

TUGAS AKHIR - EF234801

**PENGEMBANGAN SISTEM INFORMASI ORGANISASI
NIRLABA BERBASIS *SINGLE PAGE APPLICATIONS*
MENGUNAKAN KERANGKA KERJA VUE.JS: STUDI KASUS
HKBP MANYAR**

AMSAL HERBERT

NRP 5025201182

Dosen Pembimbing

Daniel Oranova Siahaan, S.Kom, M.Sc, P.D.Eng

NIP 197411232006041001

Program Studi S1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024



TUGAS AKHIR - EF234801

**PENGEMBANGAN SISTEM INFORMASI ORGANISASI
NIRLABA BERBASIS *SINGLE PAGE APPLICATIONS*
MENGUNAKAN KERANGKA KERJA VUE.JS: STUDI KASUS
HKBP MANYAR**

AMSAL HERBERT

NRP 5025201182

Dosen Pembimbing

Daniel Oranova Siahaan, S.Kom, M.Sc, P.D.Eng

NIP 197411232006041001

Program Studi S1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024



FINAL PROJECT - EF234801

**DEVELOPING NON-PROFIT ORGANIZATION
INFORMATION SYSTEMS BASED ON SINGLE PAGE
APPLICATIONS USING VUE.JS FRAMEWORK: A CASE
STUDY OF HKBP MANYAR**

AMSAL HERBERT

NRP 5025201182

Advisor I

Daniel Oranova Siahaan. S.Kom, M.Sc, P.D.Eng

NIP 197411232006041001

Study Program Bachelor of Informatics

Department of Informatics

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2024

LEMBAR PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI ORGANISASI NIRLABA BERBASIS SINGLE PAGE APPLICATIONS MENGGUNAKAN KERANGKA KERJA VUE.JS: STUDI KASUS HKBP MANYAR

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Memperoleh gelar Sarjana Komputer pada
Program Studi S-1 Teknik Informatika
Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember


Oleh : AMSAL HERBERT

NRP. 5025201182

Disetujui oleh Tim Penguji Tugas Akhir:

1. Prof. Daniel Oranova Siahaan, S.Kom., M.Sc., P.D.Eng.
2. Dr. Bilqis Amaliah, S.Kom., M.Kom.
3. Siska Arifiani, S.Kom., M.Kom.


Pembimbing

Penguji 

Penguji 

SURABAYA
Juli, 2024

(Halaman ini sengaja dikosongkan)

APPROVAL SHEET

DEVELOPING NON-PROFIT ORGANIZATION INFORMATION SYSTEMS BASED ON SINGLE PAGE APPLICATIONS USING VUE.JS FRAMEWORK: A CASE STUDY OF HKBP MANYAR

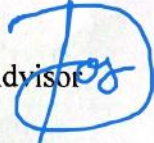


FINAL PROJECT

Submitted to fulfil one of the requirements
for obtaining a degree Bachelor of Computer Science at
Undergraduate Study Program of Informatics
Department of Informatics
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember

By: **AMSAL HERBERT**

NRP. 5025201182

Approved by Final Project Examiner Team:

1. Prof. Daniel Oranova Siahaan, S.Kom., M.Sc., P.D.Eng. Advisor 
2. Dr. Bilqis Amaliah, S.Kom., M.Kom. Examiner 
3. Siska Arifiani, S.Kom., M.Kom. Examiner 

SURABAYA
July, 2024

(Halaman ini sengaja dikosongkan)

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Amsal Herbert/5025201182
Departemen : Teknik Informatika
Dosen Pembimbing / NIP : Prof. Daniel Oranova Siahaan, S.Kom., M.Sc., P.D.Eng./
197411232006041001

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “Pengembangan Sistem Informasi Organisasi Nirlaba Berbasis Single Page Applications Menggunakan Kerangka Kerja Vue.Js: Studi Kasus HKBP Manyar” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima saksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 16 Juli 2024

Mengetahui

Dosen Pembimbing

Mahasiswa



Prof. Daniel Oranova Siahaan, S.Kom., M.Sc., P.D.Eng.

NIP. 197411232006041001



Amsal Herbert

NRP. 5025201182

(Halaman ini sengaja dikosongkan)

STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Amsal Herbert/5025201182
Department : Informatics
Advisor / NIP : Prof. Daniel Oranova Siahaan, S.Kom., M.Sc., P.D.Eng./
197411232006041001

Hereby declare that Final Project with the title of “Developing Non-Profit Organization Information Systems Based On Single Page Applications Using Vue.Js Framework: A Case Study Of HKBP Manyar” is the result of my own work, is original, and is written by following the rules of scientific writing.

If in the future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

Surabaya, 16 July 2024

Acknowledge

Advisor




Prof. Daniel Oranova Siahaan, S.Kom., M.Sc.,

P.D.Eng.

NIP. 197411232006041001

Student



Amsal Herbert

NRP. 5025201182

(Halaman ini sengaja dikosongkan)

**PENGEMBANGAN SISTEM INFORMASI ORGANISASI NIRLABA BERBASIS
SINGLE PAGE APPLICATIONS MENGGUNAKAN KERANGKA KERJA VUE.JS:
STUDI KASUS HKBP MANYAR**

Nama Mahasiswa / NRP : Amsal Herbert / 5025201182
Departemen : Teknik Informatika FTEIC - ITS
Dosen Pembimbing : Prof. Daniel Oranova Siahaan, S.Kom., M.Sc., P.D.Eng.

Abstrak

Sebuah sistem informasi merupakan suatu hal yang penting bagi keberlangsungan suatu organisasi. Perkembangan zaman membuat sistem informasi juga semakin berkembang. Dengan adanya perkembangan aplikasi web, sistem informasi dapat diakses dengan lebih mudah oleh setiap anggota melalui web. Pada organisasi nirlaba Gereja HKBP Manyar penerapan teknologi aplikasi web ini belum diterapkan, saat ini pengelolaan informasi masih dilakukan secara konvensional dan belum memiliki sistem yang mengatur dan menyimpan segala informasi yang ada. Hal tersebut membuat informasi yang ada sulit untuk diakses anggota. Saat ini, terdapat teknologi dalam pengembangan aplikasi web bernama *Single Page Application* (SPA). Istilah SPA merujuk pada aplikasi berbasis web yang hanya menampilkan satu halaman saja. Artinya, meskipun pengguna beralih ke menu lain, URL tidak menunjukkan perubahan. Dengan menggunakan SPA, dapat membuat tampilan web yang lebih dinamis karena proses pengiriman data yang sangat cepat.

Dalam tugas akhir ini dirancang sebuah sistem informasi organisasi yang dapat membantu pengurus organisasi Gereja HKBP Manyar dalam melakukan penyebaran informasi ke anggota. Bagi anggota jemaat, sistem ini dapat membantu mereka dalam melakukan pencarian informasi yang ada di HKBP Manyar. Sistem informasi berbasis web ini bisa diakses oleh semua orang, bagi pengurus perlu melakukan login untuk dapat mengelola data yang ada. Dengan halaman yang sama, tetapi memiliki sedikit perbedaan antarmuka, pengurus dapat menambahkan informasi, memperbarui informasi, dan menghapus informasi yang ada dengan cepat dan reaktif sesuai dengan konsep SPA. Pengembangan sistem dibuat menggunakan kerangka kerja Vue.Js.

Hasil aplikasi web yang dirancang dibandingkan dengan web serupa yang menggunakan metode non-SPA. Didapatkan bahwa dengan menggunakan SPA, dapat meningkatkan kecepatan transfer data dan penghematan sumber daya yang digunakan.

Kata kunci: HKBP Manyar, *Single Page Application*, Sistem Informasi Organisasi Nirlaba, Vue.Js

(Halaman ini sengaja dikosongkan)

DEVELOPING NON-PROFIT ORGANIZATION INFORMATION SYSTEMS BASED ON SINGLE PAGE APPLICATIONS USING VUE.JS FRAMEWORK: A CASE STUDY OF HKBP MANYAR

Student Name / NRP: Amsal Herbert / 5025201182

Department : Informatics FTEIC - ITS

Advisor : Prof. Daniel Oranova Siahaan, S.Kom., M.Sc., P.D.Eng.

Abstract

An information system is an essential element for the sustainability of an organization. The advancement of the times has made information systems also develop. With the development of web applications, information systems can be accessed more easily by each member through the web. In the non-profit organization HKBP Manyar Church, the application of this web application technology has not yet been implemented. Currently, information management is still done conventionally and does not yet have a system that regulates and stores all existing information. This makes the information difficult for members to access. Currently, there is technology in web application development called Single Page Application (SPA). The term SPA refers to a web-based application that displays only one page. This means that even if users switch to another menu, the URL does not show any changes. Using SPA can create a more dynamic web interface because of the very fast data transmission process.

In this final project, an organizational information system is designed to help the administrators of the HKBP Manyar Church organization in disseminating information to members. For congregation members, this system can help them in searching for information available in HKBP Manyar. This web-based information system can be accessed by everyone, but administrators need to log in to manage the existing data. With the same page but slightly different interfaces, administrators can quickly and reactively add, update, and delete information according to the SPA concept. The system development was made using the Vue.js framework.

The resulting web application is compared with similar websites using non-SPA methods. It was found that using SPA can increase data transfer speed and save resources used.

Kata kunci: HKBP Manyar, Non-Profit Organization Information Systems, Single Page Application, Vue.js

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Segala puji dan syukur kehadiran Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Pengembangan Sistem Informasi Organisasi Nirlaba Berbasis *Single Page Applications* Menggunakan Kerangka Kerja Vue.js: Studi Kasus HKBP Manyar”.

