2.2 Activity diagram.....	13
Gambar 2.3 Sequence diagram	14
Gambar 2.4 Class diagram.....	14
Gambar 2.5 State Machine diagram.....	15
Gambar 2.6 Component diagram.....	16
Gambar 4. 1 Flowchart admin.....	32
Gambar 4. 2 Flowchart staff.....	33
Gambar 4. 3Use case diagram.....	34
Gambar 4. 4 Activity diagram login	35
Gambar 4. 5 Activity diagram view profile	37
Gambar 4. 6 Activity diagram manage reports	38
Gambar 4. 7 Activity diagram manage accounts	39
Gambar 4. 8 Sequence diagram	41
Gambar 4. 9 Class diagram.....	43
Gambar 4. 10 State Machine diagram.....	45
Gambar 4. 11 Component diagram.....	46
Gambar 4. 12 Entity Relationship diagram.....	48
Gambar 4. 13 Firestore databas.....	50
Gambar 4. 14 Firestorage database	51
Gambar 4. 15 Halaman Login.....	52
Gambar 4. 16. Halaman staff cash	52
Gambar 4. 17 Halaman staff maintenance	53
Gambar 4. 18 Halaman Admin Dashboard CIT.....	53
Gambar 4. 19. Halaman Admin Dashboard maintenance.....	54
Gambar 4. 20 Modal View dan Edit	54
Gambar 4. 21 Modal Delete	55

DAFTAR TABEL

Tabel 1 Penelitian Terkait.....	20
Tabel 2 Pengujian users	56
Tabel 3 Pengujian Admin	58
Tabel 4 Pengujian UAT	61



STT - NF

BAB I PENDAHULUAN

1.1 Latar belakang

Dalam era modern saat ini, transformasi digital telah menggeser banyak aspek kehidupan kita, termasuk bagaimana kita mengakses informasi, produk, dan layanan melalui internet. Website menjadi pintu gerbang utama untuk menghubungkan pengguna dengan berbagai konten yang tersedia secara online. Kemudahan akses dan fleksibilitas yang ditawarkan oleh website memungkinkan pengguna untuk berinteraksi secara langsung dengan penyedia konten, mengakses informasi kapanpun dan dimanapun mereka berada, baik melalui perangkat seluler maupun komputer yang terhubung ke internet.

Sebuah website tidak hanya sekadar kumpulan halaman web terhubung; lebih dari itu, itu adalah medium untuk menyampaikan informasi, menyediakan layanan, dan memfasilitasi interaksi dua arah antara pengguna dan penyedia konten. Keberadaannya memungkinkan kolaborasi efektif, berbagi pengetahuan, dan komunikasi yang lebih lancar antara berbagai pihak.

Di era bisnis yang semakin terhubung dan bergerak cepat saat ini, pengelolaan data operasional menjadi krusial untuk meningkatkan efisiensi dan efektivitas bisnis. Namun, seringkali pengelolaan data ini masih mengandalkan proses manual yang rentan terhadap kesalahan, lambat, dan kurang transparan. Di PT. Swadharma Sarana Informatika, seperti halnya banyak organisasi lainnya, keterbatasan sumber daya dan cara tradisional dalam mengelola data operasional dapat menjadi penghambat utama dalam mencapai tingkat efisiensi yang diinginkan.

Oleh karena itu, penelitian ini bertujuan untuk mengatasi tantangan tersebut dengan mengembangkan sebuah website untuk pengelolaan data operasional. Melalui penerapan teknologi Next.js dan *Firebase*, proyek ini diharapkan mampu menghadirkan solusi yang lebih efisien, akurat, dan transparan dalam

pengumpulan, penyimpanan, dan analisis data secara otomatis. Kombinasi teknologi ini diharapkan dapat mengubah paradigma pengelolaan data tradisional menjadi yang lebih modern dan responsif terhadap kebutuhan bisnis saat ini.

Dalam usaha untuk memenuhi kebutuhan tersebut, peneliti akan merancang aplikasi berbasis web dengan menggunakan teknologi Next JS sebagai Frontend dan Firebase sebagai Backend As a Service

1.2 Rumusan Masalah

Berdasarkan informasi diatas, rumusan masalah penelitian ini adalah sebagai berikut.

- a) Bagaimana merancang sistem dan mengimplementasi aplikasi manajemen pengelolaan data operasional berbasis web ?
- b) Apakah aplikasi berjalan dengan baik dan sesuai dengan kebutuhan ?

1.3 Tujuan dan Manfaat Penelitian

A. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

- a) Merancang dan mengimplementasi sistem aplikasi berbasis web yang mampu memenuhi kebutuhan manajemen pengelolaan data operasional di PT.Swadharma Sarana Informatika menggunakan teknologi *Next JS* dan *Firebase*
- b) Menilai apakah aplikasi yang di kembangkan berjalan dengan baik dan sesuai dengan spesifikasi dan kebutuhan pengguna

B. Manfaat Penelitian

Adapun manfaat dari penelitian ini, yaitu:

- a) Peningkatan efisiensi dan produktivitas operasional melalui penggunaan aplikasi berbasis web sehingga dapat mempercepat pengambilan keputusan.
- b) Mengatasi hambatan atau kendala yang dihadapi dalam proses manajemen data operasional
- c) Memberikan kontribusi terhadap pengembangan teknologi *Next JS* dan *Firebase*.

1.4 Batasan Masalah

Batasan-batasan masalah dari penelitian ini adalah :

- a) Penelitian ini hanya difokuskan untuk pengelolaan data operasional di PT.SSI
- b) Pengembangan aplikasi hanya dilakukan untuk platform web dan tidak mencakup pengembangan aplikasi *mobile native*
- c) Sistem yang di buat tidak memperhitungkan segi keamanan data

1.5 Sistematika Penulisan

Dalam Menyusun karya tulis ilmiah ini, agar dalam pembahasan terfokus pada pokok permasalahan dan tidak melebar ke masalah yang lain, maka penulis membuat sistematika penulisan karya tulis ilmiah sebagai berikut

BAB I PENDAHULUAN

Dalam bab ini Penulis membahas tentang latar belakang masalah , rumusan masalah, tujuan dan manfaat penelitian, batasan masalah, serta sistematika penulisan. Pada bagian ini, diuraikan konteks dan alasan pentingnya penelitian ini, diikuti dengan pertanyaan penelitian yang ingin dijawab, tujuan yang hendak dicapai, manfaat yang diharapkan, dan struktur penulisan karya ilmiah ini

BAB II KAJIAN LITERATUR

Dalam bab ini menjelaskan landasan teori yang relevan dan akan diterapkan dalam penelitian ini. Penulis juga mengulas penelitian-penelitian terkait yang telah dilakukan sebelumnya untuk memberikan dasar yang kuat dan menunjukkan celah penelitian yang hendak diisi. Literatur yang dibahas mencakup teknologi Next JS, Firebase, pengelolaan data operasional, dan prinsip-prinsip pengembangan aplikasi berbasis web.

BAB III METODOLOGI PENELITIAN

Dalam bab ini penulis akan menjelaskan mengenai tahapan-tahapan proses penelitian yang digunakan oleh penulis. Di dalamnya terdapat rincian metode pengumpulan data, teknik analisis data, serta langkah-langkah dalam pengembangan sistem dan aplikasi berbasis web. Penulis juga menguraikan tools dan teknologi yang digunakan, seperti Next JS untuk Frontend dan Firebase untuk Backend, serta prosedur pengujian aplikasi.

BAB IV IMPLEMENTASI DAN EVALUASI

Bab ini berfokus pada proses implementasi dari desain sistem yang telah direncanakan. Penulis menjelaskan tahapan pengembangan, mulai dari instalasi Next JS, setup Firebase, hingga integrasi kedua teknologi tersebut dalam aplikasi. Selain itu, bab ini juga mencakup evaluasi hasil pengujian sistem untuk memastikan aplikasi berfungsi sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Umpan balik dari pengguna juga dianalisis untuk penyesuaian lebih lanjut.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan. Penulis merangkum temuan utama, mencerminkan bagaimana tujuan penelitian telah tercapai, dan memberikan saran untuk penelitian atau pengembangan lebih lanjut. Kesimpulan ini juga mencakup evaluasi terhadap manfaat yang dihasilkan dari penerapan aplikasi berbasis web di PT. Swadharma Sarana Informatika.

STT - NF

BAB II KAJIAN LITERATUR

2.1 Aplikasi Berbasis Website

Aplikasi berbasis website telah menjadi salah satu solusi teknologi yang sangat populer dalam berbagai bidang. Dibandingkan dengan aplikasi desktop atau mobile, aplikasi berbasis website memiliki beberapa keunggulan yang signifikan. Pertama-tama, aksesibilitasnya yang tinggi. Pengguna hanya perlu memiliki perangkat yang terhubung ke internet dan browser web untuk mengakses aplikasi tersebut, tanpa perlu mengunduh atau menginstal perangkat lunak tambahan. Hal ini membuat aplikasi berbasis website menjadi pilihan yang sangat praktis untuk berbagai penggunaan, terutama dalam lingkup bisnis[1] .

Selain itu, aplikasi berbasis website juga mendukung integrasi dan kolaborasi yang lebih baik. Dengan menggunakan teknologi web standar seperti *HTML*, *CSS*, dan *Javascript*, aplikasi dapat dengan mudah terhubung dengan layanan lainnya dan berinteraksi dengan platform atau sistem yang ada[2]. Ini memungkinkan pengguna untuk mengintegrasikan aplikasi dengan alat atau layanan lain yang mereka gunakan dalam operasi sehari-hari, meningkatkan efisiensi dan produktivitas secara keseluruhan. Dalam konteks aplikasi pendataan operasional, integrasi yang mulus dengan sistem *Backend* dan layanan lainnya menjadi kunci dalam mengoptimalkan alur kerja dan pemrosesan data[3]

Dengan demikian, aplikasi berbasis website merupakan solusi yang praktis dan efisien dalam pengembangan perangkat lunak, tetapi juga merupakan fondasi yang kuat untuk membangun sistem informasi yang handal dan terintegrasi. Dalam proyek pengembangan aplikasi pendataan operasional berbasis web dengan *Next JS*, penggunaan teknologi web modern ini menjadi kunci dalam mencapai tujuan yang ditetapkan dengan efektif dan efisien[4]

2.2 Front End

Front End pengembangan antarmuka pengguna grafis dari sebuah situs web (*UI & UX*), melalui penggunaan *HTML*, *CSS*, dan *JavaScript*, sehingga pengguna dapat melihat dan berinteraksi dengan situs web tersebut. Tugas mereka adalah memastikan bahwa pengguna dapat berinteraksi dengan situs web atau aplikasi secara efektif dan menyenangkan. Mereka bertanggung jawab atas aspek *visual* dan fungsional dari tampilan pengguna, mulai dari desain hingga implementasi. Seorang *Front End Developer* bekerja sama dengan desainer untuk menghasilkan antarmuka yang sesuai dengan visi dan tujuan proyek. Mereka juga bertanggung jawab untuk menganalisis, mengkodekan, dan menguji elemen-elemen sisi klien dari aplikasi, memastikan bahwa situs web atau aplikasi berfungsi dengan baik di berbagai *platform* dan perangkat[5].

Front End Developer memiliki tanggung jawab mengintegrasikan teknik desain, teknologi, dan pemrograman untuk menciptakan tampilan situs web atau aplikasi yang menarik dan interaktif, sambil menangani debugging jika terjadi masalah. Mereka bertugas memastikan bahwa setiap pengunjung dapat dengan mudah mengakses dan menggunakan aplikasi atau situs web dengan lancar. Sebagai contoh, setiap kali seorang pengguna mengunjungi situs web, semua elemen yang dilihat, diklik, atau digunakan merupakan hasil karya dari *Front End Developer*. Mulai dari tata letak, konten, tombol, hingga navigasi dan tautan internal, semuanya dirancang dan diimplementasikan oleh *Front End Developer* [6]

2.3 Back End

Backend merupakan bagian dari sebuah aplikasi atau situs web yang tidak terlihat oleh pengguna akhir, tetapi berperan penting dalam menyediakan fungsionalitas dan menyimpan data. Ini adalah bagian dari sistem yang berada di sisi server[7], yang melakukan berbagai tugas seperti pemrosesan data, mengelola logika bisnis, serta berinteraksi dengan database untuk menyimpan dan mengambil informasi.

Salah satu komponen utama dari *Backend* adalah server. Server ini bertindak sebagai tuan rumah untuk aplikasi web, menerima permintaan dari browser pengguna, memprosesnya, dan mengirimkan kembali respons yang sesuai. *Backend developer* biasanya bertanggung jawab untuk mengatur dan mengelola server agar dapat menjalankan aplikasi dengan lancar, serta menangani pertumbuhan dan lalu lintas yang meningkat seiring waktu[8].

Selanjutnya, aplikasi *Backend* menyimpan logika bisnis dari aplikasi, termasuk semua fungsi dan proses yang diperlukan untuk menjalankan fitur-fitur tertentu. Misalnya, dalam aplikasi e-commerce, aplikasi *Backend* akan mengatur proses pembayaran, manajemen inventaris, dan pemrosesan pesanan. *Backend developer* bertugas untuk merancang, mengembangkan, dan memelihara kode aplikasi ini agar berjalan sesuai dengan kebutuhan bisnis dan spesifikasi teknis.

Basis data juga merupakan komponen kunci dari *Backend*, karena digunakan untuk menyimpan dan mengelola data yang diperlukan oleh aplikasi. *Backend developer* akan merancang *schema database*, membuat kueri untuk mengambil dan menyimpan data, serta memastikan integritas dan keamanan data. Berbagai jenis basis data dapat digunakan, mulai dari basis data relasional tradisional hingga basis data *NoSQL* yang lebih fleksibel [9]

2.4 Tools dan Teknologi Pengembangan

1. Visual Studio Code

Visual Studio Code (VS Code) merupakan sebuah *code editor open source* yang dikembangkan oleh *Microsoft*, dirancang untuk beroperasi pada komputer desktop, dan kompatibel dengan sistem operasi *Windows, macOS, dan Linux*. Sebagai sebuah code editor, *VS Code* memiliki fokus utama pada kemudahan penggunaan dan fleksibilitas dalam pengembangan perangkat lunak. *VS Code* sangat cocok

digunakan untuk pengembangan aplikasi berbasis web dan *mobile* yang menggunakan *JavaScript*, *TypeScript*, dan *Node.JS*.

Kemampuannya dalam menyediakan fitur-fitur yang canggih, seperti *code completion*, *debugging*, dan *version control*, membuatnya menjadi pilihan populer di kalangan *developer*. *VS Code* juga memiliki ekosistem *extension* yang luas, yang memungkinkan pengguna untuk mengkustomisasi dan memperluas fungsionalitas editor sesuai dengan kebutuhan pengembangan mereka. Ekstensi yang tersedia dapat mendukung berbagai bahasa pemrograman dan runtime lainnya, seperti *C++*, *C#*, *Java*, *Python*, *PHP*, *Go*, dan *.NET*[10]

2. *Postman*

Postman adalah sebuah *software* yang dapat digunakan untuk menguji *api*. dalam *software* ini hampir semua kebutuhan *developer* untuk melakukan *testing* terhadap *api* dari mulai permintaan *HTTP* seperti *GET*, *POST*, *PUT*, *DELETE*, dan lainnya[11]

Dengan bantuan *Postman*, penulis dapat menghemat waktu dan upaya dalam menguji dan memvalidasi *API*, sehingga memungkinkan peneliti untuk fokus pada pengembangan aplikasi dengan kualitas yang lebih baik. *Postman* telah menjadi salah satu alat yang sangat penting dalam alur kerja pengembangan perangkat lunak modern, membantu para *developer* dalam menghadapi tantangan pengujian *API* dengan lebih efektif dan efisien

3. *Next JS*

Next JS adalah sebuah *Framework JavaScript* yang populer untuk pengembangan aplikasi web *React*. Dirancang untuk mempermudah pengembangan aplikasi web yang kuat dan efisien, *Next JS* menyediakan berbagai fitur yang mendukung penggunaan *React* di sisi server dan *client*[12] yang memudahkan untuk pengembangan.

Next JS adalah sebuah *Framework* yang memungkinkan pengembang untuk membangun aplikasi web React dengan mudah dan efisien. Salah satu fitur utamanya adalah kemampuannya dalam menyediakan *Server-Side Rendering (SSR)* dan *Static Site Generation (SSG)*[13], yang meningkatkan kinerja dan *SEO* aplikasi web. Selain itu, *Next JS* juga menyediakan dukungan bawaan untuk *TypeScript*, *CSS Modules*, dan berbagai pustaka *UI* populer, seperti *Material-UI* dan *Tailwind CSS*, yang memudahkan pengembang untuk menggunakan teknologi-teknologi ini tanpa perlu konfigurasi tambahan.

Dengan API yang komprehensif, *Next JS* mempermudah penanganan berbagai tugas pengembangan aplikasi web, termasuk pembaruan otomatis, pre-fetching data, dan penanganan kesalahan. Cocok untuk berbagai jenis proyek, dari yang sederhana hingga kompleks, *Next JS* menawarkan dukungan penuh untuk berbagai lingkungan pengembangan dan penyebaran.

Secara keseluruhan, *Next JS* adalah pilihan yang kuat dan fleksibel bagi pengembang yang ingin meningkatkan produktivitas dan kinerja aplikasi web mereka. Dengan kombinasi *SSR*, *SSG*, sistem routing yang kuat, dan dukungan untuk teknologi teknologi terkini, *Next JS* menjadi *Framework* yang populer dan diminati dalam pengembangan aplikasi web *React*

4. *Tailwind CSS*

Tailwind CSS merupakan sebuah *Framework CSS* yang inovatif yang dirancang untuk memudahkan pengembang dalam melakukan *styling* pada website mereka. Dibandingkan dengan pendekatan tradisional, *Tailwind CSS* menggunakan pendekatan *utility-first* yang memungkinkan penggunaan kelas-kelas utilitas yang terdefinisi dengan baik untuk membangun desain antarmuka sebuah website secara efisien.

Pendekatan *utility-first* yang digunakan oleh *Tailwind CSS* memungkinkan penulis untuk memiliki kontrol yang lebih besar terhadap desain antarmuka, sambil tetap

mempertahankan konsistensi dan kejelasan dalam kode. Dengan demikian, *Tailwind CSS* menjadi pilihan yang populer bagi penulis yang mengutamakan kecepatan dan efisiensi dalam pengembangan desain antarmuka website.[14]

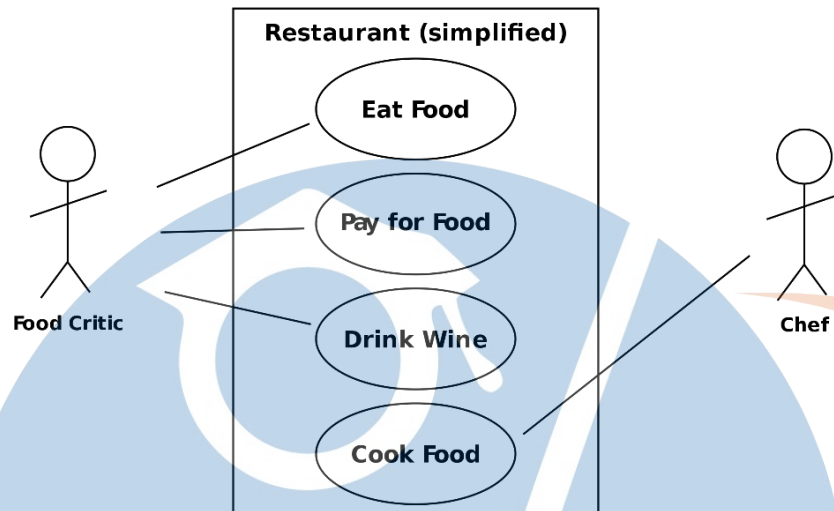
5. *Firebase*

Firebase adalah sebuah *Realtime Database* yang disimpan di *cloud* dan mendukung aplikasi *multiplatform* seperti *Android*, *iOS*, dan *Web*. Dengan basis *NoSQL* dan penyimpanan dalam format *JSON* (*JavaScript Object Notation*[15]), *Firebase* menyediakan *sinkronisasi* otomatis dengan aplikasi klien yang terhubung, menjadikannya solusi *Backend as a Service* (*BaaS*[16]) yang dikembangkan oleh Google. Dengan layanan *Firebase*, para pengembang perangkat lunak dapat dengan mudah mengembangkan aplikasi mereka, termasuk fitur autentikasi untuk *login* pengguna dan penyimpanan berkas melalui *Firebase Storage*[17].

2.5 Unified Modeling Language (UML)

Unified Modeling Language (*UML*) adalah bahasa pemodelan *visual* yang secara luas digunakan dalam industri perangkat lunak untuk merancang dan memodelkan sistem. Dengan notasi grafis yang komprehensif, *UML* memungkinkan para pengembang untuk merepresentasikan berbagai aspek dari sistem, termasuk struktur, perilaku, dan interaksi antar objek. Menurut Grady Booch, salah satu ahli yang terlibat dalam pengembangan *UML*, bahasa ini dapat digunakan untuk merepresentasikan sistem perangkat lunak dari yang sederhana hingga yang kompleks. *UML* terdiri dari berbagai diagram yang masing-masing memiliki tujuan khusus, seperti *class diagram*, *activity diagram*, dan lainnya, yang membantu para pengembang dalam memodelkan dan merancang sistem secara sistematis.[18]

2.1.1 Use Case Diagram

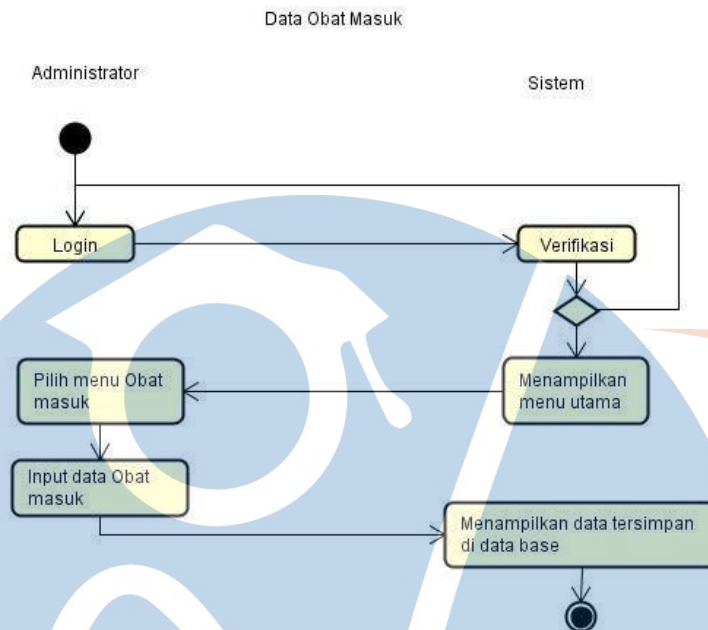


Gambar 2 1Use case diagram

Diagram ini menunjukkan bagaimana sistem digunakan oleh pengguna dalam situasi tertentu. *Use Case Diagram* membantu para pengembang untuk memahami kebutuhan dan persyaratan pengguna serta memastikan bahwa sistem yang dikembangkan memenuhi kebutuhan tersebut. Dengan melihat diagram ini, tim pengembangan dapat dengan jelas melihat bagaimana pengguna akan berinteraksi dengan sistem dalam berbagai skenario penggunaan, membantu mereka dalam merancang sistem yang responsif dan sesuai dengan kebutuhan pengguna[19].

STT - NF

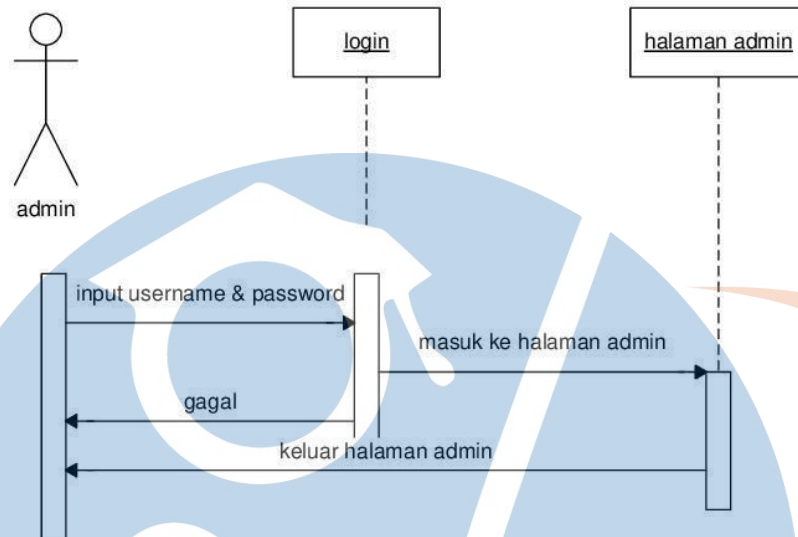
2.1.2 Activity Diagram



Gambar 2.2 Activity diagram

Activity diagram, atau yang dikenal sebagai diagram aktivitas, adalah representasi visual dari proses-proses yang terjadi dalam sistem. Diagram ini memodelkan urutan langkah-langkah atau aktivitas yang dilakukan dalam suatu proses, yang digambarkan secara vertikal. *Activity diagram* merupakan bagian dari *Unified Modeling Language (UML)* dan sering digunakan dalam pengembangan perangkat lunak sebagai pengembangan dari *Use Case Diagram*. Dengan *Activity Diagram*, kita dapat menggambarkan bagaimana proses-proses itu dilakukan, urutan langkah-langkah yang diambil, serta keputusan yang dibuat dalam proses tersebut. Ini membantu pengembang untuk memahami secara visual bagaimana sistem beroperasi dan mengidentifikasi area-area yang mungkin memerlukan perbaikan

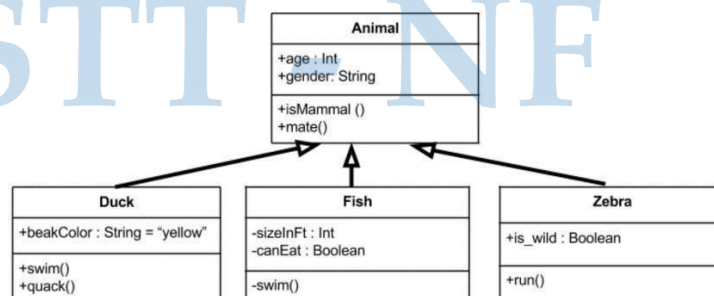
2.1.3 Sequence Diagram



Gambar 2 3 Sequence diagram

Sequence diagram merupakan diagram yang menjelaskan interaksi objek berdasarkan urutan waktu. Sequence dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu, seperti yang tertera pada *Use Case diagram*

2.1.4 Class Diagram

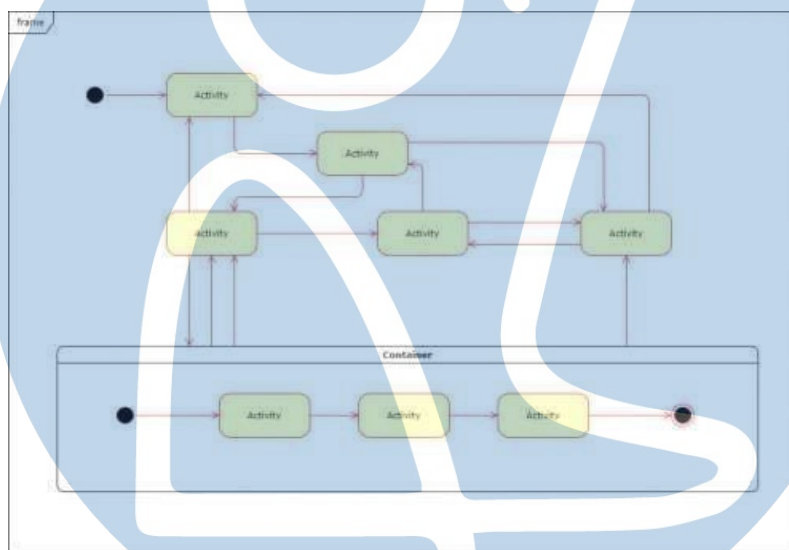


Gambar 2 4Class diagram

Class diagram, juga dikenal sebagai diagram kelas dalam UML, adalah representasi *visual* yang digunakan untuk menggambarkan struktur kelas-kelas dan hubungan antar kelas dalam sebuah sistem perangkat lunak, dengan kelas

kelas direpresentasikan sebagai kotak yang berisi atribut dan metode serta hubungan antar kelas ditunjukkan dengan panah yang menunjukkan arah hubungan dan jenis asosiasi antara kelas-kelas tersebut.

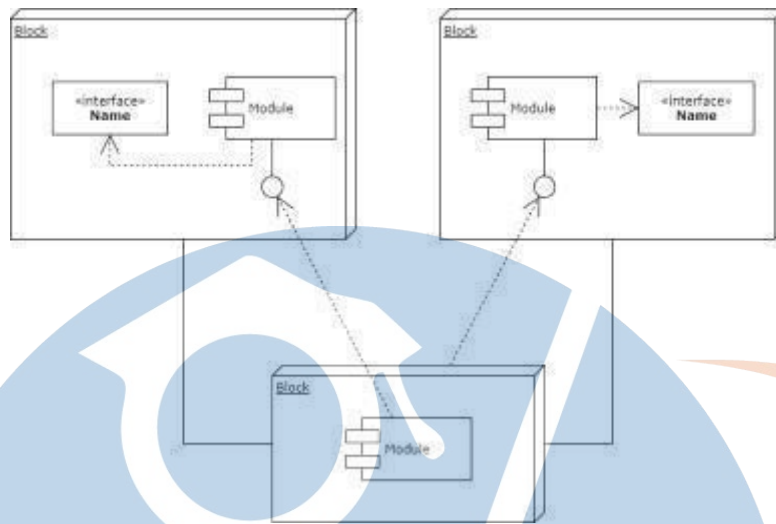
2.1.5 State Machine Diagram



Gambar 2.5 State Machine diagram

Diagram digunakan untuk menggambarkan perilaku sistem sebagai urutan keadaan yang berbeda. Diagram ini menunjukkan bagaimana sistem merespons stimulus eksternal atau perubahan kondisi *internal* dengan beralih dari satu keadaan ke keadaan yang lain

2.1.6 Component Diagram



Gambar 2 6 Component diagram

Component diagram yang berfungsi untuk menggambarkan *software* pada suatu sistem. *Component diagram* merupakan penerapan pada piranti lunak atau *software* dari satu class maupun lebih, dan biasanya berupa *file data*, *source code*, *.exe*, *table*, dokumen, atau yang lainnya

2.6 Metode Pengembangan Aplikasi

1. Rapid Application Development

Rapid Application Development (RAD) merupakan sebuah metode proses pengembangan perangkat lunak yang memprioritaskan penggunaan siklus waktu yang singkat[20]. Dalam pengembangan perangkat lunak, RAD mengadopsi model proses yang menekankan pada siklus pengembangan hidup yang singkat, menjadi sebuah versi adaptasi cepat dari model *waterfall*.

RAD memiliki 3 tahapan sebagai berikut ;

- 1) Rencana Kebutuhan (*Requirement Planning*),
- 2) Proses Desain Sistem (*Design System*),
- 3) Implementasi (*Implementation*).

Requirement planning melibatkan pengguna dalam proses merancang dan membangun sistem, dengan kegiatan yang dilakukan secara berulang-ulang hingga mencapai kesepakatan bersama. Metode ini mampu untuk mengurangi waktu dalam pembangunan perangkat lunak. Model RAD memiliki empat tahap utama, yaitu

- 1) perencanaan kebutuhan,
- 2) Desain sistem,
- 3) Proses pengembangan dan pengumpulan *feedback*,
- 4) Implementasi atau penyelesaian produk.

Dalam implementasi, ada tiga cara implementasi, yaitu

- 1) pengembangan berulang
- 2) *prototyping* sistem
- 3) *prototyping* sekali pakai.

Melalui pengembangan berulang, seluruh proyek dipecah menjadi beberapa seri / versi yang akan dikembangkan secara berurutan.

2.7 Pengujian Aplikasi

2.7.1 *Black Box Testing*

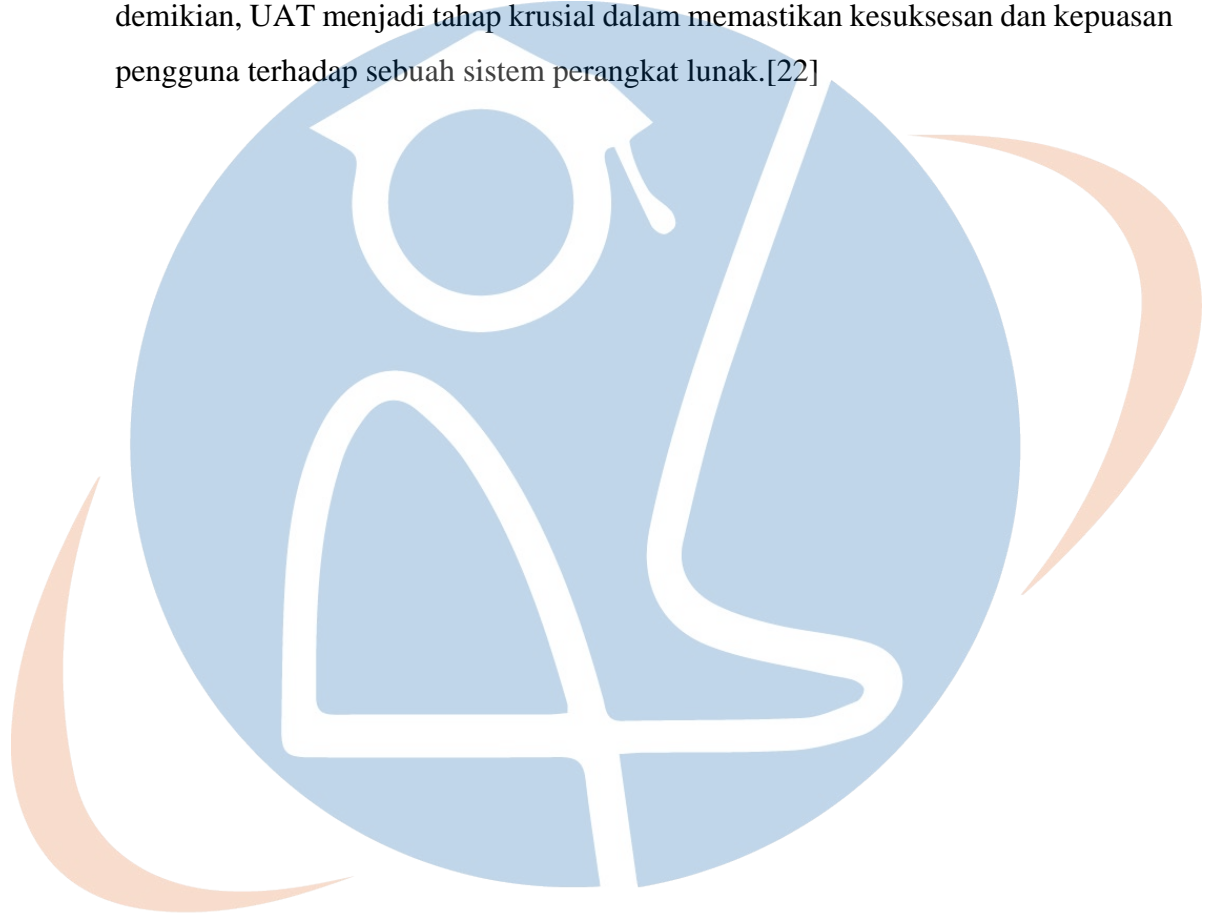
Black box testing adalah pendekatan pengujian perangkat lunak di mana fokus utamanya adalah pada fungsionalitas atau perilaku perangkat lunak dari sudut pandang *user*. Dalam pengujian ini, penguji tidak perlu memiliki pengetahuan tentang struktur kode atau implementasi *internal* dari perangkat lunak. Sebaliknya, pengujian dilakukan berdasarkan spesifikasi fungsional atau kebutuhan pengguna, dan hasil *input* dan *output* dari perangkat lunak diamati untuk menilai kinerja dan keberhasilannya.