Dalam pelaksanaan Tugas Akhir ini tentu penulis sebagai makhluk sosial tidak dapat menyelesaikannya tanpa bantuan dari pihak lain. Tanpa mengurangi rasa hormat, penulis memberikan penghargaan serta ucapan terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya yang telah diberikan selama ini serta senantiasa menemani dan memberi kemudahan kepada penulis dalam menyelesaikan Tugas Akhir.
2. Kedua orang tua dan saudara penulis yang senantiasa memberikan motivasi, semangat, mendukung dalam segala hal untuk penulis.
3. Bapak Daniel Oranova Siahaan selaku dosen pembimbing yang telah membantu penulis untuk menyelesaikan Tugas Akhir ini.
4. Ibu Hotna Pangaribuan selaku bidang kesekretariatan Gereja HKBP Manyar yang telah membantu penulis dalam memberikan masukan kebutuhan web dan memberikan informasi tentang HKBP Manyar.
5. Dosen dan tenaga pendidik di Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember yang telah memberikan pengetahuan, wawasan, dan pengalaman yang sangat berarti selama masa studi.
6. Bang Jonathan Simanjuntak, Bang Sastro Silaban, Bang Maxi Lingga, dan penghuni kos Barak River Side lainnya sebagai teman penulis yang memberikan motivasi dalam mengerjakan Tugas Akhir.
7. Ema, Gress, Merry, Astrid, Bang Jose, Bang Julian, Debora, Elda, Erick, Julian, Kak Angel, Leony, Maya, Renanda, Rogabe, Panser sebagai teman sesama perantauan yang berjuang di kampus perjuangan. Terimakasih karena sudah memaksa penulis untuk mendaftar sidang supaya bisa lulus bersama.
8. Bang James, Immanuel, Rivaldo, Rahel, Andrian, dan David sebagai teman saya sejak masuk di Teknik Informatika ITS dan memberi saya bantuan selama perkuliahan.
9. Teman-teman Naposo HKBP Manyar yang memberikan semangat dan dukungan doa.
10. Teman-teman mahasiswa angkatan 2020 Teknik Informatika, Institut Teknologi Sepuluh Nopember dan pihak-pihak lain yang tidak dapat disebutkan satu persatu.

Penulis sangat berharap bahwa apa yang dihasilkan dari tugas akhir ini bisa memberikan manfaat bagi semua pihak. Tidak ada manusia yang sempurna sekalipun penulis berusaha sebaik mungkin dalam menyelesaikan Tugas Akhir ini. Karena itu, penulis memohon maaf apabila terdapat kesalahan, kekurangan, maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun sangat diharapkan oleh penulis untuk dapat disampaikan untuk perbaikan selanjutnya.

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

LEMBAR PENGESAHAN	v
PERNYATAAN ORISINALITAS	ix
ABSTRAK	xiii
KATA PENGANTAR	xvii
DAFTAR ISI.....	xix
DAFTAR GAMBAR/GRAFIK/DIAGRAM.....	xxiii
DAFTAR TABEL.....	xxv
DAFTAR KODE.....	xxvii
DAFTAR SINGKATAN	xxix
BAB I PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Rumusan Permasalahan.....	2
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terkait.....	5
2.1.1 Implementasi SPA Pada Proses Pengisian KRS	5
2.1.3 Perancangan Sistem Informasi Organisasi Keluarga Mahasiswa.....	5
2.1.4 Rancang Bangun Sistem Informasi Himpunan Mahasiswa	5
2.1.5 Pembuatan <i>Front-end</i> Admin Aplikasi Gerejaku.....	6
2.1.6 Pengembangan Prototipe Aplikasi Bimbingan Tugas Akhir.....	6
2.2 Dasar Teori	6
2.2.1 Sistem Informasi Organisasi.....	6
2.2.2 Spesifikasi Kebutuhan Perangkat Lunak.....	7
2.2.3 HKBP Manyar	8
2.2.4 <i>Single Page Application</i>	8
2.2.5 Vue.Js	10
2.2.6 Manajemen State Pinia	11
2.2.7 Metode Pengembangan Perangkat Lunak	12
2.2.8 Firebase.....	13
2.2.9 <i>Unified Modelling Language</i>	14
2.2.10 Figma.....	14

2.2.11 Wireshark	15
BAB III METODOLOGI	17
3.1 Tahap Kajian Pustaka.....	18
3.2 Tahap Analisis Kebutuhan	18
3.3 Tahap Perancangan Desain	19
3.4 Tahap Implementasi	20
3.4.1 Pembuatan Komponen	20
3.4.2 Penggunaan <i>Routes</i>	21
3.4.3 Penggunaan State Management	22
3.5 Tahap Pengujian.....	24
3.6 Tahap Peluncuran.....	24
3.7 Tahap Penulisan Laporan	25
3.8 Peralatan Pendukung	25
BAB IV HASIL DAN PEMBAHASAN.....	27
4.1 Analisa Kebutuhan	27
4.1.1 Elisitasi Kebutuhan	27
4.1.2 Spesifikasi Kebutuhan Perangkat Lunak.....	30
4.2 Perancangan Desain	36
4.3 Implementasi	40
4.3.1 Pembuatan Satu Halaman HTML	40
4.3.2 Pembuatan Main Js.....	40
4.3.3 Pembuatan App.vue sebagai komponen pertama.....	41
4.3.4 Pembuatan <i>Router</i>	41
4.3.5 Pembuatan Komponen	43
4.3.6 Pembuatan <i>Stores</i>	45
4.3.7 Pembuatan Firebase.....	47
4.4 Tahap Pengujian.....	48
4.5 Tahap Peluncuran.....	51
4.6 Pembahasan.....	53
BAB V KESIMPULAN DAN SARAN	57
5.1 Kesimpulan.....	57
5.2 Saran.....	57
DAFTAR PUSTAKA	59
LAMPIRAN-LAMPIRAN ATAU APPENDIKS	61

A.	Surat Permohonan	61
B.	Spesifikasi Kasus, Diagram Aktivitas, dan Diagram <i>Sequence</i>	62
C.	Rancangan <i>User Acceptance Test</i>	68
D.	Rancangan Antarmuka	71
E.	Komponen <i>Page</i>	77
F.	Dokumentasi Hasil <i>Testing</i>	79
G.	Dokumentasi Pengujian Fungsional Bersama Sekretaris Gereja	82
H.	Perbandingan Tampilan Berita pada Admin yang berbeda	83
	BIODATA PENULIS	85

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR/GRAFIK/DIAGRAM

Gambar 2. 1 Contoh Diagram Kasus Penggunaan (Srinam, 2024).....	8
Gambar 2. 2 Struktur Pengurus Organisasi HKBP Manyar	9
Gambar 2. 3 Perbedaan <i>Lifecycle Multi Page Application</i> dan <i>Single Page Application</i>	9
Gambar 2. 4 Sistem Komponen dalam Vue.js (Dokumentasi Vue.js, 2024)	11
Gambar 2. 5 Diagram Metode Waterfall (Sommerville, 2011).....	12
Gambar 3. 1 Tahapan Pengembangan Berdasarkan Metode <i>Waterfall</i>	17
Gambar 3. 2 Perbedaan Tampilan saat Terjadi Perubahan Tampilan	20
Gambar 4. 1 Use Case Diagram Sistem Informasi HKBP Manyar.....	32
Gambar 4. 2 Diagram Aktivitas Melihat Informasi HKBP Manyar.....	33
Gambar 4. 3 Diagram Sekuens Melihat Informasi HKBP Manyar	34
Gambar 4. 4 Alur Pengujian Performa Web Menggunakan Wireshark	35
Gambar 4. 5 Alur Pengujian Performa Web Menggunakan Developer Tools	36
Gambar 4. 6 Pemodelan Database Sistem Informasi HKBP Manyar	38
Gambar 4. 7 Tampilan Layar Utama <i>Home</i>	39
Gambar 4. 8 <i>Website</i> HKBP Depok Melakukan <i>Reload Page</i>	49

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 4. 1 Hasil Wawancara dengan Sekretaris Gereja HKBP Manyar	28
Tabel 4. 2 Hasil Wawancara dengan Anggota Jemaat HKBP Manyar	29
Tabel 4. 3 Hasil Studi Literatur Sistem Informasi Serupa.....	30
Tabel 4. 4 Kebutuhan Utama.....	30
Tabel 4. 5 Daftar Kebutuhan Fungsional dan Non fungsional HKBP Manyar.....	31
Tabel 4. 6 Fungsi Aplikasi HKBP Manyar.....	32
Tabel 4. 7 Skenario melihat informasi yang ada	33
Tabel 4. 8 Contoh Dokumen User Acceptance Testing (UAT)	34
Tabel 4. 9 Data Koleksi pada Aplikasi Web HKBP Manyar	36
Tabel 4. 10 <i>Traceability Matrix</i> UAT	37
Tabel 4. 11 Spesifikasi Elemen-elemen Layar <i>Home</i>	39
Tabel 4. 12 Komponen pada File <i>Components</i> dan Fungsinya	45
Tabel 4. 13 <i>Store</i> yang Dapat Digunakan.....	45
Tabel 4. 14 Hasil Pengujian dengan Developer Tools pada HKBP Manyar.....	50
Tabel 4. 15 Hasil Pengujian dengan Wireshark pada HKBP Depok.....	50
Tabel 4. 16 Hasil Pengujian dengan Wireshark pada HKBP Manyar.....	50
Tabel 4. 17 Hasil Pengujian dengan Developer Tools pada HKBP Depok	51
Tabel 4. 18 Hasil Pengujian Fungsional Aplikasi	51
Tabel 4. 19 <i>Traceability Matrix</i> Sistem Testing	52
Tabel 4. 20 Perbandingan MPA dan SPA dari Pengujian Wireshark	53
Tabel 4. 21 Perbandingan Pengujian SPA dengan Developer Tools	54
Tabel 4. 22 Perbandingan Pengujian MPA dengan Developer Tools	54