Salah satu keuntungan utama dari *black box testing* adalah bahwa pengujian dapat dilakukan oleh siapa saja, tidak hanya oleh pengembang perangkat lunak. Ini memungkinkan untuk melibatkan berbagai pihak yang terlibat dalam proyek, termasuk pengguna akhir atau penguji non-teknis, untuk memastikan bahwa aplikasi memenuhi kebutuhan dan harapan mereka secara efektif. Dengan demikian, *black box testing* menjadi salah satu pendekatan yang penting dalam memastikan kualitas dan kesuksesan sebuah perangkat lunak[21]

2.7.2 *User Acceptance Testing*

User Acceptance Testing (UAT) merupakan tahap terakhir dalam proses pengujian perangkat lunak dan seringkali menjadi langkah terpenting sebelum sistem diimplementasikan secara penuh. Dalam UAT, *user* atau pengguna akhir akan menguji sistem sesuai dengan skenario penggunaan yang telah ditetapkan sebelumnya, untuk memastikan bahwa sistem berfungsi dengan baik dan sesuai dengan harapan mereka. Hasil dari UAT akan memberikan informasi penting kepada pengembang mengenai kesiapan sistem untuk digunakan secara operasional.

Salah satu aspek yang penting dalam UAT adalah keterlibatan langsung dari *user* atau pengguna akhir dalam proses pengujian. Hal ini memastikan bahwa sistem telah memenuhi kebutuhan bisnis dan pengguna secara langsung, sehingga dapat diterima dan digunakan dengan baik oleh mereka. Selain itu, UAT juga seringkali memiliki implikasi finansial, dimana hasil dari pengujian ini dapat memengaruhi proses pembayaran kepada pengembang perangkat lunak. Dengan demikian, UAT menjadi tahap krusial dalam memastikan kesuksesan dan kepuasan pengguna terhadap sebuah sistem perangkat lunak.[22]



STT - NF

2.8 Penelitian Terkait

Tabel 1 Penelitian Terkait

No	Nama dan Tahun	Judul	Tools	Topik	Subjek	Hasil
1	Dwi Kurniawan 2023	<i>Frontend learning management system development using the NextJS Framework</i>	<i>NextJS</i>	<i>Web Development</i>	<i>Students</i>	<i>Learning Management Web Application</i>
2	Abdur Rauf, Agung Tri 2021	Rancang Bangun aplikasi berbasis web sistem informasi <i>repository</i> laporan PKL siswa	<i>PHP, MySQL</i>	<i>Web Development</i>	SMKN 1 Terbanggi Besar	Aplikasi Berbasis Web
3	Zakki Pribadi 2023	Aplikasi Sistem Pengelolaan Nilai Kedisiplinan Siswa Berbasis Web	<i>PHP, CI, MySQL</i>	<i>Web Development</i>	SMAN 1 Banjar Margo	Sistem Berbasis Web

Berdasarkan penelitian yang dilakukan oleh peneliti sebelumnya mengenai rancang bangun aplikasi berbasis web, beberapa penelitian serupa juga telah dilakukan oleh peneliti lain. Berikut ini adalah perbandingan dan analisis lebih mendalam mengenai perbedaan dan persamaan penelitian-penelitian tersebut:

1. Dwi Kurniawan (2023)[5]

- *Tools*: Menggunakan *Framework* NextJS, yang merupakan *Framework* modern berbasis *JavaScript* untuk pengembangan aplikasi *frontend*.
- Topik: Fokus pada pengembangan sistem manajemen pembelajaran berbasis web.
- Hasil: Menghasilkan aplikasi web untuk manajemen pembelajaran yang ditujukan bagi pelajar.
- Perbedaan: Penelitian ini menggunakan *Next JS* yang menawarkan kemampuan rendering *server-side* dan pengoptimalan performa aplikasi. Ini berbeda dari penelitian lainnya yang menggunakan *PHP*.

2. Abdur Rauf, Agung Tri (2021)[23]

- *Tools*: Menggunakan *PHP* dan *MySQL*.
- Topik: Pengembangan sistem informasi *repository* laporan PKL siswa.
- Hasil: Menghasilkan aplikasi berbasis web untuk pengelolaan laporan PKL di SMKN 1 Terbanggi Besar.
- Perbedaan: Penelitian ini menggunakan bahasa pemrograman *PHP* dengan basis data *MySQL*, yang merupakan teknologi yang lebih tradisional dan umum digunakan untuk pengembangan aplikasi web, tetapi tidak menawarkan fleksibilitas modern seperti *JavaScript* dan *NextJS*.

3. Zakki Pribadi (2023)[24]

- *Tools*: Menggunakan *PHP*, *CodeIgniter (CI)*, dan *MySQL*.
- Topik: Pengembangan sistem pengelolaan nilai kedisiplinan siswa berbasis web.

- Hasil: Menghasilkan sistem berbasis web untuk pengelolaan nilai kedisiplinan di SMAN 1 Banjar Margo.
- Perbedaan: Seperti penelitian Abdur Rauf, penelitian ini menggunakan PHP dan
- *Framework* CI untuk pengembangan aplikasi. *Framework* CI memfasilitasi struktur yang lebih terorganisir namun tetap tidak sekompleks dan sefleksibel NextJS dalam menangani aplikasi skala besar dan fitur-fitur modern.

Dari penelitian yang telah dilakukan, hasil yang dicapai oleh masing-masing penelitian menunjukkan adanya kesamaan dalam menghasilkan aplikasi berbasis web yang dapat digunakan untuk mengatasi masalah tertentu dalam lingkup pendidikan dan pengelolaan data. Namun, perbedaan yang paling mencolok terletak pada penggunaan bahasa pemrograman dan kerangka kerja yang digunakan:

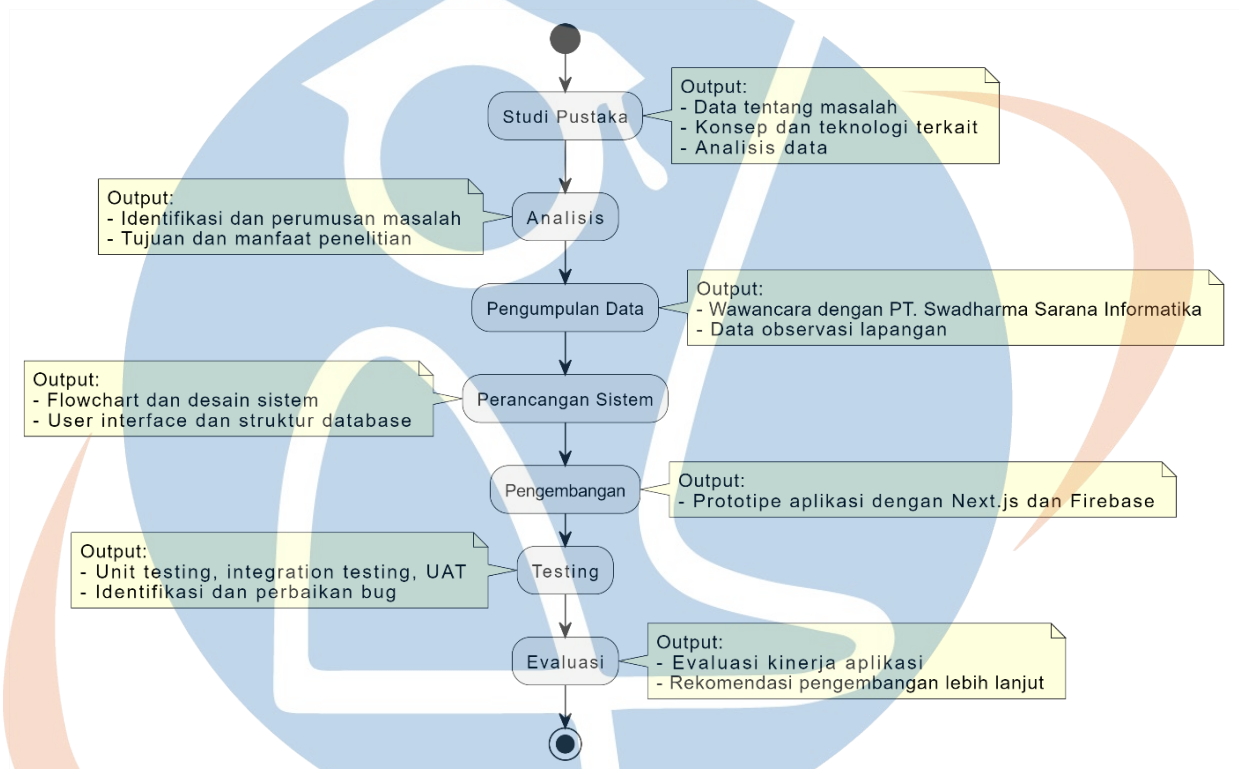
Penelitian ini menggunakan *JavaScript* dengan *Framework Next JS*, yang menawarkan banyak keunggulan seperti *server-side rendering*, optimalisasi performa, dan pengalaman pengembangan yang lebih modern dan efisien. Hal ini berbeda dengan penggunaan *PHP* dan *Framework CI* dalam penelitian lain, yang lebih tradisional dan mungkin kurang fleksibel dalam menghadapi kebutuhan aplikasi web modern yang kompleks.

Dengan demikian, penelitian ini tidak hanya berkontribusi dalam menyediakan solusi praktis melalui aplikasi berbasis web, tetapi juga memperkenalkan penggunaan teknologi modern yang lebih canggih untuk pengembangan aplikasi web di lingkungan pendidikan dan manajemen data operasional.

BAB III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Dibawah ini adalah langkah-langkah yang akan penulis lakukan secara umum untuk melakukan penelitian



Gambar 3.1 Tahapan Penelitian

Berikut adalah penjelasan dari tahap-tahapan yang terdapat pada gambar 7 :

3.1.1 Studi Pustaka

Pada tahap studi Pustaka, data dikumpulkan melalui observasi dan analisis data. Observasi digunakan untuk memperoleh data tentang masalah melalui pengamatan langsung terhadap objek kajian. Peneliti mengamati kondisi lapangan, proses operasional, dan interaksi pengguna dengan sistem yang ada. Analisis data dilakukan untuk mengubah data penelitian menjadi informasi yang berguna, yang kemudian dapat dijadikan dasar untuk mengambil kesimpulan dan merumuskan

solusi yang tepat. Studi pustaka juga mencakup kajian literatur untuk memahami konsep-konsep dan teknologi yang relevan dengan penelitian ini.

3.1.2 Tahapan Perancangan

Pada tahap perancangan, peneliti melakukan identifikasi dan perumusan masalah yang dihadapi oleh PT.Swadharma Sarana Informatika. Peneliti menetapkan batasan masalah untuk memastikan fokus penelitian tetap jelas dan terarah. Selain itu, tujuan penelitian dan manfaat penelitian juga ditetapkan untuk menggambarkan apa yang ingin dicapai dan bagaimana hasil penelitian ini dapat bermanfaat bagi organisasi. Pada tahap ini, peneliti juga membuat *Flowchart* dan desain sistem yang akan dikembangkan, termasuk *user interface* dan struktur database.

3.1.3 Pengembangan

Dalam tahap perencanaan untuk pengembangan aplikasi berbasis web menggunakan metode *Rapid Application Development (RAD)*, peneliti memilih teknologi yang mendukung pengembangan cepat dan fleksibel. Keputusan jatuh pada penggunaan Next.js sebagai *Framework* frontend dan *Firebase* sebagai *Backend as a Service (BaaS)*. Next.js dipilih karena kemampuannya dalam menyediakan pengalaman pengguna yang responsif dan performa yang baik dalam aplikasi web modern. *Firebase* dipilih karena kemudahan dalam menyediakan fitur *Backend* seperti basis data, autentikasi pengguna, dan penyimpanan *file*, yang sesuai dengan kebutuhan pengelolaan data operasional kami.

Langkah selanjutnya adalah menyiapkan lingkungan pengembangan dengan menginstal dan mengkonfigurasi *Next JS* di lingkungan lokal, serta menyiapkan proyek *Firebase* untuk menyimpan data dan mengelola *Backend*. Struktur proyek ditentukan untuk memastikan efisiensi dalam pengembangan iteratif, dengan menyesuaikan folder dan komponen frontend serta mengorganisir koleksi data dan logika bisnis di *Firebase*. Metodologi pengembangan yang dipilih adalah RAD, yang mengutamakan pendekatan iteratif dengan siklus pengembangan singkat (*sprint*), melibatkan pemangku kepentingan dan pengguna akhir untuk

mendapatkan umpan balik yang cepat dan relevan selama proses pengembangan

3.1.4 *Testing*

Tahap pengujian aplikasi adalah langkah penting untuk memastikan bahwa aplikasi yang telah dikembangkan berfungsi sesuai dengan spesifikasi dan kebutuhan yang telah ditentukan. Pengujian meliputi beberapa jenis tes, seperti unit *testing*, integration *testing*, dan *user acceptance testing* (UAT). Pada tahap ini, peneliti mengidentifikasi dan memperbaiki bug atau kesalahan dalam aplikasi, serta memastikan bahwa aplikasi dapat diakses dan digunakan dengan baik oleh pengguna akhir. Feedback dari pengguna juga dikumpulkan untuk perbaikan lebih lanjut.

3.1.5 Kesimpulan

Pada tahap akhir ini, peneliti menyusun kesimpulan atau ringkasan dari seluruh penelitian. Kesimpulan mencakup evaluasi terhadap aplikasi yang telah dibuat, termasuk apakah aplikasi berhasil memenuhi tujuan penelitian dan mengatasi masalah yang telah diidentifikasi. Peneliti juga memberikan rekomendasi untuk pengembangan lebih lanjut dan penggunaan aplikasi di masa mendatang. Selain itu, peneliti mencatat kontribusi penelitian ini terhadap bidang studi terkait dan potensi implementasi di organisasi lain yang memiliki kebutuhan serupa.

Dengan menyelesaikan setiap tahapan penelitian ini, peneliti berharap dapat mengembangkan aplikasi pendataan operasional berbasis web yang efektif dan efisien untuk PT. Swadharma Sarana Informatika

3.2 Rancangan Penelitian

3.2.1 Jenis Penelitian

Pada penelitian ini, penulis memilih jenis penelitian pengembangan (Research and Development, R&D) untuk mengatasi permasalahan di dalam dunia industri. Pendekatan ini dipilih karena memungkinkan peneliti untuk secara sistematis mengembangkan, menguji, dan menyempurnakan produk atau sistem baru. Dalam konteks ini, peneliti akan berfokus pada pengembangan aplikasi berbasis web untuk pengelolaan data operasional, dengan mempertimbangkan kebutuhan dan perspektif para pemangku kepentingan yang terlibat.

Melalui penelitian pengembangan ini, penulis berharap dapat menyediakan solusi yang relevan dan berkelanjutan untuk meningkatkan efektivitas pengelolaan data operasional di PT. SSI. Selain itu, penelitian ini juga diharapkan dapat memberikan kontribusi bagi perkembangan teknologi dan praktik industri secara keseluruhan, dengan menghasilkan aplikasi yang dapat diimplementasikan dan diadaptasi oleh organisasi lain yang memiliki kebutuhan serupa.

3.2.2 Metode Analisis Data

Penelitian ini menggunakan metode analisis data kuantitatif untuk menganalisis dan menginterpretasi data yang dikumpulkan. Analisis data kuantitatif adalah proses penggunaan metode statistik dan teknik matematis untuk memeriksa dan memahami data dalam bentuk angka atau variabel numerik. Pendekatan ini bertujuan untuk menyajikan pola-pola, hubungan, dan tren yang tersembunyi dalam data secara objektif dan terukur. Melalui analisis kuantitatif, peneliti dapat mengidentifikasi asosiasi antara variabel, menguji hipotesis, dan membuat generalisasi yang lebih luas tentang populasi. Hasil dari analisis ini akan diinterpretasikan untuk menentukan efektivitas aplikasi web yang dikembangkan dalam meningkatkan pengelolaan data operasional di PT. Swadharma Sarana Informatika. Prosedur analisis mencakup pengumpulan data dari pengguna

aplikasi, pengolahan data menggunakan perangkat lunak statistik, dan interpretasi hasil berdasarkan hipotesis penelitian.

Dengan demikian, penelitian ini secara eksplisit menggunakan analisis data kuantitatif untuk memberikan wawasan yang terukur dan objektif mengenai kinerja dan efektivitas aplikasi yang dikembangkan.

3.2.3 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah kuesioner. Kuesioner merupakan alat yang digunakan untuk mengumpulkan data dari responden melalui serangkaian pertanyaan yang disusun secara sistematis. Dalam konteks penelitian ini, kuesioner akan dibagikan kepada responden yang terlibat dalam pengelolaan data operasional di PT. SSI.

Kuesioner dirancang untuk memperoleh tanggapan dan persepsi responden tentang penggunaan aplikasi pendataan operasional yang dikembangkan. Kuesioner ini mencakup berbagai aspek, termasuk pengalaman pengguna, preferensi fitur, kemudahan penggunaan, serta evaluasi keseluruhan terhadap aplikasi.

Untuk memastikan *validitas* dan reliabilitas data, kuesioner akan diuji coba terlebih dahulu pada sejumlah kecil responden sebelum dibagikan secara luas. Hal ini bertujuan untuk mengidentifikasi dan memperbaiki potensi masalah atau ketidakjelasan dalam pertanyaan yang diajukan. Setelah kuesioner disempurnakan, distribusi akan dilakukan baik secara daring maupun luring, tergantung pada preferensi dan aksesibilitas responden.

Responden yang menjadi target pengumpulan data yang paling utama adalah para 10 *staff* dan 1 *admin* yang terlibat langsung dalam penggunaan aplikasi pendataan operasional di PT. SSI. Data yang dikumpulkan akan dianalisis secara kuantitatif untuk mengidentifikasi pola, hubungan, dan tren yang muncul dari tanggapan responden. Analisis ini akan membantu dalam mengevaluasi efektivitas aplikasi dan mengidentifikasi area yang memerlukan perbaikan lebih lanjut.

Dengan menggunakan kuesioner sebagai metode pengumpulan data, penelitian ini

diharapkan dapat memperoleh gambaran yang komprehensif tentang kinerja dan penerimaan aplikasi di lingkungan operasional PT. SSI, serta memberikan wawasan yang berguna untuk pengembangan dan penyempurnaan lebih lanjut.

3.2.4 Metode Pengujian

1. *Black Box Testing*

Dalam penelitian ini, metode pengujian yang digunakan adalah *blackbox testing*. *Blackbox testing* merupakan salah satu pendekatan yang umum digunakan dalam pengujian perangkat lunak di mana fokus utamanya adalah pada fungsionalitas aplikasi, tanpa memperhatikan struktur internal atau kode sumber yang mendasarinya. Dengan demikian, peneliti dapat menguji kinerja dan respons aplikasi pendataan operasional yang dikembangkan tanpa harus mengetahui detail implementasi atau alur logika di dalamnya.

Tujuan dari pengujian *blackbox* adalah memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasi yang telah ditentukan dan memberikan respons yang diharapkan terhadap input yang diberikan. Dengan menggunakan pendekatan ini, peneliti dapat mengidentifikasi potensi kesalahan atau bug yang mungkin terjadi dalam aplikasi, sehingga memastikan bahwa aplikasi yang dihasilkan memiliki kualitas yang baik dan dapat digunakan secara optimal oleh pengguna akhir. Dengan demikian, pengujian *blackbox* menjadi langkah penting dalam memastikan keberhasilan aplikasi pendataan operasional yang dikembangkan

2. *User Acceptance Testing*

User Acceptance Testing (UAT), metode pengujian yang digunakan melibatkan pengguna akhir aplikasi sebagai penguji. Fokus utama dari UAT adalah memastikan bahwa aplikasi yang dikembangkan memenuhi kebutuhan dan harapan pengguna dengan baik. Dalam konteks ini, pengguna akan menguji aplikasi secara menyeluruh untuk memvalidasi apakah aplikasi tersebut sesuai dengan standar kualitas yang diharapkan dan apakah fungsionalitas serta kinerjanya memenuhi kepuasan pengguna.

Selama proses UAT, pengguna akan menguji berbagai fitur dan fungsionalitas aplikasi sesuai dengan skenario penggunaan yang telah ditetapkan sebelumnya. Pengguna juga diberikan kesempatan untuk memberikan umpan balik terhadap aplikasi, baik itu mengenai bug atau masalah yang ditemui maupun memberikan saran untuk perbaikan atau peningkatan yang diperlukan. Dengan partisipasi aktif dari pengguna dalam proses UAT, diharapkan bahwa aplikasi dapat diuji secara komprehensif dan memenuhi standar kualitas yang diharapkan oleh pengguna akhir.

3.2.5 Lingkungan Pengembangan

1. Bahasa pemrograman : *JavaScript*
2. *Framework* : *Next JS*
3. *Tools* : *Visual Studio Code*
4. Spesifikasi Komputer : Windows 10, RAM 16GB,SSD 256,
Processor intel core i3 gen 10
5. Lokasi Penelitian : PT.Swadharma Sarana Informatika
Pondok Bambu II
6. Narasumber Penelitian : *Manager Central Operational Unit*

BAB IV IMPLEMENTASI DAN EVALUASI

4.1 Proses Perancangan Sistem

4.1.1 Analisis Masalah

Pada tahap ini, penulis melakukan identifikasi masalah yang dihadapi oleh PT. Swadharma Sarana Informatika terkait dengan pengelolaan data operasional. Hal ini meliputi:

- **Identifikasi Tantangan Operasional:** Memahami kendala atau masalah yang muncul dalam pengumpulan, pengelolaan, dan pemrosesan data operasional.
- **Analisis Kebutuhan:** Mengidentifikasi kebutuhan utama perusahaan dalam mengelola data operasional dengan lebih efisien, efektif serta transparan.

4.1.2 Wawancara dan Observasi

- **Wawancara** Melibatkan berbagai pihak terkait di PT. Swadharma Sarana Informatika untuk memahami perspektif mereka terhadap masalah yang ada dan harapan mereka terhadap solusi yang akan dikembangkan.
- **Observasi Proses Operasional** Mengamati langsung bagaimana data operasional dikumpulkan, diproses, dan digunakan dalam kegiatan sehari-hari perusahaan. Tujuannya adalah untuk mendapatkan pemahaman mendalam tentang proses yang sedang berjalan dan potensi perbaikan yang dapat dilakukan.

Setelah melewati berbagai tahapan, mulai dari analisis masalah hingga melakukan wawancara dan observasi untuk mengidentifikasi kendala yang dihadapi oleh PT. Swadharma Sarana Informatika, kini penulis berada pada tahap yang telah ditunggu-tunggu yaitu tahap implementasi atau perancangan sistem.

Analisis sistem yang berjalan menjadi landasan penting untuk memahami konteks dan lingkungan di mana aplikasi pendataan operasional berbasis web akan dikembangkan. Hal ini memastikan bahwa solusi yang diusulkan tidak hanya mengatasi masalah yang ada, tetapi juga sesuai dengan harapan dan kebutuhan pengguna akhir serta memanfaatkan teknologi yang tepat untuk mencapai tujuan bisnis perusahaan dengan efektif.

Pada bab ini akan dijelaskan secara rinci proses perancangan sistem dan juga evaluasi mengenai hasil pengujian sistem. Proses ini mencakup pengumpulan kebutuhan, pemilihan teknologi, desain arsitektur, pengembangan *Frontend* dan *Backend*, serta integrasi dan pengujian sistem. Setiap tahapan akan diuraikan secara detail untuk memberikan gambaran menyeluruh tentang langkah-langkah yang diambil dalam pengembangan aplikasi pendataan operasional berbasis web dengan *Next JS* dan *Firebase*.

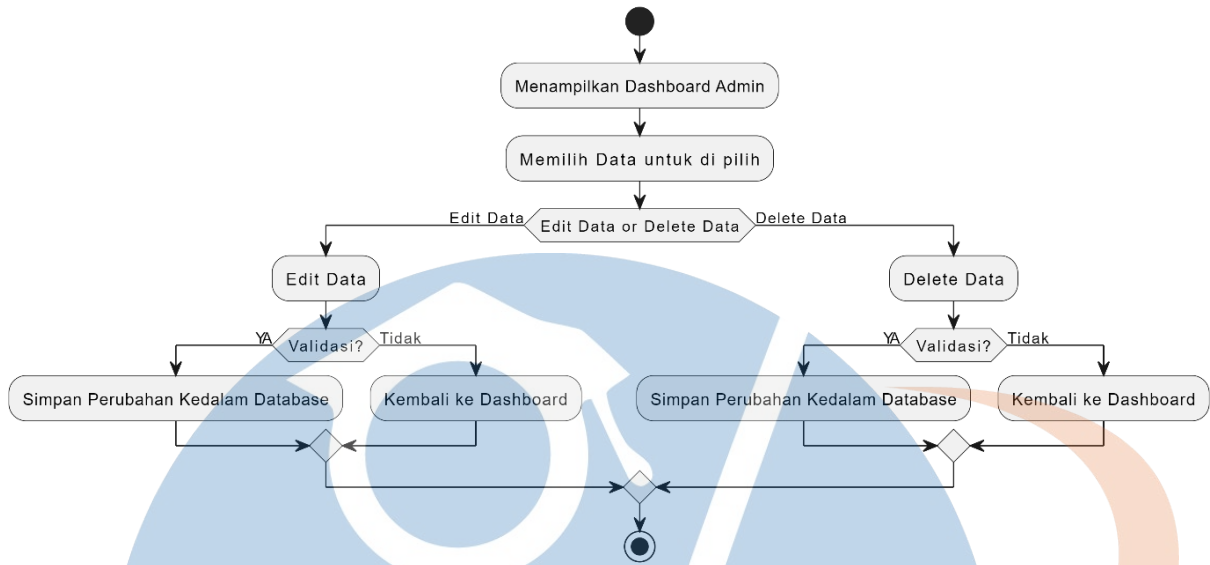
4.2 Flowchart sistem berjalan

Flowchart atau bagan alur adalah diagram yang menunjukkan langkah-langkah dan keputusan dalam sebuah proses program. Setiap langkah ditampilkan dalam bentuk diagram dan dihubungkan dengan garis atau panah.

Flowchart memiliki peran penting dalam menentukan langkah atau fungsionalitas dalam sebuah proyek pembuatan program yang melibatkan banyak orang. Dengan menggunakan bagan alur, proses dari sebuah program menjadi lebih jelas, ringkas, dan mengurangi kemungkinan kesalahan penafsiran. *Flowchart* juga merupakan alat yang efektif untuk menghubungkan kebutuhan teknis dan non-teknis dalam dunia pemrograman[25].

Dapat diambil kesimpulan bahwa flowchart adalah gambaran alur pada aplikasi yang digambarkan menggunakan sebuah diagram. Berikut ini *Flowchart* untuk ke 2 user untuk aplikasi yang di kembangkan :

4.2.1 Flowchart Admin



Gambar 4. 1 Flowchart admin

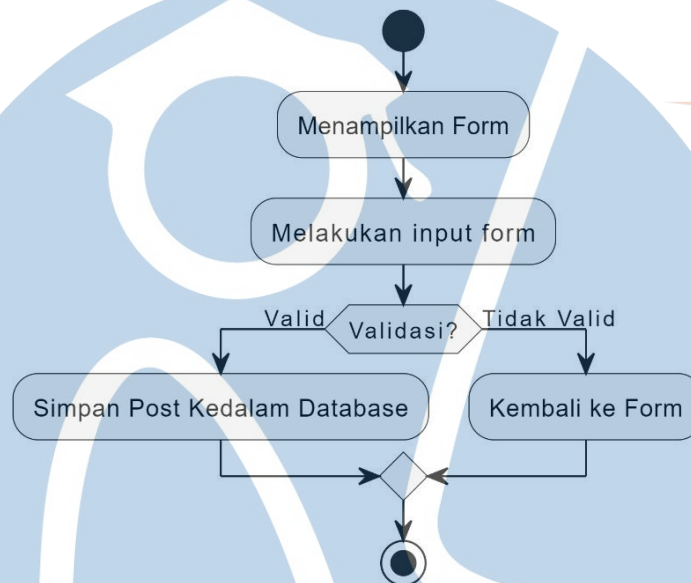
Flowchart dimulai dengan langkah "Start", yang menandai awal dari proses manajemen data oleh *admin*. Setelah memulai, sistem menampilkan "Dashboard Admin", yang merupakan antarmuka utama di mana *admin* dapat melihat dan mengelola data yang telah dimasukkan ke dalam sistem. Dari dashboard ini, *admin* dapat melanjutkan dengan "Memilih Data untuk dipilih", yang berarti *admin* memilih data spesifik yang ingin mereka edit atau hapus.

Setelah data dipilih, *admin* memiliki dua opsi: "Edit Data" atau "Delete Data". Jika *admin* memilih untuk mengedit data, mereka dapat melakukan perubahan yang diperlukan. Jika *admin* memilih untuk menghapus data, data tersebut akan dihapus dari sistem. Kedua tindakan ini kemudian membutuhkan langkah "Validasi?", di mana sistem memeriksa apakah perubahan atau penghapusan data memenuhi kriteria validasi tertentu.

Jika validasi berhasil ("Ya"), langkah berikutnya adalah "Simpan Perubahan ke Dalam Database", di mana perubahan yang telah divalidasi disimpan secara permanen dalam database sistem. Akhirnya, proses berakhir dengan langkah "Stop", menandakan bahwa tindakan *admin* telah selesai.

Flowchart ini menjelaskan alur kerja yang jelas dan terstruktur untuk *admin* dalam mengelola data, memastikan bahwa semua perubahan atau penghapusan data *divalidasi* sebelum disimpan, sehingga menjaga integritas dan akurasi data dalam sistem.

4.2.2 *Flowchart Staff*



Gambar 4. 2 *Flowchart staff*

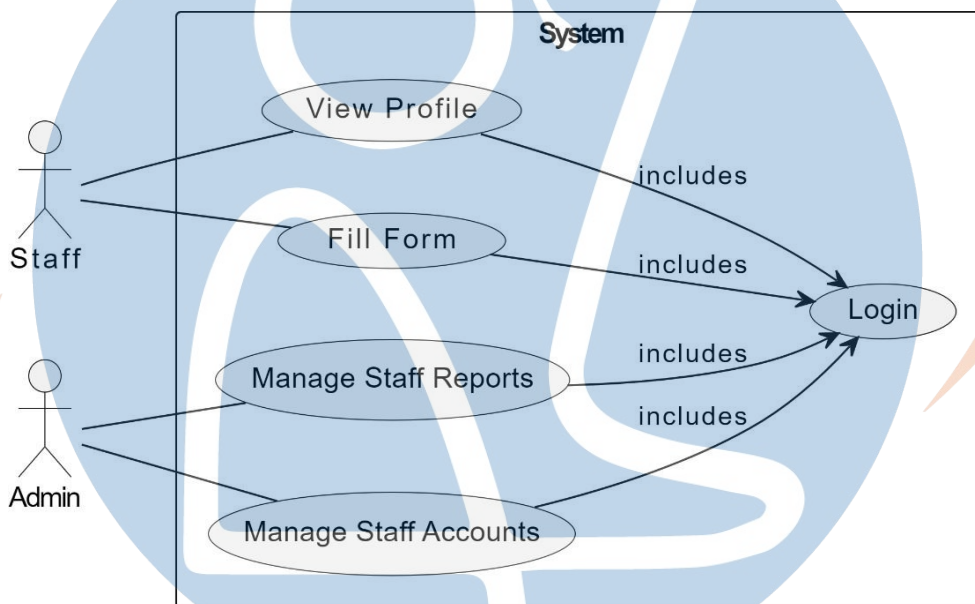
Flowchart dimulai dengan langkah "Start", yang menandai awal dari proses manajemen data oleh *Staff*. Setelah memulai, sistem menampilkan "Formulir untuk diisi oleh *staff*", yang merupakan antarmuka utama di mana *staff* dapat melakukan post data ke dalam sistem. tindakan ini kemudian membutuhkan langkah "Validasi?", di mana sistem memeriksa apakah form sudah memenuhi kriteria *validasi* tertentu.