(Halaman ini sengaja dikosongkan)

DAFTAR KODE

Kode 3. 1 Contoh Pembuatan Komponen Pada SPA	21
Kode 3. 2 Contoh Penggunaan Komponen di Komponen lain SPA	21
Kode 3. 3 Pembuatan <i>Routes</i> di MPA	22
Kode 3. 4 Pembuatan <i>Routes</i> di SPA	22
Kode 3. 5 Komponen <i>UserProfil</i>	23
Kode 3. 6 Pembuatan store useUserStore dengan Pinia.....	23
Kode 4. 1 Dokumen HTML	40
Kode 4. 2 Dokumen Main.Js	41
Kode 4. 3 Dokumen App.vue	41
Kode 4. 4 Dokumen Router.....	42
Kode 4. 5 Dokumen BerandaPage.vue.....	43
Kode 4. 6 File CouroselCom	44
Kode 4. 7 File <i>Store</i> Admin.js	46
Kode 4. 8 File Firebase.js	47
Kode 4. 9 <i>Rules</i> Pada Firestore Database	47
Kode 4. 10 <i>Rules</i> Pada Storage.....	48

(Halaman ini sengaja dikosongkan)

DAFTAR SINGKATAN

AJAX	<i>Asynchronous JavaScript and XML</i>
API	<i>Application Programming Interface</i>
MPA	<i>Multi Page Application</i>
HKBP	Huria Kristen Batak Protestan
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
SPA	<i>Single Page Application</i>

(Halaman ini sengaja dikosongkan)

BAB I PENDAHULUAN

1.1 Latar belakang

HKBP Manyar merupakan gereja yang berada di Jalan Manyar Kertoarjo IV No 2, Surabaya, Jawa Timur. HKBP Manyar sudah berdiri sejak 1983, serta tumbuh menjadi gereja yang mempunyai banyak jemaat dengan berbagai komunitas yang beragam di dalamnya. Selain menjadi tempat beribadah, HKBP Manyar juga merupakan suatu organisasi yang di dalamnya terdapat pengurus maupun anggota. Salah satu pelayanan yang diberikan ke anggota HKBP Manyar adalah penyebaran informasi. Penyebaran informasi merupakan pesan yang disebarkan untuk memberikan pengertian tentang pesan yang ingin disampaikan, informasi tersebut disampaikan dengan memberikan fakta yang ada dengan tujuan untuk memberikan pesan yang benar dan jelas. Penyebaran informasi di HKBP Manyar saat ini dilakukan melalui media cetak maupun pesan yang dikirim melalui pesan grup.

Penyampaian informasi merupakan hal yang esensial dalam suatu organisasi. Supaya HKBP Manyar dapat memberikan informasi kepada seluruh anggota dengan efektif, penyebaran informasi harus dapat disimpan dalam satu tempat yang dapat diakses dengan mudah oleh setiap anggota. Pembuatan aplikasi web merupakan salah satu cara untuk penyimpanan informasi tersebut. Di dalam sistem informasi berbasis aplikasi web, setiap anggota organisasi dapat mengakses informasi yang ada dimana saja dan kapan saja. Dari sisi pengurus, pengurus dapat melakukan pembaruan informasi dengan cepat. Sehingga diperlukan adanya pengembangan sebuah sistem informasi web yang sesuai dengan perkembangan zaman.

Pada saat ini teknologi dalam pembuatan aplikasi web sudah banyak mengalami perkembangan. Perkembangan ini mulai dari desain yang semakin menarik sampai kinerja web yang semakin cepat sehingga pengalaman antar pengguna dapat semakin *user-friendly*. Sebelumnya, aplikasi web hanya menggunakan HTML (*Hypertext Markup Language*) sebagai dasar, dan CSS (*Cascading Stylesheet*) untuk mengatur tampilan agar menarik. Keduanya digabungkan untuk menciptakan halaman web yang utuh, dapat diakses dari berbagai perangkat, seperti ponsel atau komputer desktop (Santoso, 2021). Saat ini, aplikasi web tidak hanya ditinjau dari sisi desain nya saja, tetapi juga kecepatan pengiriman data. Pengoptimalan pengiriman data sangat penting bagi pengguna karena dengan pengiriman data yang cepat pengguna tidak perlu membutuhkan waktu yang lama untuk menunggu halaman web terbuka.

Pengiriman data pada aplikasi web mengalami perubahan dari waktu ke waktu. Awalnya pembuatan aplikasi web menggunakan cara tradisional atau biasa dikenal dengan *Multi Page Application* (MPA) dengan server sebagai pusatnya. MPA *me-render* halaman web ketika menerima permintaan HTTP, protokol jaringan, dari *browser*. Halaman yang di-*render* dikirim kembali sebagai respons HTTP dan memicu *refresh* halaman web di *browser* sehingga dapat membuka halaman baru. Dalam aplikasi web tradisional, ketika pengguna berinteraksi dengan halaman web, dia dapat memicu permintaan HTTP berulang kali, yang ditangani oleh server. Misalnya, jika pengguna mengklik tombol di halaman web, klik tersebut akan memicu permintaan, yang ditangani oleh server. Nantinya, pengguna akan mendapatkan *refresh* halaman web di *browser* dan hasilnya interaksi tombol ditampilkan. Jika pengguna kembali ke navigasi yang sama setelah pergi ke navigasi lain, maka akan mengirimkan permintaan dan server mengirimkan HTTP yang sama untuk kedua kalinya. Hal ini menyebabkan server melakukan pekerjaan yang sama dua kali dan sebagai pengguna harus mengunduh *resource* yang sama dua kali. Tentu saja membuat pengguna menunggu lebih lama, ini merupakan pengalaman pengguna yang buruk. Aspek lain dalam MPA adalah penyimpanan status dan pengelolaan data. Status pengguna dan status aplikasi dipertahankan terutama di sisi server, menggunakan mekanisme seperti status sesi atau aplikasi, dan data juga disimpan di sisi server. Hal ini menimbulkan masalah lain, yaitu kebutuhan untuk menanyakan apapun ke server

keadaan waktu atau data diperlukan (Fink & Flatow, 2014).

Untuk menjawab permasalahan yang ada pada pengembangan aplikasi web dengan teknik MPA adalah dengan menciptakan satu halaman web sebagai tampilan utama aplikasi. Dalam pendekatan ini, halaman tidak perlu dimuat ulang sepenuhnya setiap kali pengguna berinteraksi. Ini dikenal sebagai konsep *Single Page Application* (SPA), teknologi yang bekerja di dalam *browser* yang tidak membutuhkan *reload page* saat digunakan. *Reload page* akan digantikan dengan HTML yang isinya diganti dari satu tampilan ke tampilan lainnya. Dengan kata lain, pengguna tidak akan berpindah halaman dengan melakukan request kepada server setiap kali terjadi interaksi pada aplikasi (Wibowo & Wiguna, 2019). SPA adalah aplikasi web yang memuat satu halaman HTML dan secara dinamis memperbarui halaman tersebut berdasarkan interaksi pengguna. Server hanya perlu mengirim dan memperbarui bagian tertentu dari halaman sesuai dengan permintaan pengguna, sehingga menghemat data untuk mengunduh *resource*. *Browser* dapat mengubah tampilan di bagian tertentu saja secara dinamis, menjadikan SPA sangat responsif dan memberikan pengalaman yang nyaman bagi pengguna. Sebagian besar logika bisnis dalam SPA diimplementasikan di sisi klien, sedangkan server berfungsi sebagai API untuk keperluan seperti autentikasi, validasi, atau penyimpanan data ke database. Karena logika dijalankan di klien, SPA tidak memerlukan *refresh* halaman, sehingga tidak perlu menunggu logika server selesai. Semua faktor ini menjadikan SPA sebagai konsep yang ideal untuk membangun aplikasi web modern (Fink & Flatow, 2014).

Tugas akhir ini merancang proyek pengembangan web SPA untuk sistem informasi HKBP Manyar dan membandingkan aplikasi tersebut dengan sistem informasi gereja lain yang menggunakan metode tradisional. Objek penelitian adalah sistem informasi HKBP Manyar yang diakses oleh anggota aktif HKBP Manyar. Perbandingan dilakukan dari segi kecepatan akses dan penggunaan *resource* yang diunduh. Hasil penelitian menunjukkan bahwa aplikasi SPA dapat membantu mengurangi waktu yang diperlukan untuk menampilkan informasi pada halaman web dengan melakukan penghematan *resource* dan mempercepat waktu akses terhadap sistem. Harapannya, dengan adanya sistem informasi berbasis web ini, anggota HKBP Manyar dapat mempunyai sistem informasi yang efektif dan efisien.

1.2 Rumusan Permasalahan

Rumusan masalah yang terdapat dalam latar belakang di atas sebagai berikut:

1. Bagaimana mengembangkan sistem informasi berbasis aplikasi web yang bisa menjadi wadah untuk penyebaran informasi yang ada di HKBP Manyar?
2. Bagaimana perbandingan kecepatan akses dan penggunaan *resource* dari web *Single Page Application* dengan *Multi Page Application* pada aplikasi web yang serupa?

1.3 Batasan Masalah

Adapun dalam menyelesaikan permasalahan ini, ada beberapa batasan, yaitu:

1. Aplikasi web yang dikembangkan menggunakan kerangka kerja Vue.
2. Dalam pembuatan sistem informasi HKBP Manyar informasi yang bisa diakses berupa informasi ibadah, struktur organisasi, informasi kegiatan dan pengumuman.

1.4 Tujuan

Adapun tujuan dari tugas akhir ini adalah untuk membuat sistem informasi HKBP Manyar dengan menggunakan konsep *Single Page Application* yang memiliki kecepatan akses yang lebih cepat dan penggunaan *resource* yang lebih kecil dibanding dengan *Multi Page Application*.

1.5 Manfaat

Adapun manfaat dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Membantu pengurus organisasi HKBP Manyar untuk melakukan penyebaran informasi kepada jemaat dengan efisien.
2. Membantu jemaat HKBP Manyar untuk mendapatkan informasi tentang organisasi yang ada di HKBP Manyar.
3. Aplikasi web ini dapat menjadi awal untuk pengembangan sistem informasi berbasis web HKBP Manyar sehingga .

(Halaman ini sengaja dikosongkan)

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan teori-teori yang berkaitan dengan pembangunan sistem informasi berbasis web pada HKBP Manyar yang diajukan untuk tugas akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap perangkat lunak yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Penelitian Terkait

Beberapa penelitian yang telah dilakukan sebelumnya pada topik SPA dijelaskan secara rinci dalam sub bab 2.1.1 dan 2.1.2. Sementara penelitian terkait sistem informasi organisasi dijelaskan dalam sub bab 2.1.3 dan 2.1.4. Selanjutnya, penelitian terkait implementasi Vue dijelaskan dalam sub bab 2.1.5 dan 2.1.6.

2.1.1 Implementasi SPA Pada Proses Pengisian KRS

Di penelitian ini dilakukan pembuatan sistem pengisian data rencana studi di Universitas Muhammadiyah Surakarta (UMS) menggunakan konsep SPA. Pengujian kinerja SPA dilakukan dengan membandingkan jumlah data dan waktu transfer yang dibutuhkan dengan sistem Kartu Rencana Studi (KRS) *online* yang sedang berjalan. Penulis menemukan bahwa SPA dapat memperbaiki kinerja sistem dengan mengurangi *bandwidth* yang dibutuhkan dan mengurangi waktu total proses KRS (Ristyabudi & Thamrin, 2016).

2.1.2 Penerapan SPA Pada Aplikasi Lelang Barang

Tujuan penelitian ini merancang aplikasi lelang barang *secondhand* berbasis web dengan pendekatan SPA untuk meningkatkan efektivitas dan efisiensi proses penawaran dan transaksi. Aplikasi lelang ini berhasil dibangun dengan fitur-fitur yang lengkap, seperti pendaftaran pengguna, penawaran lelang secara *real-time*, notifikasi, dan informasi detail barang lelang. Aplikasi ini menunjukkan kinerja yang baik dengan respons yang cepat dan antarmuka yang intuitif. Pada penelitian ini didapatkan kesimpulan bahwa SPA memberikan solusi inovatif yang memberikan pengalaman lelang *online* yang mudah dan interaktif. Aplikasi ini dapat memenuhi kebutuhan pasar dan dapat menjadi dasar untuk pengembangan lebih lanjut di masa depan (Putra Mahendri et al., 2023).

2.1.3 Perancangan Sistem Informasi Organisasi Keluarga Mahasiswa

Penelitian ini melakukan pembuatan sistem informasi keanggotaan organisasi Keluarga Mahasiswa Tegal Ciputat berbasis web untuk mempermudah pengelolaan data, sejarah, struktur, dan kegiatan organisasi. Penelitian ini menggunakan metode pengembangan *waterfall*, yang terdiri dari tahapan analisis, perancangan, implementasi, dan pengujian. Tampilan yang dihasilkan dari penelitian ini seperti *login*, menu utama, tampilan data pengurus dan anggota, pendaftaran anggota, dan media galeri. Hasil akhir dari penelitian ini disimpulkan bahwa sistem informasi keanggotaan organisasi berbasis web sangat efektif untuk menata dan menyimpan data dalam database, serta memudahkan dan memperlancar proses pencarian data (Chandra et al., 2020).

2.1.4 Rancang Bangun Sistem Informasi Himpunan Mahasiswa

Penelitian ini bertujuan untuk merancang dan membuat sistem informasi manajemen organisasi Himpunan Mahasiswa Rokan Hulu (HIMAROHU) yang berbasis web, untuk membantu pengelolaan administrasi, informasi, dan komunikasi antara anggota dan pengurus organisasi. Penelitian ini menggunakan metode *waterfall*, yang terdiri dari lima tahap, yaitu analisis sistem, perancangan sistem, implementasi pembuatan kode, pengujian, dan evaluasi.

Terdapat beberapa fitur yang ada, seperti pendataan anggota, informasi agenda kegiatan, informasi beasiswa, laporan pertanggungjawaban pengurus, dan diskusi *online*. Sistem informasi ini dapat diakses oleh empat jenis pengguna, yaitu admin, pengurus, anggota, dan pengawas organisasi, dengan hak akses yang berbeda-beda. Hasil dari penelitian ini menyimpulkan bahwa sistem informasi web yang dibuat dapat memenuhi kebutuhan organisasi HIMAROHU dalam hal administrasi, informasi, dan komunikasi. Sistem informasi ini juga dapat meningkatkan kinerja pengurus, kenyamanan anggota, dan profesionalisme organisasi (Annur M & Delianti V, 2020).

2.1.5 Pembuatan *Front-end* Admin Aplikasi Gerejaku

Penelitian ini berisi pembuatan *frontend* admin untuk aplikasi Gerejaku, sebuah aplikasi yang digunakan untuk mengetahui perkembangan rohani jemaat Gereja Anugrah, dengan menggunakan kerangka kerja Vue. Menggunakan metode *waterfall*, yang terdiri dari empat tahapan: analisis kebutuhan perangkat lunak, desain dan pemrograman, pengujian, dan pendukung dan pemeliharaan. Penelitian ini menghasilkan *frontend* admin yang terintegrasi dengan aplikasi Gerejaku, sehingga admin dapat mengelola data gereja, jemaat, sinode, departemen, posisi, acara, kategori, artikel, dan *slider* dengan mudah, cepat, efektif, dan efisien (Fajarianditya Nugroho et al., 2022).

2.1.6 Pengembangan Prototipe Aplikasi Bimbingan Tugas Akhir

Penelitian ini merupakan pengembangan prototipe aplikasi bimbingan tugas akhir mahasiswa berbasis web yang dapat mengatasi permasalahan pencatatan dan dokumentasi proses bimbingan antara mahasiswa dan dosen pembimbing. Penelitian ini menggunakan metode *prototyping* dengan membagi aplikasi menjadi dua bagian, yaitu *backend* dan *frontend*. *Backend* menggunakan kerangka kerja Laravel, sedangkan *frontend* menggunakan kerangka kerja Vue. Aplikasi juga menggunakan state management dengan vuex untuk mengatur data pada *frontend*. Kerangka kerja Vue dipilih karena mempermudah pembuatan aplikasi dengan pendekatan komponen, yang artinya sebuah aplikasi dibagi menjadi komponen-komponen kecil yang memiliki tampilan dan fungsi yang ter-enkapsulasi (Johan et al., 2023).

2.2 Dasar Teori

Dasar teori merupakan penjelasan yang mendetail mengenai konsep dan prinsip yang menjadi dasar dalam sistem informasi organisasi HKBP Manyar, serta memberikan gambaran keseluruhan tentang cara implementasi sistem informasi yang sedang dibangun. Dalam pengembangan sistem informasi, penggunaan dasar teori sangat penting untuk memastikan bahwa aplikasi berfungsi dengan baik dan sesuai dengan tujuan penggunaannya. Informasi tentang penggunaan dasar teori ini juga membantu mempercepat dan menyederhanakan proses pengembangan sistem informasi secara efektif dan efisien. Penjelasan singkat mengenai poin-poin dasar teori yang relevan untuk sistem Informasi HKBP Manyar adalah sebagai berikut:

2.2.1 Sistem Informasi Organisasi

Menurut Jogianto (dalam Hutahaean, 2014) mengatakan bahwa sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi. sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. Sedangkan informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai nyata atau yang dapat dirasakan dalam keputusan-keputusan yang sekarang

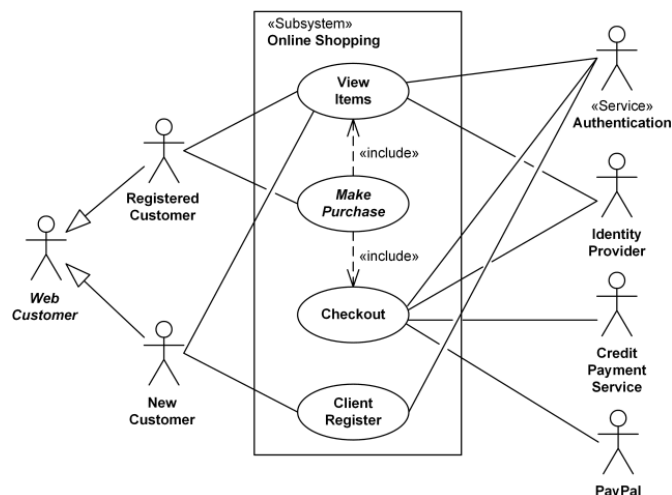
atau keputusan-keputusan yang akan datang, menurut Gordon B. Davis (dalam Hutahaean, 2014). Sehingga sistem informasi bisa dikatakan sebagai suatu kombinasi dari elemen-elemen sistem yang mengolah data menjadi informasi yang bermanfaat bagi pengguna. Sistem informasi menghubungkan antara objek-objek nyata yang terlibat dalam suatu kegiatan atau sasaran tertentu dengan menggunakan teknologi. Sistem informasi mendukung manajemen dan operasional organisasi atau perusahaan dengan menyediakan informasi yang relevan dan bernilai untuk pengambilan keputusan, pengendalian, koordinasi, analisis, dan inovasi. Organisasi adalah sebuah wadah untuk sekumpulan orang yang bekerja sama secara rasional serta sistematis yang dipimpin atau terkendali untuk mencapai tujuan tertentu memanfaatkan sumber daya yang ada di dalamnya. Informasi merupakan salah satu jenis sumber daya yang paling utama dalam suatu organisasi. Bentuk dan kemampuan perusahaan mengelola informasi selain akan mempengaruhi kualitas informasi yang dihasilkan di dalam organisasi tersebut juga akan mempengaruhi kualitas hubungan atau integrasi diantara pelaku organisasi (Susanto A, 2017). Sistem informasi merupakan salah satu komponen dalam suatu organisasi. Di dalam suatu organisasi sistem informasi merupakan suatu alat yang dapat memberikan informasi yang diperlukan kepada semua pihak yang berkepentingan. Karena sistem informasi merupakan komponen dari organisasi maka komponen-komponen sistem informasi juga pada dasarnya merupakan komponen dari organisasi.