Jika *validasi* berhasil ("Valid"), langkah berikutnya adalah "Simpan Perubahan ke Dalam Database", di mana perubahan yang telah *divalidasi* disimpan secara permanen dalam database sistem. Akhirnya, proses berakhir dengan langkah "Stop", menandakan bahwa tindakan *admin* telah selesai.

Flowchart ini menjelaskan alur kerja yang jelas dan terstruktur untuk *Staff* dalam melakukan post kedalam aplikasi.

4.3 Use Case Diagram

Use case diagram adalah salah satu jenis diagram dalam UML yang digunakan untuk menggambarkan interaksi antara pengguna (atau sistem lain) dengan sistem yang sedang dirancang. Diagram ini menunjukkan bagaimana berbagai aktor (pengguna atau sistem eksternal) berinteraksi dengan sistem melalui berbagai *use case* (kasus penggunaan), yang mewakili fungsi atau layanan yang disediakan oleh sistem.



Gambar 4. 3 Use case diagram

Aktor: Staf

- Melakukan *Post Data*: *Staff* dapat memposting data baru ke dalam sistem.
- Melihat *Profile*: *Staff* dapat melihat profil mereka di dalam sistem.

Aktor: Admin

- Melakukan *Edit Data*: *Admin* dapat mengedit data yang ada di dalam sistem.
- Melihat *Semua Data*: *Admin* memiliki akses untuk melihat semua data dalam sistem.
- Menghapus *Data*: *Admin* dapat menghapus data dari sistem.

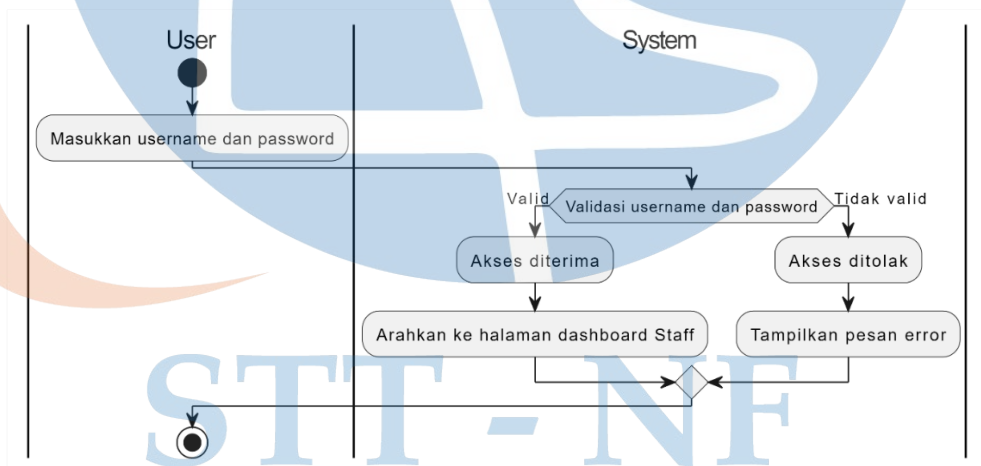
- Membuat Staf Baru: *Admin* dapat menambah staf baru ke dalam sistem.
- Menghapus Staf: *Admin* dapat menghapus akun staf dari sistem.

4.4 Activity Diagram

Activity diagram adalah salah satu jenis diagram dalam UML (*Unified Modeling Language*) yang digunakan untuk memodelkan alur kerja atau aktivitas dalam suatu sistem. Diagram ini menggambarkan langkah-langkah atau aktivitas yang terjadi dalam sebuah proses bisnis atau alur kerja sistem dan bagaimana langkah-langkah tersebut saling berkaitan.

Berikut adalah activity diagram nya :

4.4.1 Activity Diagram Login



Gambar 4. 4 Activity diagram login

User:

- Masukkan *username* dan password: Langkah pertama adalah pengguna (*user*) memasukkan nama pengguna (*username*) dan kata sandi (*password*) mereka ke dalam sistem.

System:

- *Validasi username* dan password: Sistem kemudian memvalidasi nama pengguna dan kata sandi yang dimasukkan oleh pengguna.
- Jika *valid*, maka akses diterima.
- Jika tidak *valid*, akses ditolak.

Valid:

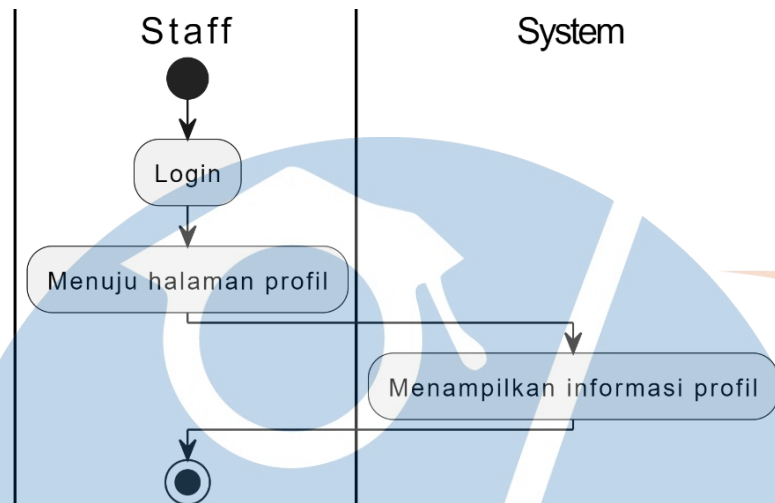
- Akses diterima: Jika *validasi* berhasil, akses pengguna diterima.
- Arahkan ke halaman *dashboard Staff*: Pengguna kemudian diarahkan ke halaman *dashboard* yang sesuai (dalam hal ini, halaman *dashboard Staff*).

Tidak valid:

- Akses ditolak: Jika *validasi* gagal, akses pengguna ditolak.
- Tampilkan pesan error: Sistem akan menampilkan pesan kesalahan kepada pengguna.

STT - NF

4.4.2 Activity Diagram View Profile (staff)



Gambar 4. 5 Activity diagram view profile

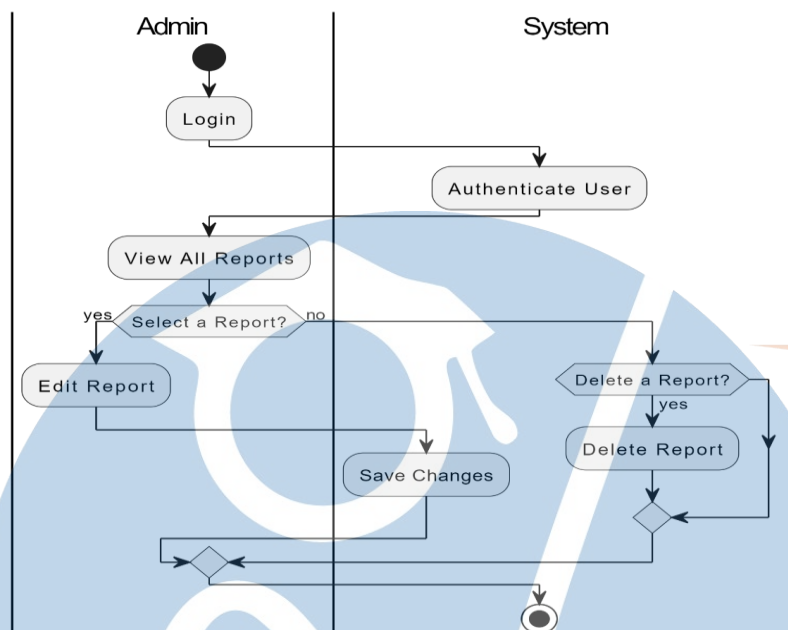
Staff:

- *Login*: Langkah pertama adalah pengguna (*Staff*) melakukan *login* ke dalam sistem.
- *Menuju halaman profil*: Setelah *login* berhasil, pengguna akan menuju halaman profil.

System:

- *Menampilkan informasi profil*: Sistem kemudian menampilkan informasi profil dari pengguna yang bersangkutan.

4.4.3 Activity Diagram Manage Reports

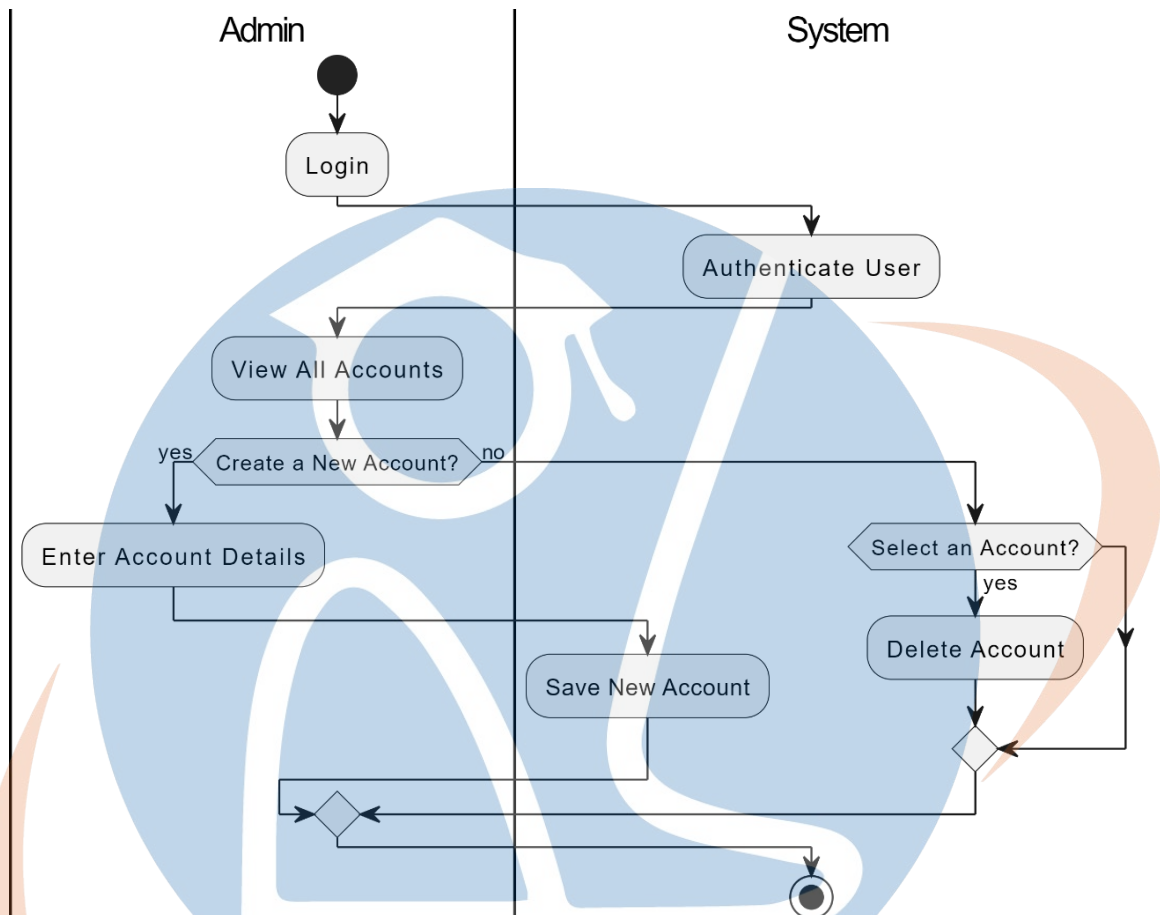


Gambar 4. 6 Activity diagram manage reports

Activity Diagram untuk Mengelola Reports:

- *Admin* melakukan *login (User)*.
- Sistem mengautentikasi pengguna (*System*).
- *Admin* melihat semua laporan (*User*).
- Jika *Admin* memilih sebuah laporan, sistem mengambil detail laporan (*System*).
- *Admin* mengedit laporan (*User*) dan sistem menyimpan perubahan (*System*).
- Jika *Admin* memilih untuk menghapus laporan, sistem menghapus laporan tersebut (*System*).
- Jika tidak ada tindakan lebih lanjut, *Admin* *logout (User)*.

4.4.4 Activity Diagram Manage Account



Gambar 4. 7 Activity diagram manage accounts

Activity Diagram untuk Mengelola Account Staff:

- Admin melakukan login (User).
- Sistem mengautentikasi pengguna (System).
- Admin melihat semua akun staff (User).
- Jika Admin memilih untuk membuat akun baru, Admin memasukkan detail akun (User) dan sistem menyimpan akun baru tersebut (System).
- Jika Admin memilih sebuah akun, sistem menghapus akun tersebut (System).
- Jika tidak ada tindakan lebih lanjut, Admin logout (User).

Keterangan :

- *Login* : *user* memasukkan *username* dan *password*, sistem memvalidasi dan mengarahkan ke valid halaman *dashboard* jika *valid* atau menampilkan pesan error jika tidak *valid*.
- *View Profile (Staff)*: *Staff login*, menuju halaman profil, dan sistem menampilkan informasi profil.
- *Create Post (Staff)*: *Staff login*, mengisi konten post, sistem validasi konten dan menyimpan post jika *valid* atau menampilkan pesan error jika tidak *valid*.
- *Login (Admin)*: *Admin* memasukkan *username* dan *password*, sistem memvalidasi dan mengarahkan ke halaman *dashboard* jika *valid* atau menampilkan pesan error jika tidak *valid*.
- *View Data (Admin)*: *Admin login*, menuju halaman data, dan sistem menampilkan data yang ada.
- *Edit Data (Admin)*: *Admin login*, memilih data untuk diedit, mengedit data, sistem memvalidasi data dan menyimpan perubahan jika *valid* atau menampilkan pesan error jika tidak *valid*.
- *Delete Data (Admin)*: *Admin login*, memilih data untuk dihapus, sistem mengkonfirmasi penghapusan dan menghapus data jika *valid* atau membatalkan penghapusan.
- *Create User (Admin)*: *Admin login*, mengisi data pengguna baru, sistem memvalidasi data dan menyimpan pengguna baru jika *valid* atau menampilkan pesan error jika tidak *valid*.
- *Delete User (Admin)*: *Admin login*, memilih pengguna untuk dihapus, sistem mengkonfirmasi penghapusan dan menghapus pengguna jika *valid* atau membatalkan penghapusan

4.5 Sequence Diagram



Gambar 4. 8 Sequence diagram

Mengirim Data oleh *User*:

- Pengguna yang sudah terautentikasi mengirimkan data melalui form.
- Web browser mengirimkan permintaan post data ke aplikasi *Next.js*.
- Aplikasi *Next.js* menyimpan data tersebut di *Firebase*.
- Hasil penyimpanan dikembalikan ke pengguna sebagai pesan sukses.

Mengedit Data oleh *Admin*:

- *Admin* mengirim permintaan edit data melalui web browser.
- Permintaan diteruskan ke aplikasi *Next.js* yang kemudian mengupdate data di *Firebase*.
- Hasil update dikembalikan ke *admin* sebagai pesan sukses.

Menghapus Data oleh *Admin*:

- *Admin* mengirim permintaan delete data melalui web browser.
- Permintaan diteruskan ke aplikasi *Next.js* yang kemudian menghapus data di *Firebase*.
- Hasil penghapusan dikembalikan ke *admin* sebagai pesan sukses.

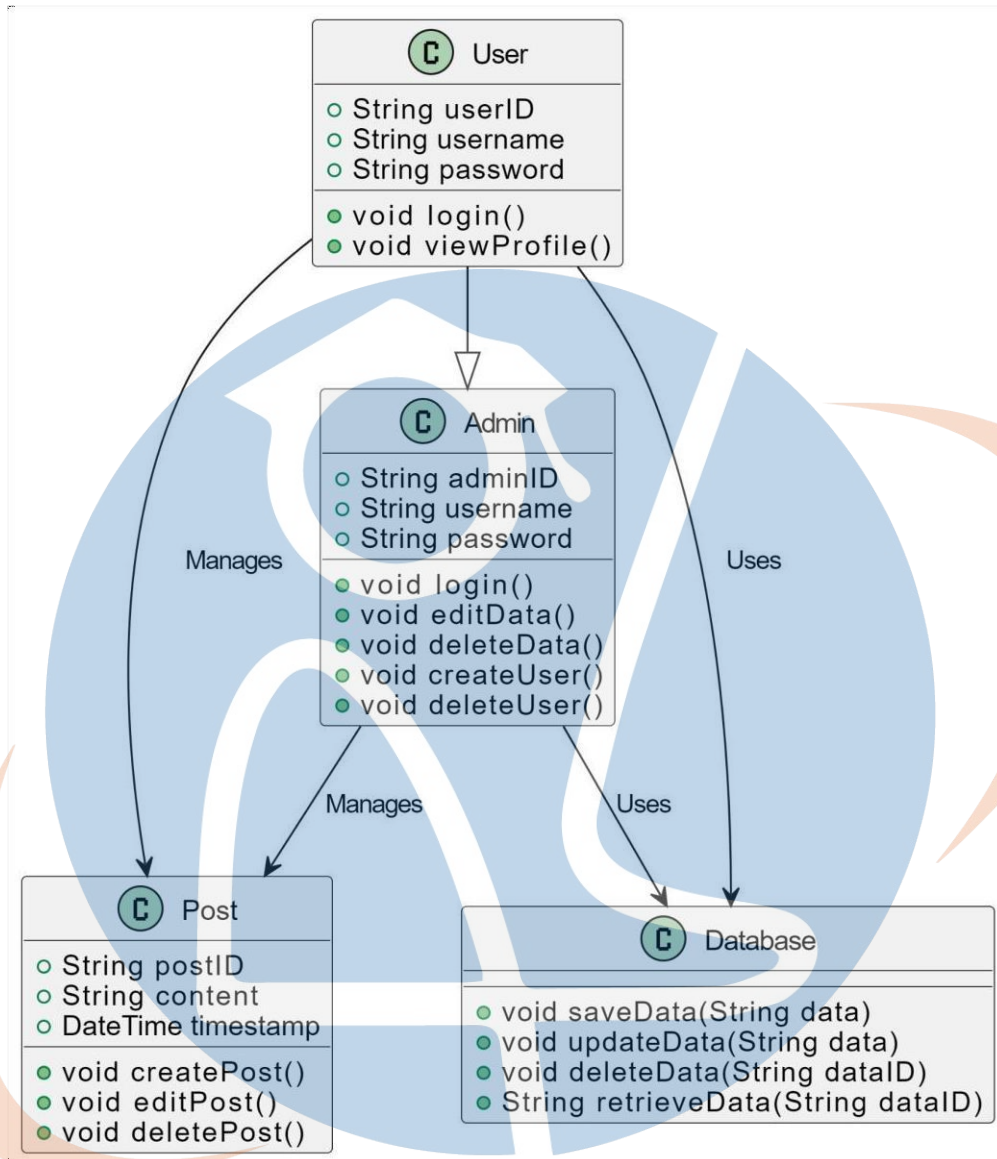
Membuat *User* Baru oleh *Admin*:

- *Admin* mengirim permintaan untuk membuat pengguna baru.
- Aplikasi *Next.js* menyimpan data pengguna baru di *Firebase*.
- Hasil penyimpanan dikembalikan ke *admin* sebagai pesan sukses.