2.2.2 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau *Software Requirements Specification* merupakan spesifikasi lengkap dan deskripsi persyaratan perangkat lunak yang harus dipenuhi untuk keberhasilan pengembangan sistem perangkat lunak. Persyaratan ini dapat bersifat fungsional dan non fungsional tergantung pada jenis persyaratannya. Interaksi antara pelanggan dan kontraktor yang berbeda dilakukan karena perlunya memahami sepenuhnya kebutuhan pelanggan.

Dalam pembuatan SKPL ada beberapa kebutuhan yang dijelaskan yaitu kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional merupakan semua persyaratan fungsional yang mungkin mencakup penghitungan, pemrosesan data, dan lainnya, ditempatkan dalam urutan peringkat. Persyaratan fungsional menentukan perilaku yang diharapkan dari sistem-keluaran mana yang harus dihasilkan dari masukan yang diberikan. Kebutuhan fungsional menggambarkan hubungan antara *input* dan *output* sistem. Untuk setiap persyaratan fungsional, deskripsi rinci semua masukan data dan sumbernya, satuan ukuran, dan rentang masukan yang valid harus ditentukan. Sedangkan kebutuhan non fungsional adalah atribut non fungsional yang diperlukan oleh sistem perangkat lunak untuk kinerja yang lebih baik. Contohnya mungkin termasuk keamanan, portabilitas, keandalan, Kompatibilitas Aplikasi, Integritas Data, dan lainnya (Madhuri, 2023).

Dalam pembuatan SKPL sistem informasi HKBP Manyar hanya membuat kebutuhan fungsional saja. Kebutuhan fungsional yang dibuat berupa diagram kasus penggunaan (*Use Case Diagram*). Diagram kasus penggunaan adalah alat penting dalam desain sistem, yang memberikan representasi visual tentang bagaimana pengguna berinteraksi dengan sistem. Ini berfungsi sebagai cetak biru untuk memahami persyaratan fungsional sistem dari sudut pandang pengguna, membantu komunikasi antar pemangku kepentingan dan memandu proses pengembangan (Srinam, 2024). Contoh diagram kasus pengguna terdapat di Gambar 2.1.



Gambar 2. 1 Contoh Diagram Kasus Penggunaan (Srinam, 2024).

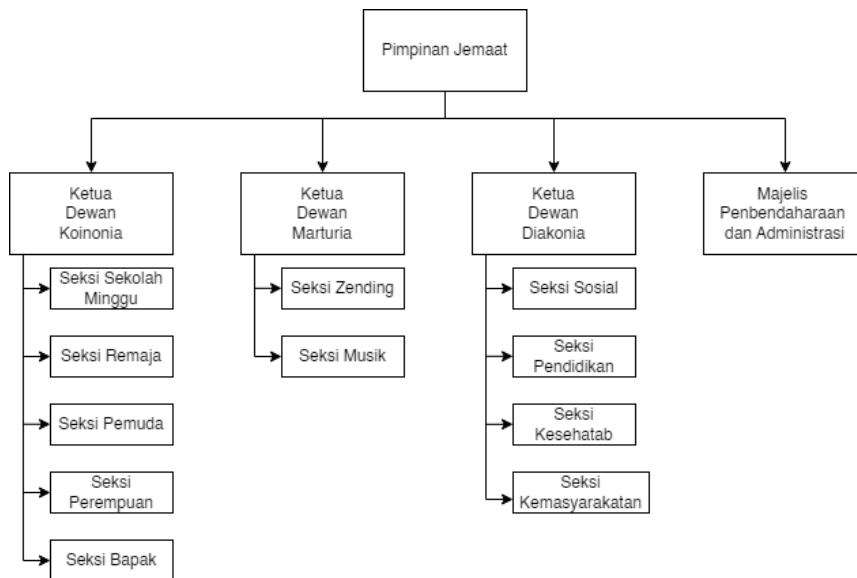
2.2.3 HKBP Manyar

Gereja Huria Kristen Batak Protestan (HKBP) merupakan satu dari berbagai macam gereja Kristen Protestan yang ada di Indonesia. HKBP merupakan gereja Kristen yang berkembang dengan kebudayaan Batak. Gereja HKBP ini memiliki penyebaran yang cukup luas di Indonesia, ini seiring dengan penyebaran orang-orang yang berasal dari suku Batak di seluruh Indonesia. Salah satu gereja yang ada adalah HKBP Manyar, berada di Jalan Manyar Kertoarjo IV No 2, Surabaya, Jawa Timur. Sebagai organisasi nirlaba, HKBP Manyar memiliki anggota yang menjadi bagian di gereja yang disebut jemaat. Jemaat bukan hanya sekedar persekutuan biasa (seperti persekutuan doa atau ibadah) tetapi jemaat adalah persekutuan yang konkrit, sama halnya dengan persekutuan yang lain yaitu memiliki anggota, peraturan, susunan yang tertentu, dan lain-lain. Di dalam kepengurusan HKBP Manyar dibagi menjadi beberapa dewan seperti Koinonia, Marturia, dan Diakonia yang masing-masing memiliki peran dan fungsi yang berbeda. Pada setiap dewan memiliki pengurus dan sistem kerja yang berbeda, sehingga untuk mempermudah jemaat lain mengakses informasi yang ada diperlukan suatu sistem informasi.

Pada organisasi yang ada di HKBP Manyar memiliki struktur Pendeta sebagai pimpinan jemaat, lalu di bawahnya terdapat Ketua Dewan Koinonia, Ketua Dewan Marturia, Ketua Dewan Diakonia, dan Majelis Perbendaharaan dan Administrasi. Di bawah dewan terdapat seksi-seksi dengan kepengurusan yang berbeda-beda. Jemaat merupakan orang yang berada di bawah pimpinan jemaat. Ketua dewan dan pengurus yang ada di suatu seksi juga disebut sebagai jemaat. Gambar struktur organisasi yang ada di HKBP Manyar terdapat di Gambar 2.2.

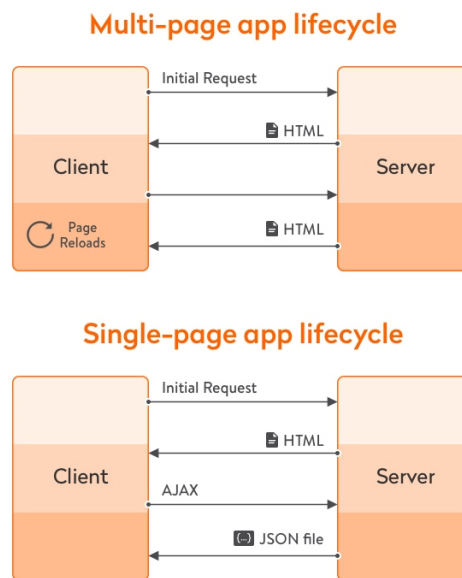
2.2.4 Single Page Application

Single Page Application (SPA) adalah suatu pendekatan di mana seluruh aplikasi berjalan sebagai satu halaman web tunggal. Dalam pendekatan ini, lapisan presentasi untuk seluruh aplikasi telah dipisahkan dari server dan dikelola dari dalam *browser* (Scott, 2015). Terdapat perbedaan antara pendekatan SPA dengan pendekatan tradisional atau *Multi Page Application* (MPA). Perbedaan umumnya bisa terlihat pada *lifecycle* antara kedua pendekatan tersebut. Pada MPA *request* dan *respons* akan mengikuti permintaan HTTP awal, sedangkan SPA menggunakan *Asynchronous JavaScript and XML* (AJAX) untuk meminta data, dan *respons* yang diterima berupa data dalam format *JavaScript Object Notation* (JSON) atau potongan HTML yang sudah di-*render* sebelumnya. Begitu data tiba di sisi klien, klien akan sebagian me-*render* HTML untuk merepresentasikan perubahan (Fink & Flatow, 2014).



Gambar 2. 3 Struktur Pengurus Organisasi HKBP Manyar

SPA dan MPA memiliki perbedaan *lifecycle* ketika memuat isi dari halaman konten yang ditampilkan. Perbedaan tersebut digambarkan di Gambar 2.3.



Gambar 2. 5 Perbedaan *Lifecycle Multi Page Application* dan *Single Page Application*

Penjelasan perbedaan antara SPA dan MPA dapat dijelaskan menjadi tiga poin di bawah ini (Fink & Flatow, 2014) :

1. Tidak ada *refresh* browser: Dalam SPA, tampilan *browser* bukanlah halaman HTML lengkap. Mereka hanyalah bagian dari *Document Object Model (DOM)* yang membentuk area yang dapat dilihat di layar. Setelah halaman pertama di-*load*, semua alat yang diperlukan untuk membuat dan menampilkan tampilan sudah diunduh dan siap digunakan. Jika diperlukan tampilan baru, tampilan tersebut dibuat secara lokal di browser dan

dipasang secara dinamis ke DOM melalui JavaScript. Sehingga tidak diperlukan *refresh browser* saat membuka isi konten yang ada di halaman browser tersebut.

2. Presentasi logic terjadi di sisi klien: Karena logika presentasi dalam SPA sebagian besar berada di sisi klien, tugas menggabungkan HTML dan data dipindahkan dari server ke browser. Seperti di sisi server, HTML sumber berisi *placeholder* di mana data akan dimasukkan (dan mungkin instruksi *render* lainnya). Templat sisi klien ini digunakan sebagai dasar untuk mencetak tampilan baru di klien. Ini bukan templat HTML untuk halaman lengkap, namun hanya untuk bagian dari halaman yang direpresentasikan oleh tampilan.
3. Server dapat berupa data saja: Dalam SPA, beberapa pendekatan dapat digunakan untuk *me-render* data dari server. Ini termasuk *render* sebagian di sisi server, di mana potongan HTML dikombinasikan dengan data dalam respons server. *Render* dilakukan di sisi klien dan hanya data yang dikirim dan diterima selama transaksi bisnis. Ini selalu dilakukan secara asinkron melalui API XHR. Format pertukaran data biasanya berupa JSON, meskipun tidak harus demikian.

Bagian satu halaman pada SPA mengacu pada *file* HTML awal, atau *shell*. HTML tunggal ini di-*load* sekali dan hanya sekali, dan ini berfungsi sebagai titik awal untuk sisa *file* aplikasi. Ini adalah satu-satunya pemuatan *browser* penuh yang terjadi di SPA. Bagian selanjutnya dari aplikasi dimuat secara dinamis dan independen dari *shell*, tanpa memuat ulang satu halaman penuh, ini akan memberikan persepsi kepada pengguna bahwa halaman tersebut telah berubah.

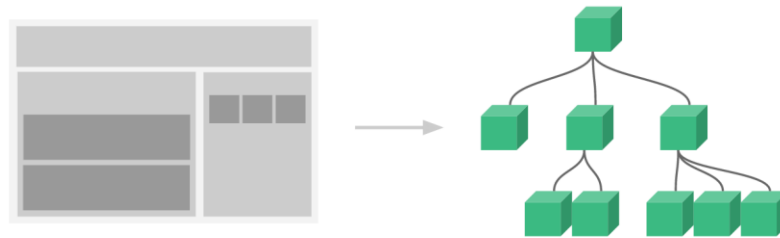
Konsep SPA awalnya berasal dari ide untuk membuat sebuah halaman web yang responsif dan bisa meningkatkan kualitas web dari sisi *User Experience*. Dalam pembuatan aplikasi web dengan SPA terdapat keuntungan lain yang bisa menjadi pertimbangan untuk menggunakan konsep SPA dalam pengembangan aplikasi web, yaitu:

1. Lapisan presentasi yang terpisah: Kode yang mengatur tampilan UI dan perilaku disimpan di sisi klien membuat server dan klien terpisah. Manfaatnya adalah keduanya dapat dipelihara dan diperbarui secara terpisah.
2. Muatan transaksi lebih cepat dan ringan: Transaksi dengan server lebih ringan dan cepat, karena setelah pengiriman awal, hanya data yang dikirim dan diterima dari server. Ini berbeda dengan MPA karena MPA memiliki beban berupa harus merespons dengan konten halaman berikutnya. Karena seluruh halaman di-*render* ulang, konten yang dikembalikan dalam MPA juga mencakup markup HTML.
3. Waktu tunggu pengguna lebih sedikit: SPA dimuat dengan *shell* dan sejumlah kecil *file* pendukung dan kemudian dibangun seiring navigasi pengguna, membuat aplikasi web ini berjalan dengan cepat. Tampilan di-*render* secara dinamis lalu disajikan dalam bentuk pertukaran tampilan konten membuat pengguna tidak perlu menunggu lama.

2.2.5 Vue.Js

Vue.js (Vue) adalah sebuah kerangka kerja (*framework*) JavaScript *open-source* yang digunakan untuk membangun antarmuka pengguna (*User Interface*) dalam aplikasi web yang dinamis. Pustaka intinya difokuskan pada layer tampilan saja, dan sangat mudah untuk diintegrasikan dengan pustaka yang lain atau dengan proyek yang sudah ada. Di sisi lain, Vue sangat mampu memberikan dan mendukung SPA yang canggih ketika dikombinasikan dengan fitur Vue dan pustaka yang ada (Dokumentasi Vue.js, 2024).

Vue dirancang dengan fokus pada konsep komponen, yang memungkinkan pengguna Vue untuk membagi aplikasi web menjadi bagian-bagian yang independen dan dapat digunakan kembali. Ini membantu dalam mengorganisasi dan manajemen kode dengan lebih efisien, serta mempermudah pengembangan, pemeliharaan, dan pengujian aplikasi.



Gambar 2. 7 Sistem Komponen dalam Vue.js (Dokumentasi Vue.js, 2024)

Vue memiliki sistem reaktivitas yang kuat, yang memungkinkan tampilan untuk secara otomatis memperbarui dirinya ketika data yang mendasarinya berubah, menjadikannya sangat cocok untuk membangun aplikasi web yang responsif dan interaktif. Vue juga terkenal dengan sintaksis yang mudah dipahami dan komunitas pengembang yang aktif, menjadikannya salah satu kerangka kerja JavaScript yang paling populer dan sering digunakan dalam pengembangan aplikasi web modern.

2.2.6 Manajemen State Pinia

Pinia adalah pustaka manajemen state (*state management library*) untuk Vue. Ini memungkinkan untuk berbagi status antara elemen dan halaman. Ini menciptakan kerangka kerja manajemen state yang mudah digunakan dan diketik dengan menggunakan konsep reaktivitas baru di Vue (Dokumentasi Vue.js, 2024). Pinia sangat ringan, hanya mencapai 1KB. Pinia dapat menawarkan penyelesaian otomatis yang lengkap dan tepat dalam JavaScript karena Pinia menyimpulkan semua tipe data. Pinia juga memiliki desain modular, mudah digunakan, dan dapat diperluas. Secara keseluruhan, Pinia tampak ringan, tidak rumit, dan lugas. Ini sangat mirip berurusan dengan elemen dan berisi semua alat yang diperlukan untuk pemrograman dinamis seperti pembuatan aplikasi SPA.

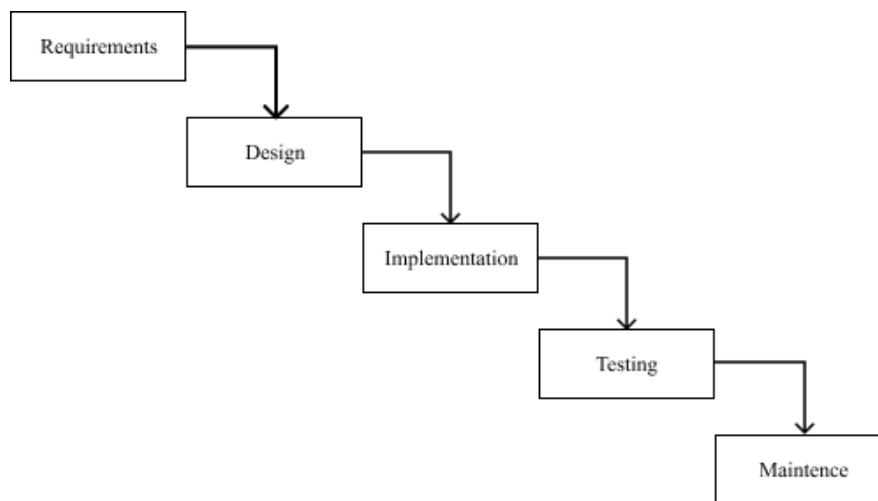
Pinia berbasis pada konsep *store*, yang berfungsi sebagai wadah untuk state aplikasi. Setiap store dapat berisi *state*, *getters*, dan *actions*. Ini menjadi konsep penting dalam pustaka manajemen state.

1. **State:** Merujuk pada data yang disimpan dan dikelola oleh pustaka manajemen *state*. Ini bisa berupa variabel, objek, atau struktur data lainnya yang merepresentasikan keadaan aplikasi pada suatu waktu tertentu. *State* bersifat reaktif, artinya ketika nilai *state* berubah, komponen yang menggunakannya akan secara otomatis diperbarui.
2. **Getter:** Adalah metode yang digunakan untuk mengakses dan mengambil data dari *state*. Mereka berfungsi sebagai “pencari” nilai dari *state*. *Getter* biasanya digunakan untuk menghitung atau memanipulasi data sebelum mengembalikannya.
3. **Actions:** Adalah metode yang digunakan untuk melakukan operasi yang mempengaruhi *state*. Mereka berfungsi sebagai “pemroses” yang dapat mengubah nilai *state*. *Actions* dapat melakukan pemanggilan API, mengubah data, atau melakukan operasi lain sebelum mengubah *state*.

2.2.7 Metode Pengembangan Perangkat Lunak

Software Development Life Cycle (SDLC) adalah metodologi untuk merancang, membangun, dan memelihara sistem informasi dan industri (Alshamrani & Bahattab, 2015). Semua model SDLC yang telah disarankan berbagi sifat dasar. Terdapat beberapa metode dalam pengembangan perangkat lunak seperti *Waterfall*, *Spiral*, *Incremental*, *Rational Unified Process* (RUP), *Rapid Application Development* (RAD), *Agile Software Development*, dan *Prototyping*. Semuanya terdiri dari rangkaian tahapan atau langkah yang harus diikuti dan diselesaikan pengembang dan perancang sistem untuk mencapainya mengembangkan sistem dan memberikan produk yang dibutuhkan.