Menghapus *User* oleh *Admin*:

- *Admin* mengirim permintaan untuk menghapus *User*.
- Aplikasi *Next.js* menghapus data pengguna di *Firebase*.
- Hasil penghapusan dikembalikan ke *admin* sebagai pesan sukses.

4.6 Class Diagram



Gambar 4. 9 Class diagram

1. *User:*

- Class ini merepresentasikan pengguna biasa yang memiliki atribut *userID*, *username*, dan *password*.
- Metode yang tersedia adalah *login()* untuk autentikasi pengguna dan *viewProfile()* untuk melihat profil pengguna.

2. **Admin:**

- *Class* ini adalah *subclass* dari *User*, dengan tambahan atribut *adminID*.
- *Admin* memiliki kemampuan tambahan dengan metode seperti *editData()*, *deleteData()*, *createUser()*, dan *deleteUser()* yang tidak dimiliki oleh pengguna biasa

3. **Post:**

- *Class* ini merepresentasikan data postingan dengan atribut *postID*, *content*, dan *timestamp*.
- Metode yang tersedia adalah *createPost()*, *editPost()*, dan *deletePost()* untuk mengelola postingan.

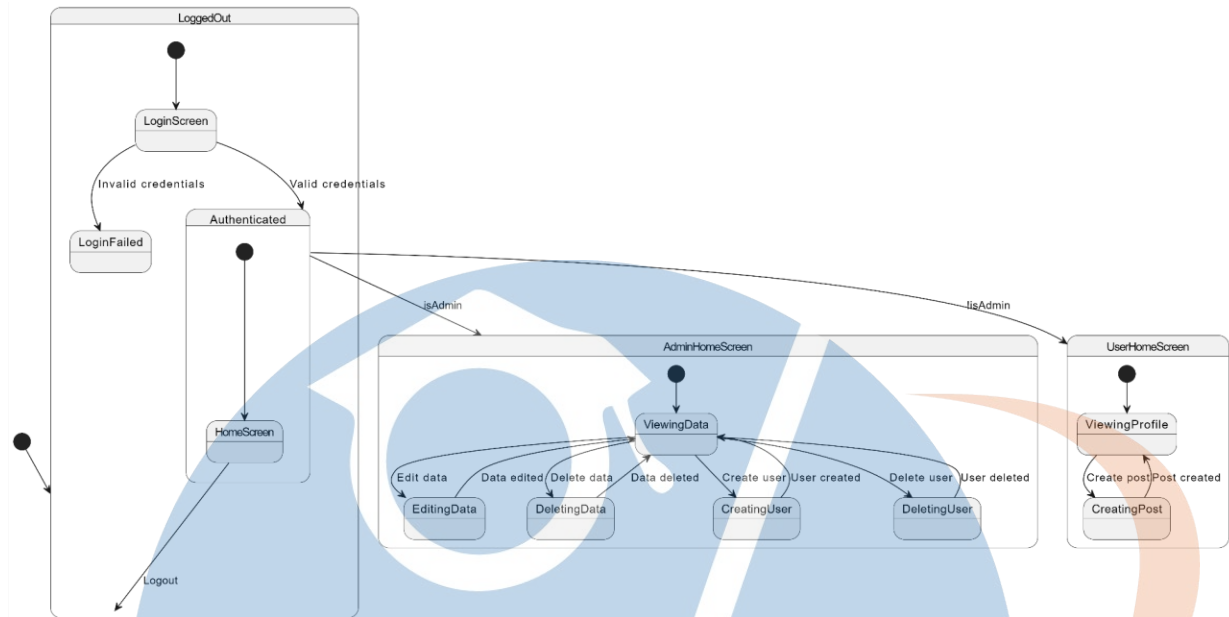
4. **Database:**

- *Class* ini bertanggung jawab untuk operasi data di *Backend* menggunakan *Firebase*.
- Metode yang tersedia adalah *saveData()*, *updateData()*, *deleteData()*, dan *retrieveData()* untuk mengelola data dalam *database*.



STT - NF

4.7 State Machine Diagram



Gambar 4.10 State Machine diagram

1. **LoggedOut:**

- Status awal ketika pengguna belum *login*.
- Transisi ke *Login Screen* di mana pengguna memasukkan kredensial mereka.
- Jika kredensial *valid*, transisi ke **Authenticated**. Jika tidak *valid*, transisi ke *Login Failed*.

2. **Authenticated:**

- Status setelah pengguna berhasil *login*.
- Terdiri dari dua sub-state: *Admin HomeScreen* dan *User Home Screen*.
- Transisi ke **LoggedOut** jika pengguna memilih *logout*.

3. **AdminHomeScreen:**

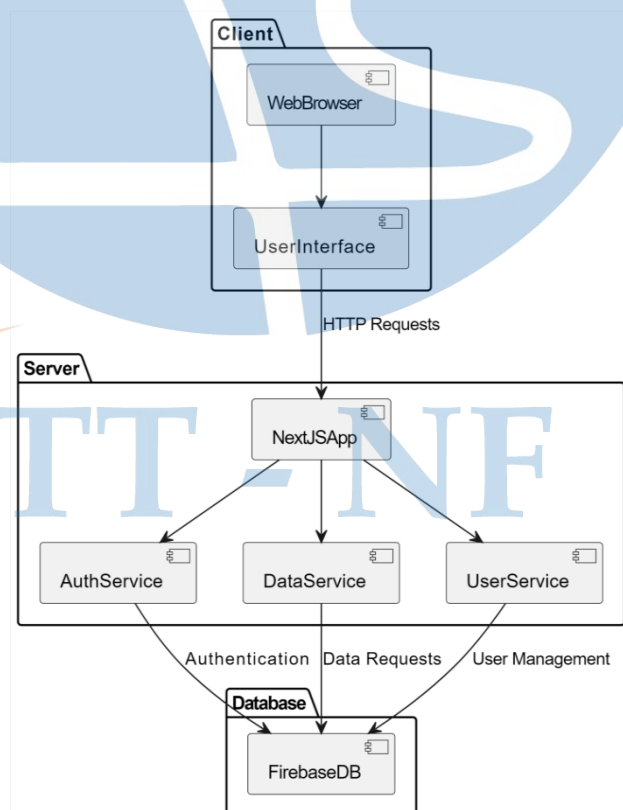
- Setelah *login*, jika pengguna adalah *admin*, sistem transisi ke *Admin HomeScreen*.
- Di dalam *Admin HomeScreen*, *admin* dapat mengedit data, menghapus data, membuat pengguna baru, dan menghapus pengguna.
- Transisi di dalam *Admin HomeScreen* meliputi:

- *Viewing Data* ke *Editing Data* untuk mengedit data.
- *Viewing Data* ke *Deleting Data* untuk menghapus data.
- *Viewing Data* ke *Creating User* untuk membuat pengguna baru.
- *Viewing Data* ke *Deleting User* untuk menghapus pengguna.

4. **UserHomeScreen:**

- Setelah *login*, jika pengguna bukan *admin*, sistem transisi ke *User Home Screen*.
- Di dalam *User Home Screen*, pengguna dapat melihat profil dan membuat posting.
- Transisi di dalam *User HomeScreen* meliputi:
 - *Viewing Profile* ke *CreatingPost* untuk membuat *post*.
 - *Creating Post* kembali ke *Viewing Profile* setelah post dibuat.

4.8 **Component Diagram**



Gambar 4. 11 Component diagram

Aliran Kerja

- *User Interface* di browser pengguna mengirimkan permintaan *HTTP* ke *NextJSApp* di server.
- *NextJSApp* menangani permintaan tersebut dengan bantuan komponen-komponen khusus:
- *AuthService* berkomunikasi dengan *Firebase* untuk autentikasi pengguna.
- *DataService* berinteraksi dengan *FirebaseDB* untuk operasi data seperti *post data*.
- *UserService* berkomunikasi dengan *FirebaseDB* untuk manajemen pengguna seperti membuat atau menghapus pengguna.

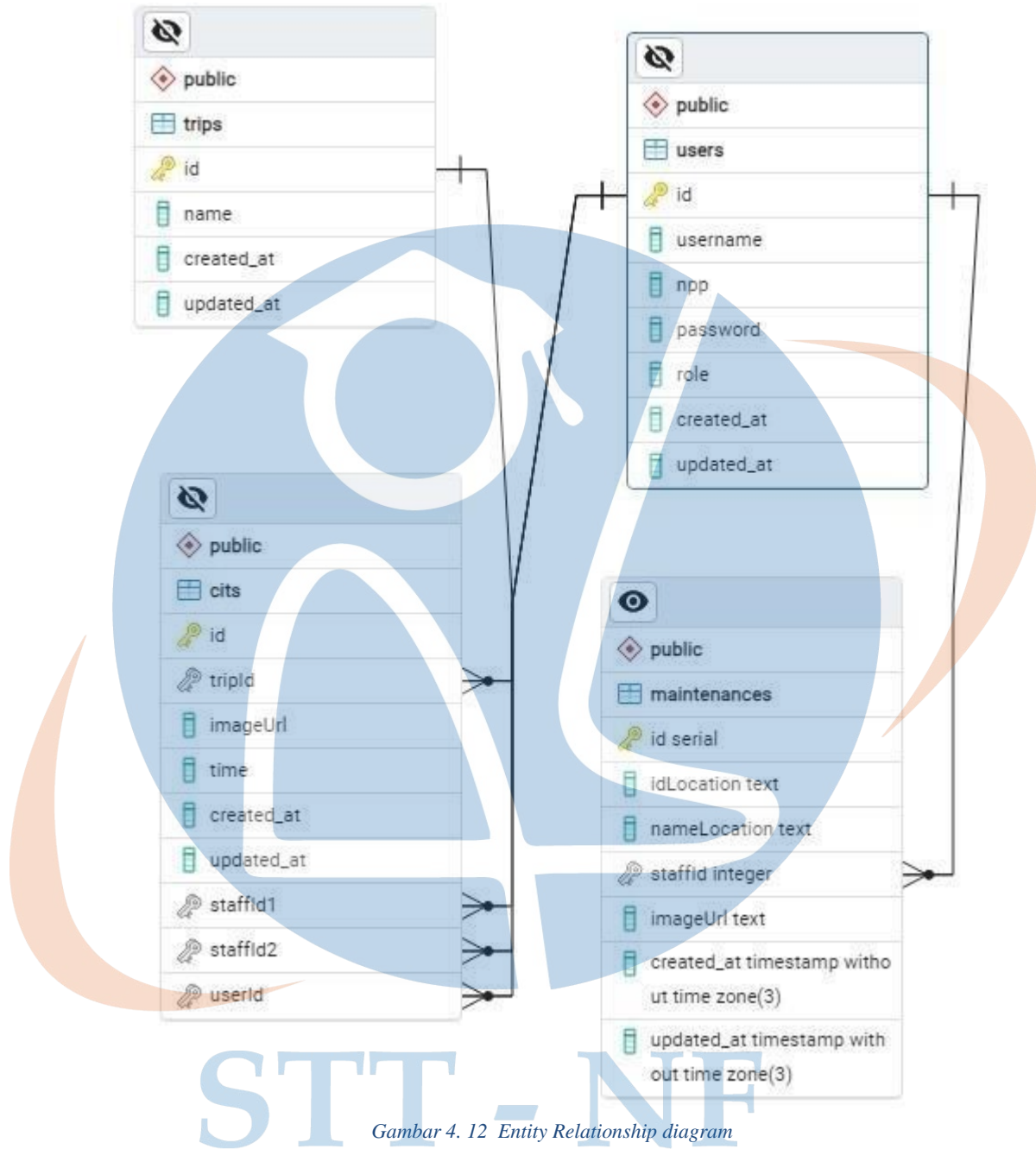
4.9 Development

Melewati berbagai tahapan, mulai dari analisis masalah hingga wawancara dan observasi untuk mengidentifikasi kendala yang dihadapi PT. Swadharma Sarana Informatika, kini penulis berada pada tahap implementasi dan perancangan sistem. Pada tahap ini, penulis menggunakan *Framework Next.js* dan *Firebase* untuk membangun aplikasi pendataan operasional berbasis web.

4.9.1 Persiapan Lingkungan Pengembangan

Langkah pertama adalah menyiapkan lingkungan pengembangan dengan menginstal *Node JS* dan *NPM*, diikuti oleh inisialisasi proyek *Next JS* dan menjalankan *server* pengembangan. Kemudian, peneliti membuat proyek *Firebase* dan menambahkan *Firebase SDK* ke proyek *Next JS* Konfigurasi *Firebase* dilakukan dengan mengatur file *FirebaseConfig.js* dan menyiapkan *Firestore* untuk penyimpanan data.

4.9.2 Entity Relationship Diagram



Gambar 4. 12 Entity Relationship diagram

1. Tabel *users*

- *id*: Primary key, unik untuk setiap pengguna.
- *username*: Nama *staff* atau *admin*.
- *npp*: Nomor pegawai .
- *password*: Kata sandi untuk *login*.

- role: Peran pengguna (misalnya, *admin*, *staff*).
- created_at: Timestamp ketika pengguna dibuat.
- updated_at: Timestamp ketika pengguna diperbarui.

2. Tabel trips

- id: Primary key, unik untuk setiap perjalanan.
- name: Nama perjalanan.
- created_at: Timestamp ketika perjalanan dibuat.
- updated_at: Timestamp ketika perjalanan diperbarui.

3. Tabel cits

- id: Primary key, unik untuk setiap entri CITS.
- tripId: Foreign key yang mengacu ke tabel trips, menunjukkan perjalanan terkait.
- imageUrl: URL gambar terkait entri CITS.
- time: Waktu terkait entri CITS.
- created_at: Timestamp ketika entri dibuat.
- updated_at: Timestamp ketika entri diperbarui.
- staffId1: Foreign key yang mengacu ke tabel *users*, menunjukkan *staff* pertama terkait entri.
- staffId2: Foreign key yang mengacu ke tabel *users*, menunjukkan *staff* kedua terkait entri.
- userId: Foreign key yang mengacu ke tabel *users*, menunjukkan pengguna terkait entri.

4. Tabel maintenances

- id: Primary key, unik untuk setiap entri pemeliharaan.
- idLocation: Id lokasi dilakukannya proses *maintenance*.
- nameLocation: Nama lokasi pemeliharaan.

- *staffId*: Foreign key yang mengacu ke tabel *users*, menunjukkan *staff* terkait entri pemeliharaan.
- *imageUrl*: URL gambar terkait entri pemeliharaan.
- *created_at*: *Timestamp* ketika entri pemeliharaan dibuat.
- *updated_at*: *Timestamp* ketika entri pemeliharaan diperbarui.