Dalam pengembangan sistem informasi ini menggunakan metode *Waterfall*. Model *Waterfall* adalah model SDLC tertua dan paling terkenal. Ciri khusus model ini adalah langkah-langkahnya yang berurutan. Ini berjalan ke bawah melalui tahapan analisis kebutuhan, desain, pengkodean, pengujian, dan pemeliharaan. Apalagi itu memastikan kekurangan desain sebelum pengembangan produk. Model ini bekerja dengan baik untuk proyek-proyek yang pengendalian kualitas menjadi perhatian utama karena intensifnya dokumentasi dan perencanaan. Tahapan yang membangun ini model bukanlah tahapan yang tumpang tindih, yang berarti bahwa model air terjun dimulai dan diakhiri satu tahap sebelum memulai yang selanjutnya (Alshamrani & Bahattab, 2015). Diagram alur metode *Waterfall* terdapat di Gambar 2.5.



Gambar 2. 9 Diagram Metode Waterfall (Sommerville, 2011)

1. **Requirements:** Membuat Analisis dan Definisi Persyaratan Layanan sistem, kendala, dan tujuan ditetapkan melalui konsultasi dengan pengguna sistem. Kemudian, didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.
2. **Design:** Mengalokasikan persyaratan ke sistem perangkat keras atau perangkat lunak dengan menetapkan arsitektur sistem secara keseluruhan. Desain perangkat lunak melibatkan identifikasi dan deskripsi abstraksi dasar sistem perangkat lunak dan hubungan mereka.
3. **Implementation:** Mewujudkan tahap desain sebagai serangkaian program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.
4. **Testing:** Sistem unit program atau program individu diintegrasikan dan diuji sebagai sistem lengkap untuk memastikan bahwa persyaratan perangkat lunak telah terpenuhi. Setelah pengujian, sistem perangkat lunak diserahkan kepada pelanggan.
5. **Maintenance:** Sistem diinstal dan digunakan secara praktis. Pemeliharaan melibatkan perbaikan kesalahan yang tidak ditemukan pada tahap-tahap sebelumnya dalam siklus hidup, peningkatan implementasi unit sistem, dan peningkatan layanan sistem seiring dengan penemuan persyaratan baru (Sommerville, 2011).

2.2.8 Firebase

Firebase Realtime Database adalah database yang di-*hosting* di cloud. Data disimpan sebagai JSON dan disinkronkan secara *real-time* dengan setiap klien yang terhubung. Ketika membuat aplikasi lintas platform dengan SDK platform Apple, Android, dan JavaScript kami, semua klien akan menggunakan satu instance Realtime Database yang sama dan menerima perubahan data terbaru secara otomatis (Dokumentasi Firebase, 2024b). Firebase memungkinkan untuk menciptakan aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi klien. Data dipertahankan secara lokal, dan meskipun sedang offline, peristiwa *real-time* terus dipicu, sehingga pengguna akhir akan merasakan pengalaman yang responsif. Ketika koneksi perangkat pulih kembali, Realtime Database akan menyinkronkan perubahan data lokal dengan pembaruan jarak jauh yang terjadi selama klien offline, sehingga setiap perbedaan akan otomatis dihilangkan.

Salah satu fitur pada Firebase adalah menyediakan berbagai layanan *backend as a service* (BaaS). Ini berarti Firebase menawarkan infrastruktur *backend* yang sudah siap pakai untuk membantu pengembang dalam membangun dan mengelola aplikasi tanpa perlu mengurus infrastruktur server secara langsung. Salah satu layanan utama dari Firebase sebagai BaaS adalah:

1. **Database Realtime:** Firebase menyediakan database NoSQL yang dapat disinkronkan secara *real-time* antara klien (aplikasi *frontend*) dan server. Ini memungkinkan aplikasi untuk meng-*update* data secara instan dan menampilkan perubahan kepada pengguna tanpa memerlukan *polling* server.
2. **Authentication:** Firebase menyediakan layanan autentikasi yang mudah digunakan untuk mengelola proses *login* dan otorisasi pengguna. Ini mencakup autentikasi dengan email/password, integrasi dengan OAuth provider (seperti Google, Facebook, Twitter), serta autentikasi dengan nomor telepon.
3. **Storage:** Firebase Storage adalah layanan penyimpanan *file* yang aman, yang memungkinkan pengguna untuk menyimpan dan membagikan berbagai jenis konten, seperti gambar, video, dan *file* lainnya.
4. **Cloud Functions:** Firebase Cloud Functions memungkinkan pengembang untuk menulis dan mengelola kode backend tanpa harus mengelola server fisik. Fungsi ini dapat digunakan untuk memproses data, memicu tindakan berdasarkan peristiwa tertentu, dan melakukan operasi backend lainnya.
5. **Analytics:** Firebase menyediakan alat analitis yang kuat untuk melacak kinerja aplikasi, pemahaman pengguna, dan keberhasilan kampanye pemasaran.
6. **Performance Monitoring:** Firebase Performance Monitoring membantu pengembang dalam memahami performa aplikasi mereka dengan memberikan wawasan tentang waktu respons, kinerja jaringan, dan masalah lain yang mempengaruhi pengalaman pengguna.
7. **Remote Config:** Firebase Remote Config memungkinkan pengembang untuk mengubah perilaku dan penampilan aplikasi mereka secara dinamis tanpa perlu merilis pembaruan aplikasi. Ini membantu dalam melakukan eksperimen dengan cepat dan menyesuaikan aplikasi dengan kebutuhan pengguna.

Firebase memungkinkan pengembang untuk fokus pada pengembangan fitur dan pengalaman pengguna tanpa harus menghabiskan waktu dan upaya untuk mengelola infrastruktur server. Untuk pengembangan sistem informasi HKBP Manyar menggunakan layanan pada firebase untuk membuat *backend*-nya.

Firebase selain menjadi database juga bisa digunakan sebagai alat untuk melakukan *hosting* pada tahap peluncuran. Firebase menyediakan fitur hosting bernama Firebase Hosting, fitur hosting yang cepat dan aman untuk aplikasi web, konten statis dan dinamis, serta

microservice (Dokumentasi Firebase, 2024a). Firebase *Hosting* merupakan platform yang dirancang untuk mendukung pengembangan web modern dengan menyediakan infrastruktur, fitur, dan alat yang disesuaikan. Dengan dukungan untuk kerangka kerja JavaScript *front-end* seperti Vue. *Hosting* memungkinkan para pengembang untuk men-*deploy* berbagai jenis situs dan aplikasi, mulai dari halaman landing sederhana hingga SPA. Pengguna dapat dengan mudah men-*deploy* konten statis menggunakan Firebase CLI, sementara *Cloud Run* memungkinkan penayangan konten dinamis dan *hosting microservice*. Opsi konfigurasi *hosting* yang ringan memungkinkan pengguna untuk membangun SPA yang kompleks dengan mudah, termasuk pengaturan URL perutean sisi klien, *header* kustom, dan penayangan konten yang dilokalkan. Firebase Hosting juga memberikan opsi domain dan subdomain, termasuk subdomain bawaan pada domain *web.app* dan *firebaseapp.com*.

2.2.9 Unified Modelling Language

Unified Modelling Language (UML) adalah suatu alat atau media untuk memvisualkan perancangan sistem yang berorientasi pada objek. UML yang digunakan dalam tugas akhir ini adalah *use case diagram* dan *activity diagram*. Tujuan utama penggunaan UML adalah sebagai berikut:

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik (Haviluddin, 2011)

2.2.10 Figma

Figma adalah aplikasi yang digunakan oleh UI dan UX designer untuk membuat tampilan antarmuka *website* dan *mobile*. Figma dapat digunakan sebagai alat untuk mendesain aplikasi dan alat *prototyping*. Figma dibuat untuk dapat membantu para penggunanya agar bisa berkolaborasi dalam proyek dan bekerja dalam bentuk tim sekaligus di mana saja. Tim Designer dapat saling memberikan komentar dan saran, merancang desain secara bersamaan, dan aplikasi Figma dapat diakses oleh tim *frontend*. Salah satu kelebihan Figma adalah kemampuannya yang *real-time*, di mana setiap perubahan yang dilakukan oleh pengguna akan disimpan secara otomatis (Suryaningsih et al., 2020). Figma dan Adobe Photoshop adalah dua perangkat lunak yang berbeda untuk desain grafis. Figma adalah alat desain kolaboratif yang didesain khusus untuk tim, sedangkan Adobe Photoshop lebih terfokus pada pengeditan foto dan manipulasi gambar. Perbedaan utama antara Figma dan Adobe Photoshop adalah bahwa Figma didesain untuk bekerja dalam *cloud* dan memungkinkan tim untuk bekerja secara bersamaan pada proyek yang sama, sementara Adobe Photoshop biasanya digunakan oleh seorang desainer individual yang bekerja pada proyek-proyek yang lebih kecil (Pramudita et al., 2021). Figma dalam Tugas akhir ini digunakan untuk pembuatan desain *wireframe* untuk mempermudah dalam implementasi pembuatan sistem.

2.2.11 Wireshark

Wireshark adalah salah satu analisis paket bebas yang bisa diakses semua pengguna karena merupakan *software open source*. Perangkat ini digunakan sebagai pemecah permasalahan jaringan, analisis perangkat lunak, serta mengembangkan protokol komunikasi. Wireshark juga berfungsi sebagai alat pembanding dari sekian banyak aplikasi Network Analyzer yang banyak digunakan oleh Network Administrator untuk menganalisa kinerja jaringan dan mengontrol lalu lintas data di jaringan yang dikelola. Wireshark mampu menangkap paket-paket data yang ada pada jaringan tersebut. Semua jenis paket informasi dalam berbagai format protokol dapat dengan mudah ditangkap dan dianalisa.