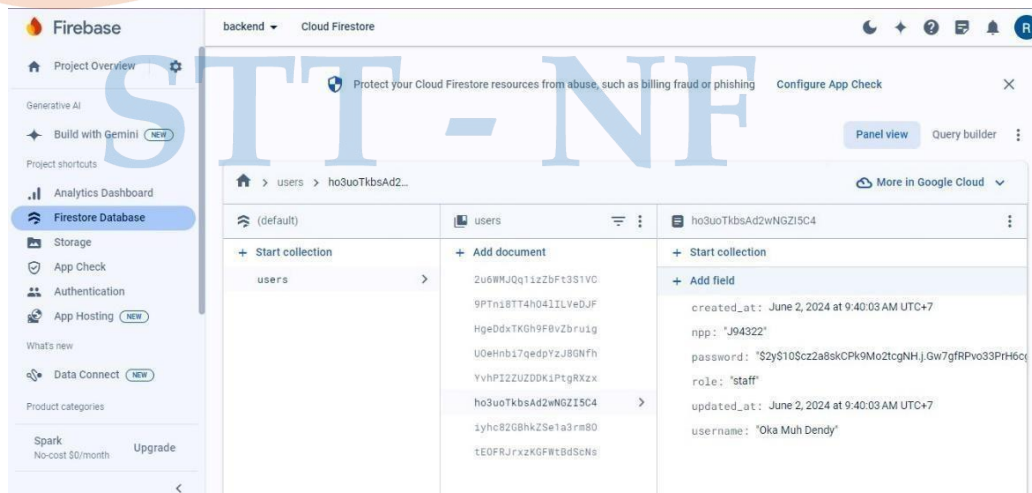
Relasi Antar Tabel

- Tabel *cits* memiliki relasi ke tabel *trips* melalui *tripId*.
- Tabel *cits* juga memiliki relasi ke tabel *users* melalui *staffId1*, *staffId2*, dan *userId*.
- Tabel *maintenances* memiliki relasi ke tabel *users* melalui *staffId*.

ERD ini menunjukkan struktur basis data untuk mengelola pengguna, perjalanan, entri CITS, dan pemeliharaan, serta bagaimana entitas-entitas ini saling berhubungan satu sama lain.

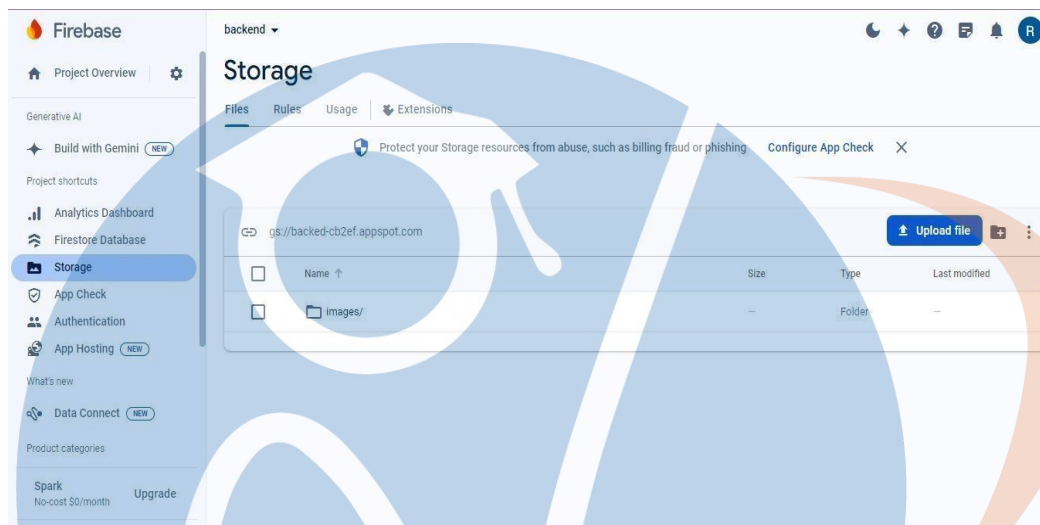
4.9.3 Implementasi *Backend*

Pada implementasi *Backend* ini melibatkan hubungan antara *Frontend Next JS* dan juga *Firebase* sebagai *Backend as a service* Dimana setiap aktivitas yang dilakukan oleh *user* atau *admin* akan disimpan di dalam *Firebase*.



Gambar 4. 13 Firestore databas

Pada gambar diatas penulis menggunakan *Firebase firestore* sebagai penyimpanan seluruh data aplikasi Dimana data tersebut disimpan menjadi beberapa *document* di dalam *Firebase* yang nanti nya dapat di konsumsi oleh *Frontend*



Gambar 4. 14 *Firestore database*

Pada gambar diatas penulis menggunakan *Firebase Storage* sebagai penyimpanan gambar yang dapat gambar tersebut disimpan di dalam *folder images* yang nanti nya akan digunakan oleh *frontend* untuk membuat data tersebut tampil menjadi gambar yang sesuai

4.9.4 Implementasi Frontend

Implementasi frontend melibatkan penerjemahan alur pengguna ke dalam kode menggunakan Next.js. Halaman autentikasi dibuat untuk *login* dan register pengguna dengan menggunakan *Firebase Authentication*. *Dashboard* staf memungkinkan staf memasukkan data yang disimpan di *Firestore*, sementara *dashboard admin* memungkinkan *admin* melihat, mengedit, dan menghapus data yang telah diinput oleh *staff*.



Gambar 4. 15 Halaman Login

Pada halaman ini *staff* dan *admin* dapat melakukan *login* terlebih dahulu agar sistem dapat memvalidasi untuk melanjutkan halaman apa yang akan ditampilkan

CASH IN TRANSIT

Add Data

Staff I:

Select Staff I

Staff II:

Select Staff II

Trip:

Select Trip

Time:

e.g. 13.00 - 15.00

Choose File

Upload File

Add Entry

Gambar 4. 16. Halaman staff cash

MAINTENANCE

Add Data

Staff

Select Staff

Id Location

Select Id Location

Locatin Name

Choose File










No file chosen

Upload File

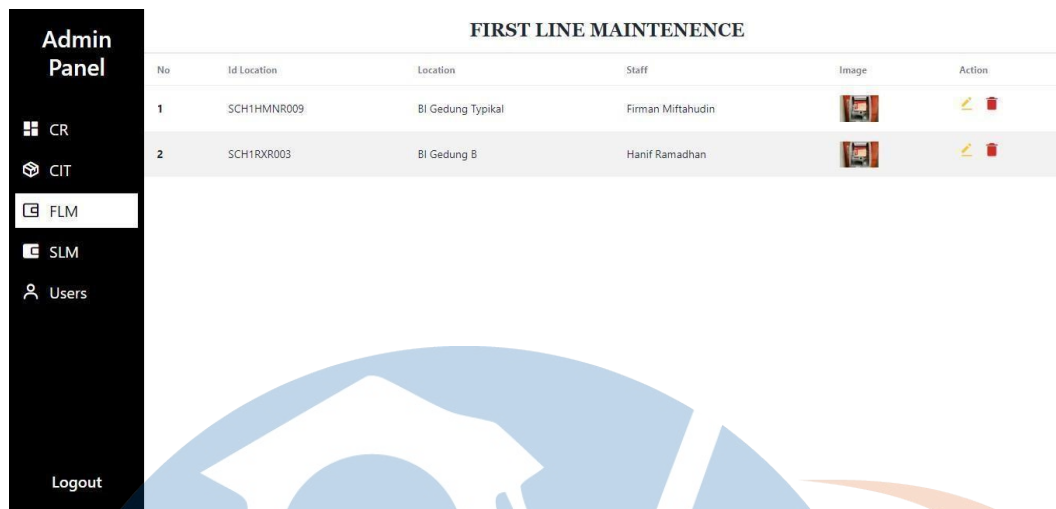
Add Entry

Gambar 4. 17 Halaman staff maintenance

Selanjutnya jika *user* yang *login* mempunyai role staf maka yang akan ditampilkan dalam form ini , dimana di dalam form ini staf dapat melakukan input sesuai instruksi yang telah di terima

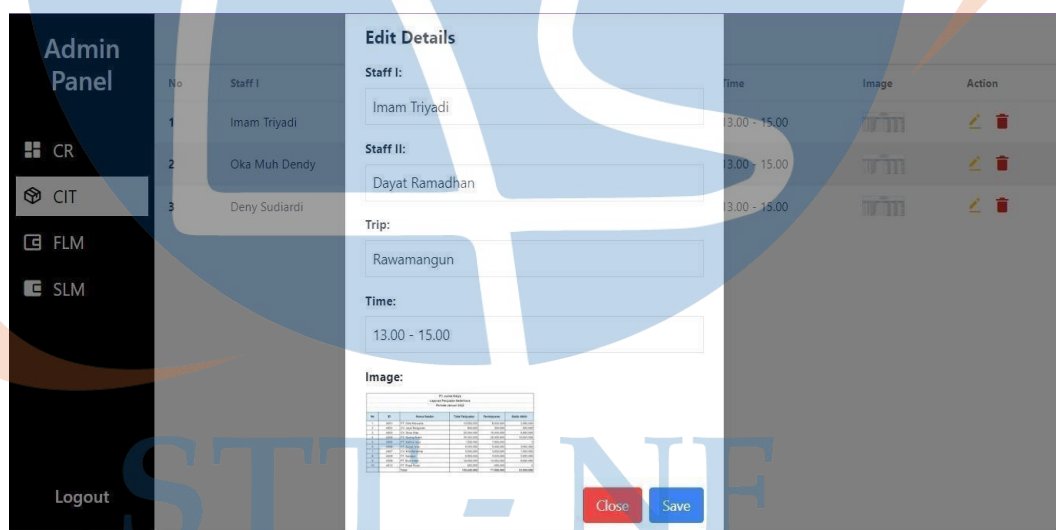
Admin Panel							
CASH IN TRANSIT							
No	Staff I	Staff II	Trip	Time	Image	Action	
1	Imam Triyadi	Dayat Ramadhan	Rawamangun	13.00 - 15.00		 	
2	Oka Muh Dendy	Bachtiar	Jakarta Kota	13.00 - 15.00		 	
3	Deny Sudiardi	Kusnadi	Harmoni 1	13.00 - 15.00		 	

Gambar 4. 18 Halaman Admin Dashboard CIT



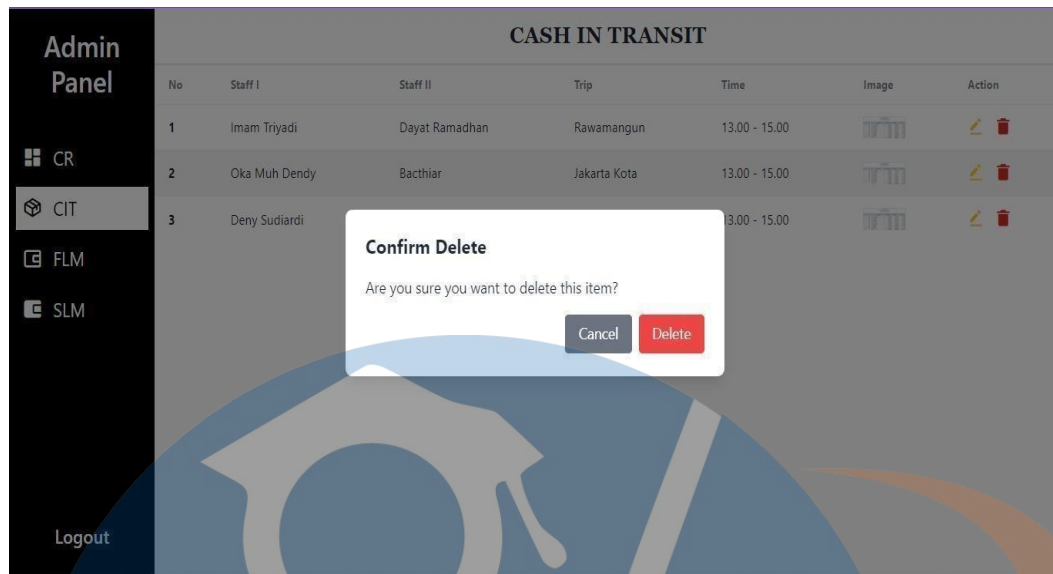
Gambar 4. 19. Halaman Admin Dashboard maintenance

Jika sistem membaca bahwa *user* yang *login* mempunyai role *admin* maka sistem akan memberikan halaman *dashboard admin* ini sebagai halaman utama , disini *admin* dapat melakukan berbagai macam konfigurasi untuk memanipulasi data dari mulai mengubah data, melihat data , dan juga menghapus data jika diperlukan.



Gambar 4. 20 Modal View dan Edit

Jika *admin* ingin mengubah data *admin* dapat menekan action *edit* yang kemudian akan muncul *modal* seperti pada gambar , dimana *admin* dapat mengubah data sesuai yang diinginkan kemudian dapat menyimpan perubahan yang akan di simpan ke dalam *database*



Gambar 4. 21 Modal Delete

Jika *admin* ingin menghapus data *admin* dapat menekan action *delete* yang kemudian akan muncul *modal* seperti pada gambar , dimana *admin* dapat menghapus data sesuai yang diinginkan kemudian dapat menyimpan perubahan yang akan di simpan ke dalam *database*

STT - NF

4.9.5 Pengujian Aplikasi dengan Metode Blackbox

Tabel 2 Pengujian users

No.	Skenario	Input	Output Yang Diharapkan	Langkah-langkah	Kesimpulan
1	<i>Login Staf yang Valid</i>	Nama Pengguna dan Kata Sandi yang <i>valid</i>	Staf berhasil <i>login</i> dan diarahkan ke halaman staf	Masukkan Nama Pengguna dan Kata Sandi yang benar. <i>Staff</i> diarahkan ke halaman setelah <i>login</i>	Berhasil
2	<i>Login Staf yang Tidak Valid</i>	Nama Pengguna dan Kata Sandi yang tidak <i>valid</i>	Sistem menampilkan pesan kesalahan dan tidak membiarkan <i>login</i>	Masukkan Nama Pengguna dan Kata Sandi yang salah. Verifikasi bahwa sistem menampilkan pesan kesalahan yang sesuai Pastikan bahwa staf tidak dapat mengakses halaman staf atau melakukan aktivitas apapun setelah <i>login</i> gagal.	Berhasil
3	Melihat Profil	-	Profil pengguna ditampilkan dengan benar	Akses halaman profil <i>staff</i> . Verifikasi bahwa semua informasi profil <i>staff</i> ditampilkan dengan benar. Pastikan <i>staff</i> tidak dapat mengedit informasi profil secara langsung..	Berhasil
4	Mengisi Form (<i>Valid</i>)	Data yang akan diisi ke dalam form	Form berhasil diisi dan data tersimpan di sistem	Isi form dengan data yang <i>valid</i> Verifikasi bahwa setelah form diisi, data tersimpan dengan benar. Pastikan data yang diisi dapat ditemukan kembali di sistem.	Berhasil

5	Mengisi Form (Tidak <i>Valid</i>)	Data yang tidak <i>valid</i> untuk diisi ke dalam form	Sistem menampilkan pesan kesalahan dan tidak menyimpan data	Isi form dengan data yang tidak <i>valid</i> (misalnya, input kosong atau format yang tidak sesuai). Verifikasi bahwa sistem menampilkan pesan kesalahan yang sesuai. Pastikan data tidak disimpan	Berhasil
---	------------------------------------	--	---	---	----------



STT - NF

Tabel 3 Pengujian Admin

No.	Skenario	Input	Output Yang Diharapkan	Langkah-langkah	Kesimpulan
1	<i>LoginAdmin yang Valid</i>	Nama Pengguna dan Kata Sandi <i>admin</i> yang <i>valid</i>	<i>Admin</i> berhasil <i>login</i> dan diarahkan ke halaman <i>admin</i>	Masukkan Nama Pengguna dan Kata Sandi <i>admin</i> yang benar.	Berhasil
2	<i>LoginAdmin yang Tidak Valid</i>	Nama Pengguna dan Kata Sandi <i>admin</i> yang tidak <i>valid</i>	Sistem menampilkan pesan kesalahan dan tidak membiarkan <i>login</i>	Masukkan Nama Pengguna dan Kata Sandi <i>admin</i> yang salah. Verifikasi bahwa sistem menampilkan pesan kesalahan yang sesuai. Pastikan bahwa <i>admin</i> tidak dapat mengakses halaman <i>admin</i> atau melakukan aktivitas apapun setelah <i>login</i> gagal.	Berhasil
3	Mengedit Data (<i>Valid</i>)	Data form yang akan diedit dan perubahannya	Form berhasil diubah dengan perubahan yang diinginkan	Pilih data yang akan diedit. Masukkan perubahan yang diinginkan pada data tersebut. Verifikasi bahwa data telah diubah dengan benar. Pastikan data form yang diubah tersimpan dengan baik	Berhasil

4	Mengedit Data (Tidak <i>Valid</i>)	Data form yang akan diedit dengan perubahan yang tidak <i>valid</i>	Sistem menampilkan pesan kesalahan dan tidak menyimpan perubahan	<p>Pilih form yang akan diedit.</p> <p>Masukkan perubahan yang tidak <i>valid</i> (misalnya, input kosong atau format yang tidak sesuai).</p> <p>Verifikasi bahwa sistem menampilkan pesan kesalahan yang sesuai</p> <p>Pastikan perubahan yang tidak <i>valid</i> tidak disimpan di sistem.</p>	Berhasil
5	Menghapus Data	Data yang akan dihapus	Data berhasil dihapus dari sistem	<p>Pilih form yang akan dihapus dari sistem.</p> <p>Konfirmasi penghapusan data.</p> <p>Verifikasi bahwa form telah dihapus dengan sukses.</p> <p>Pastikan data tersebut tidak dapat ditemukan lagi di sistem setelah dihapus</p>	Berhasil
6	Menambahkan <i>User Staff</i>	Data <i>user staff</i> baru (nama, kata sandi, dll.)	<i>User</i> baru berhasil ditambahkan ke dalam sistem	<p>Masukkan data <i>user</i> baru seperti nama, kata sandi,</p> <p>Verifikasi bahwa <i>user</i> baru berhasil ditambahkan dengan sukses.</p>	Berhasil

7	Menghapus <i>User Staff</i>	<i>User yang akan dihapus</i>	<i>User berhasil dihapus dari sistem</i>	Pilih pengguna yang akan dihapus dari sistem. Konfirmasi penghapusan pengguna. Verifikasi bahwa pengguna telah dihapus dengan sukses. Pastikan pengguna tersebut tidak dapat <i>login</i> ke sistem lagi setelah dihapus.	Berhasil
---	--------------------------------	-------------------------------	--	--	----------

STT - NF

4.9.6 Pengujian Aplikasi dengan Usability Testing

Pengujian yang dilakukan dengan *Usability Testing* yang akan diuji untuk aplikasi berbasis web ini di Pengujian *usability testing* ini melibatkan 11 responden, yang terdiri dari 10 *user staff* dan 1 *user admin*. Setiap responden diminta untuk memberikan penilaian pada lima aspek yang diuji dengan skala 1-5, dimana 1 berarti sangat tidak puas dan 5 berarti sangat puas.

Tabel 4 Pengujian UAT

NO	Aspek Yang Diuji	Pertanyaan	Rata rata skala 1-5
1	Kemudahan Pengguna	Seberapa Mudah menggunakan aplikasi ini ?	4
2	Tampilan Antar Muka	Seberapa Menarik tampilan aplikasi ini ?	4
3	Fungsionalitas	Seberapa lengkap fitur yang dibutuhkan untuk menjalankan operasional ?	5
4	Kemudahan dalam mengelola data	Seberapa mudah anda mengelola data dengan aplikasi ini ?	4
5	Kepuasan Pengguna	Seberapa puas anda dengan keseluruhan aplikasi ini	4

- Kemudahan Pengguna: Responden menilai seberapa mudah aplikasi ini digunakan dalam aktivitas sehari-hari. Hasil rata-rata menunjukkan nilai 4, menandakan aplikasi cukup mudah digunakan.
- Tampilan Antar Muka: Responden menilai seberapa menarik dan *user-friendly* tampilan aplikasi ini. Hasil rata-rata menunjukkan nilai 4, menunjukkan bahwa antarmuka aplikasi ini cukup menarik dan mudah digunakan.
- Fungsionalitas: Responden menilai kelengkapan fitur yang disediakan

aplikasi untuk mendukung operasional. Hasil rata-rata menunjukkan nilai 5, menandakan bahwa fitur yang tersedia sangat lengkap dan memenuhi kebutuhan operasional.

- Kemudahan dalam Mengelola Data: Responden menilai kemudahan dalam mengelola data melalui aplikasi. Hasil rata-rata menunjukkan nilai 4, menunjukkan bahwa aplikasi ini cukup mudah digunakan untuk pengelolaan data.
- Kepuasan Pengguna: Responden menilai kepuasan secara keseluruhan terhadap aplikasi ini. Hasil rata-rata menunjukkan nilai 4, menunjukkan bahwa pengguna cukup puas dengan aplikasi ini.

Pengujian ini menunjukkan bahwa aplikasi berbasis web yang dikembangkan telah berhasil memenuhi kebutuhan pengguna dan mendapatkan respons positif dari segi kemudahan penggunaan, tampilan antarmuka, fungsionalitas, kemudahan dalam mengelola data, dan kepuasan pengguna secara keseluruhan. Evaluasi ini memberikan dasar yang kuat untuk pengembangan lebih lanjut dan penyempurnaan aplikasi di masa mendatang.

4.9.7 Evaluasi

Berdasarkan hasil pengujian sistem yang telah ditunjukkan pada tabel di atas, hasilnya menunjukkan kinerja yang baik sesuai dengan sistem yang telah dibuat. Evaluasi ini menunjukkan bahwa aplikasi berjalan dengan baik dan sesuai dengan kriteria yang telah ditentukan sebelumnya. Penulis berhasil melakukan implementasi sistem yang telah dirancang menggunakan *Next JS* dan *Firebase*. Evaluasi ini juga mencatat bahwa sistem yang telah diimplementasikan berfungsi dengan baik, memenuhi kebutuhan pengguna, dan menunjukkan stabilitas serta keandalan dalam pengoperasian sehari-hari. Namun, penulis juga menyadari bahwa sistem ini memiliki potensi untuk dikembangkan lebih lanjut lagi, terutama dalam hal peningkatan fitur, optimasi performa, dan integrasi dengan sistem lain yang relevan untuk mendukung operasi PT. Swadharma Sarana Informatika secara keseluruhan.

BAB V KESIMPULAN DAN SARAN

Pada bab ini penulis akan menyajikan Kesimpulan dari seluruh proses penelitian yang telah dilakukan. Selain itu, pada bab ini juga mencakup saran yang dapat digunakan sebagai masukan untuk penelitian selanjutnya.

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengembangan serta pengujian yang telah dilakukan oleh peneliti, dapat disimpulkan beberapa hal sebagai berikut:

1. Aplikasi manajemen pengelolaan data operasional berbasis web dirancang dan diimplementasikan dengan menggunakan *Framework Next JS* sebagai *Development tools* pengembangan web di bagian *Frontend* dan *Firebase* sebagai *Backend as a Service*
2. Berdasarkan hasil pengujian dan evaluasi yang telah dilakukan, dapat disimpulkan bahwa aplikasi manajemen pengelolaan data operasional berbasis web menggunakan *Next JS* dan *Firebase* berjalan dengan baik dan sesuai dengan kebutuhan PT. Swadharma Sarana Informatika. Pengujian fungsional menunjukkan bahwa semua fitur utama, seperti manajemen pengguna, pelacakan data operasional, dan pembuatan laporan, berfungsi dengan baik tanpa adanya *bug* signifikan. Aplikasi mampu menangani berbagai skenario penggunaan yang kompleks dengan tingkat keandalan yang tinggi.

Pengujian kinerja mengindikasikan bahwa aplikasi memiliki responsivitas yang baik dan cepat dalam memproses serta menampilkan data operasional. *Firebase* sebagai *backend* memberikan kinerja yang stabil dan mampu mengelola data *secara real-time*, yang merupakan salah satu kebutuhan utama dalam manajemen data operasional. Integrasi antara *Next JS* dan *Firebase* juga berjalan lancar, memastikan alur data yang efisien dan konsisten.

5.2 Saran

Untuk pengembangan lebih lanjut dan penerapan hasil penelitian ini, beberapa saran yang dapat diberikan untuk peneliti selanjutnya adalah:

1. Peningkatan Fitur Keamanan:

Dalam pengembangan selanjutnya, perlu diperhatikan peningkatan fitur keamanan data untuk melindungi informasi sensitif dari akses yang tidak sah. Implementasi enkripsi data dan autentikasi multi-faktor bisa menjadi pertimbangan.

2. Pengembangan Fitur Analitik:

Menambahkan fitur analitik dan pelaporan yang lebih mendalam dapat membantu manajemen dalam menganalisis data operasional dan membuat keputusan strategis. Fitur ini dapat mencakup *dashboard* visualisasi data dan laporan otomatis.

3. Integrasi dengan Sistem Lain:

Untuk lebih meningkatkan efisiensi operasional, aplikasi ini dapat dikembangkan agar terintegrasi dengan sistem lain yang sudah ada di perusahaan.

STT - NF

DAFTAR PUSTAKA

- [1] A. Infante and R. Mardikaningsih, “The Potential of Social Media as a Means of Online Business Promotion,” 2022.
- [2] R. Andrianto and M. Haris Munandar, “Aplikasi E-Commerce Penjualan Pakaian Berbasis Android menggunakan Firebase Realtime Database,” 2022. [Online]. Available: <http://jurnal.ulb.ac.id/index.php/JCoInT/index>
- [3] N. : Reza and M. Fauzan, “ Skripsi Perancangan Sistem Pemesanan Produk Berbasis Web Pada C.V. Hanif Niagara Group.”
- [4] R. A. Ananda and G. F. Nama, “Analisis dan Perancangan Layanan Streaming Film Berbasis Web Langganan Menggunakan Framework NextJS,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 1, Jan. 2024, doi: 10.23960/jitet.v12i1.3967.
- [5] A. Baehaqi, M. S. Basit, R. E. Indrajit, and R. D. Kurniawan, “Frontend Learning Management System Development Using The NextJS Frameworkk,” *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 4, pp. 899–911, Aug. 2023, doi: 10.52436/1.jutif.2023.4.4.1273.
- [6] S. Mufti Prasetyo, M. Ivan Prayogi Nugroho, R. Lima Putri, and O. Fauzi, “BULLET : Jurnal Multidisiplin Ilmu Pembahasan Mengenai Front-End Web Developer dalam Ruang Lingkup Web Development”, [Online]. Available: <https://journal.mediapublikasi.id/index.php/bullet>
- [7] Y. Gong, K. Chen, F. Gu, and F. Wang, “The design and implementation of a campus web information system based on micro-service architecture,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Sep. 2020. doi: 10.1088/1742-6596/1629/1/012044.
- [8] P. D. Dutonde, “Website Development Technologies: A Review,” *Int J Res Appl Sci Eng Technol*, vol. 10, no. 1, pp. 359–366, Jan. 2022, doi: 10.22214/ijraset.2022.39839.
- [9] Samu Uunonen, “Backend as a service in web development,” 2023.
- [10] Nurul Huda, “Visual Studio Code,” dewaweb.com. Accessed: Apr. 24, 2024. [Online]. Available: <https://www.dewaweb.com/blog/mengenal-visual-studio-code/>
- [11] Alphi, “Postman,” idmetafora.com.
- [12] Aorinka, “Next JS,” dewaweb.com. Accessed: Apr. 28, 2024. [Online]. Available: <https://www.dewaweb.com/blog/framework-next-js/>

- [13] R. Hanafi, A. Haq, and N. Agustin, "Comparison of Web Page Rendering Methods Based on Next.js Framework Using Page Loading Time Test," *Teknika*, vol. 13, no. 1, pp. 102–108, Mar. 2024, doi: 10.34148/teknika.v13i1.769.
- [14] KENNY JINGGA, "tailwind css," binus.co.id. Accessed: Apr. 24, 2024. [Online]. Available: <https://socs.binus.ac.id/2020/11/26/tailwind-css/>
- [15] R. Patnaik, R. Pradhan, S. Rath, C. Mishra, and L. Mohanty, "Study on Google Firebase for Real- Time Web Messaging," in *Smart Innovation, Systems and Technologies*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 461–469. doi: 10.1007/978-981-15-5971-6_50.
- [16] M. Dudjak and G. Martinović, "An API-first methodology for designing a microservice-based backend as a service platform," *Information Technology and Control*, vol. 49, no. 2, pp. 206–223, 2020, doi: 10.5755/j01.itc.49.2.23757.
- [17] C. S. Lwin, N. Khan Dim, S. Mya, and M. Aye, "Firebase As a Backend Service in Mobile and Web Development *," 2021.
- [18] Aulia Reta Faulina, "UML," sekawanmedia.co. Accessed: Apr. 24, 2024. [Online]. Available: <https://www.sekawanmedia.co.id/blog/apa-itu-uml/>
- [19] M. R. Kurniawan, H. Hadiyanto, J. D. P. Zulkarnaen, and C. Harito, "Use Case Diagram for Enhancing Warehouse Performance at PT. MDA Through the Implementation of 5S, Economic Order Quantity, Safety Stock, and Warehouse Management System," *Engineering, MAThematics and Computer Science Journal (EMACS)*, vol. 6, no. 1, pp. 69–78, Jan. 2024, doi: 10.21512/emacsjournal.v6i1.11204.
- [20] M. Tabrani, H. Priyandaru, and S. -, "Application of the Rapid Application Development Method to the Baznas Zakat Receipt Information System in Karawang," *JURNAL TEKNOLOGI DAN OPEN SOURCE*, vol. 4, no. 1, pp. 78–84, Jun. 2021, doi: 10.36378/jtos.v4i1.1365.
- [21] Rony Setiawan, "Black Box," dicoding.com. Accessed: Apr. 24, 2024. [Online]. Available: <https://www.dicoding.com/blog/black-box-testing/>
- [22] Michelle Andriana, "UAT," binus.ac.id. Accessed: Apr. 24, 2024. [Online]. Available: <https://sis.binus.ac.id/2020/10/28/user-acceptance-test/>
- [23] A. Rauf and A. T. Prastowo, "Rancang Bangun Aplikasi Berbasis Web Sistem Informasi Repository Laporan PKL Siswa (Studi Kasus SMK N 1 Terbanggi)," *Jurnal Teknologi dan Sistem Informasi (JTISI)*, vol. 2, no. 3, p. 26, 2021, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/JTISI>

- [24] Z. Pribadi, A. S. Puspaningrum, P. Parjito, M. I. Takaendengan, and N. F. Utomo, "Aplikasi Sistem Pengelolaan Nilai Kedisiplinan Siswa Sman X Berbasis Web," *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 4, no. 2, pp. 192–200, Jun. 2023, doi: 10.33365/jatika.v4i2.2598.
- [25] Rony Setiawan, "Flowchart." Accessed: Jun. 02, 2024. [Online]. Available: <https://www.dicoding.com/blog/flowchart-adalah/>

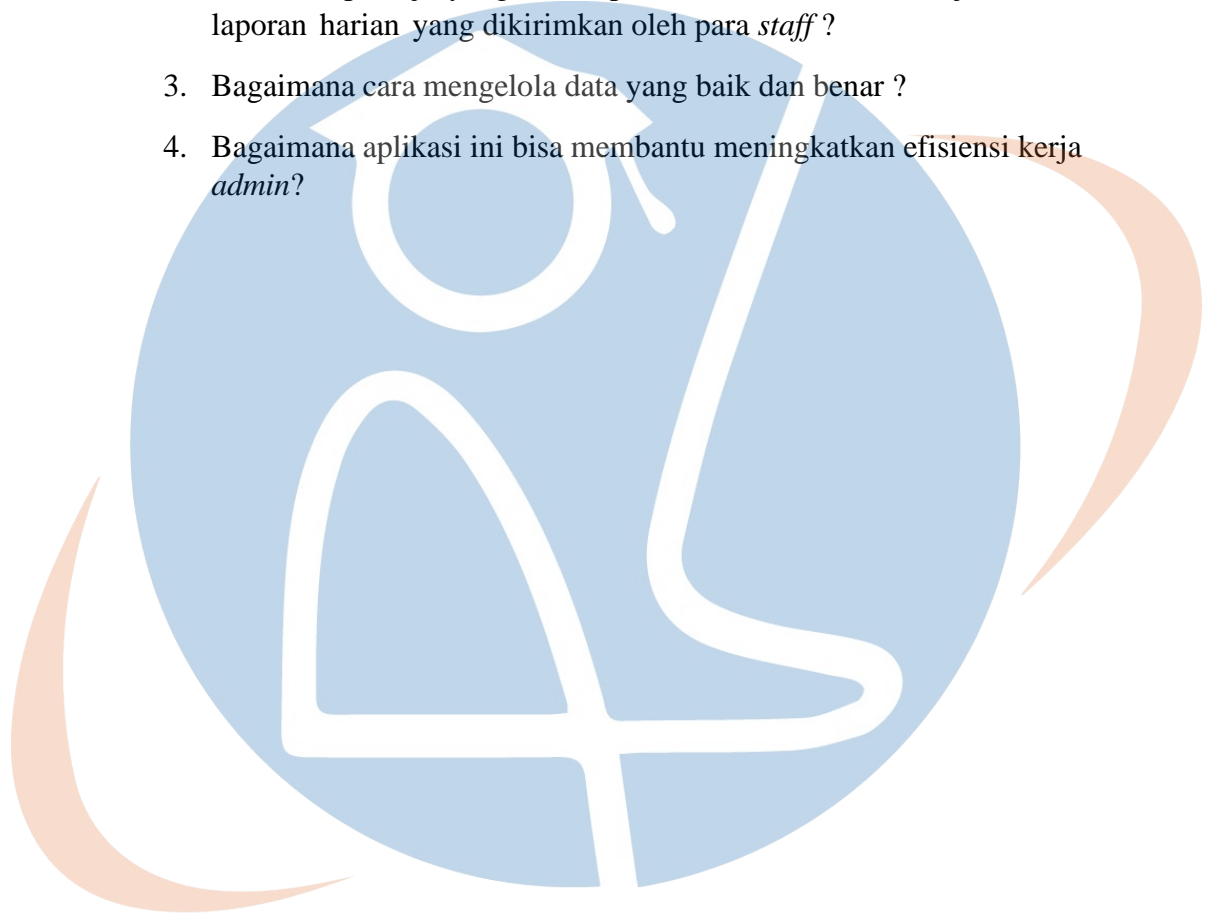


STT - NF

LAMPIRAN

Berisi tentang pertanyaan kepada pihak *staff* dan *admin*

1. Hal apa saja yang dapat membantu jalanya operasional ?
2. Kendala apa saja yang dihadapi untuk melakukan manajemen data laporan harian yang dikirimkan oleh para *staff* ?
3. Bagaimana cara mengelola data yang baik dan benar ?
4. Bagaimana aplikasi ini bisa membantu meningkatkan efisiensi kerja *admin*?



STT - NF