Wireshark dapat menangkap lalu lintas dari berbagai jenis media jaringan, termasuk Ethernet, LAN Nirkabel, Bluetooth, dan USB, dengan dukungan yang bergantung pada perangkat keras dan sistem operasi yang digunakan. Selain itu, Wireshark mampu membuka pengambilan paket dari berbagai program penangkapan lainnya dan menyimpan paket yang ditangkap dalam berbagai format. Fitur-fitur ini menjadikan Wireshark alat yang sangat *powerful* dan fleksibel dalam analisis jaringan, memungkinkan para profesional untuk menangkap, menganalisa, dan menyimpan data jaringan dalam berbagai format yang berguna untuk berbagai keperluan (Richard Sharpe et al., 2024).

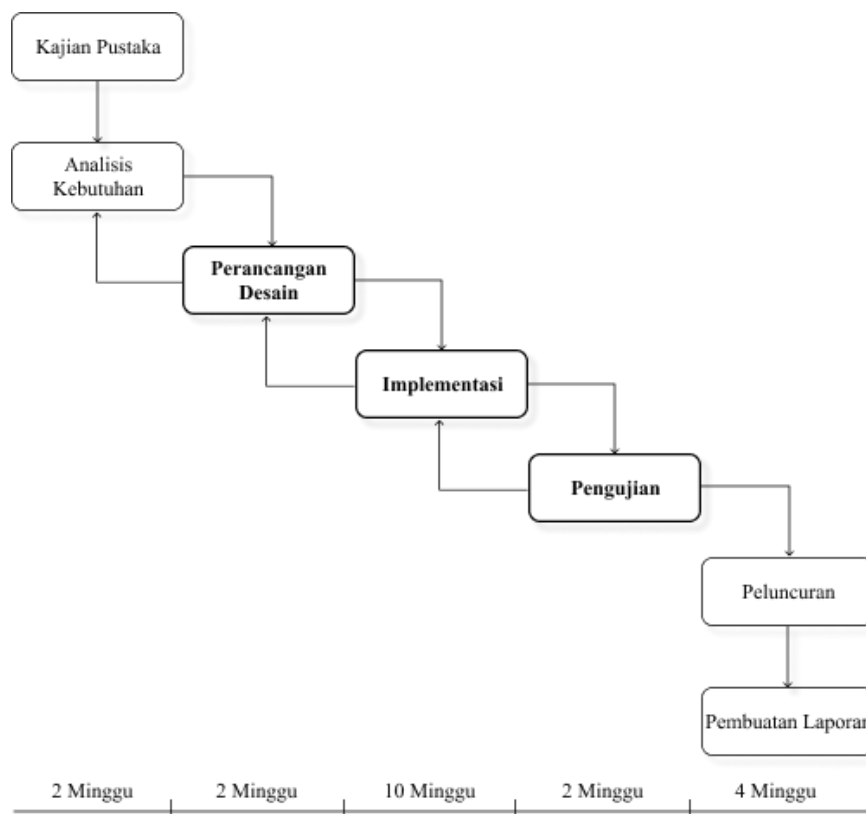
(Halaman ini sengaja dikosongkan)

BAB III METODOLOGI

Bab ini berisi penjelasan mengenai tahapan-tahapan yang dilakukan dalam pengembangan sistem informasi HKBP Manyar. Setiap luaran dari proses yang dilakukan akan menjadi masukan untuk proses berikutnya.

Membangun sistem memerlukan penggunaan metode pengembangan sistem yang dapat membimbing pembuatnya dalam menciptakan sebuah sistem yang berkualitas. Metode pengembangan sistem informasi memiliki daur hidup, daur hidup tersebut yang dinamakan SDLC (*Systems Development Life Cycle*) atau daur hidup pengembangan sistem. SDLC adalah suatu pendekatan klasik yang digunakan untuk mengembangkan, merawat, dan menggunakan sistem informasi. HKBP Manyar, dalam pengembangan sistem informasinya, mengadopsi model air terjun dari SDLC, yang juga dikenal sebagai *Waterfall*.

Proses pengembangan proyek perangkat lunak sistem informasi di organisasi HKBP Manyar terbagi menjadi tujuh tahapan, dimulai dari kajian pustaka hingga pembuatan laporan. Rincian mengenai alur fase tahapannya dapat dilihat di Gambar 3.1.



Gambar 3. 1 Tahapan Pengembangan Berdasarkan Metode *Waterfall*

Tahapan-tahapan tersebut akan berlangsung selama lima bulan dengan alokasi waktu yang terperinci seperti yang tercantum di bawah gambar. Konsep SPA akan dijelaskan pada tahap dengan penulisan yang lebih tebal, yaitu tahap perancangan desain, tahap implementasi, dan tahap pengujian. Pada tahap yang sudah disebutkan, terdapat dua skenario dimana tahapan tersebut dapat kembali ke tahap sebelumnya atau melanjutkan ke tahap berikutnya. Pengulangan ke tahap sebelumnya terjadi jika terdapat penambahan kebutuhan yang sebelumnya belum terdefinisi, sementara tahap akan berlanjut ke tahap berikutnya jika telah selesai tanpa adanya penambahan kebutuhan.

3.1 Tahap Kajian Pustaka

Sebelum memulai proyek pengembangan sistem informasi untuk organisasi HKBP Manyar, langkah awal yang diperlukan adalah melakukan proses kajian pustaka. Pada tahap ini, dilakukan pengkajian terhadap penelitian yang relevan, termasuk sistem informasi, SPA, dan kerangka kerja Vue. Kajian pustaka ini memiliki tujuan untuk menganalisis objek penelitian dan memberikan pemahaman awal dalam pengembangan sistem informasi yang mengadopsi konsep SPA dan menggunakan kerangka kerja Vue. Dari hasil kajian pustaka tersebut, dapat ditemukan dasar-dasar pengembangan yang diperlukan dalam membangun sistem informasi HKBP Manyar.

Dalam proyek ini, akan dilakukan pencarian kajian-kajian terkait yang telah dilakukan oleh pengembang lain dan terdokumentasikan dalam bentuk karya tulis. Kajian pustaka ini akan dilakukan secara mendalam dalam topik pengembangan aplikasi web dengan menggunakan metode SPA. Dari hasil kajian tersebut, akan dilakukan analisis mengenai kelebihan dan kelemahan metode SPA dibandingkan dengan metode lainnya, serta menentukan kebutuhan umum yang perlu dipenuhi. Selain itu, akan ditentukan juga alat-alat bantu yang akan digunakan dalam pengembangan. Melalui proses kajian ini, diperoleh dasar ilmu dan alat bantu yang berguna dalam mengembangkan sistem informasi untuk gereja HKBP Manyar dengan menggunakan metode SPA.

3.2 Tahap Analisis Kebutuhan

Pada tahap ini, dilakukan identifikasi terhadap kebutuhan-kebutuhan yang diperlukan dalam pembangunan sistem informasi untuk organisasi. Kebutuhan-kebutuhan ini akan menjadi dasar untuk merancang desain sistem aplikasi web. Penetapan kebutuhan dilakukan melalui dua metode, yaitu analisis kajian pustaka yang telah dilakukan sebelumnya dan melalui proses wawancara dengan pihak-pihak terkait.

Wawancara dilakukan dengan pemangku kepentingan yang terdiri dari pengurus dan jemaat gereja. Pemangku kepentingan yang akan diwawancarai adalah sekretaris bersama pendeta dan jemaat yang diwawancarai adalah jemaat yang sudah lama tergabung dalam organisasi HKBP Manyar. Agar mendapatkan pemahaman yang mendalam, akan digunakan metode kualitatif dalam mencari jawaban kebutuhan. Pertanyaan yang akan disampaikan kepada sekretaris organisasi adalah sebagai berikut:

- Bagaimana sistem informasi dapat membantu tugas sehari-hari Anda sebagai sekretaris?
 - Apa saja tantangan atau hambatan yang sering dihadapi dalam pengelolaan informasi di organisasi?
 - Apakah ada proses atau kegiatan spesifik yang Anda rasa dapat ditingkatkan efisiensinya melalui sistem informasi?
 - Bagaimana proses pelaporan dan dokumentasi saat ini, dan apa yang dapat ditingkatkan?
 - Apakah pernah mengetahui ada sistem informasi yang bisa dibuat dalam bentuk aplikasi web dan bisa mempermudah dalam menyebarkan informasi?
- Sedangkan pertanyaan untuk anggota organisasi sebagai berikut:
- Bagaimana Anda berinteraksi dengan informasi yang disediakan oleh gereja saat ini, terutama melalui media digital?
 - Apakah ada harapan atau kebutuhan khusus dari jemaat terkait penyampaian informasi dan aktivitas gereja melalui platform digital?
 - Apakah ada ide atau saran untuk fitur-fitur tertentu yang dapat disertakan dalam aplikasi web untuk memenuhi kebutuhan jemaat?
 - Bagaimana preferensi Anda terkait desain antarmuka dan pengalaman pengguna (*user experience*) dalam menggunakan aplikasi web gereja?

Dari pertanyaan yang disiapkan ini akan muncul pertanyaan tambahan untuk menggali kebutuhan lebih lanjut. Hasil dari wawancara tersebut akan menentukan kebutuhan yang diperlukan atau tidak sehingga dapat mencegah terjadinya kebutuhan yang berlebihan. Hasil dari langkah ini adalah dokumen berikut:

1. Dokumen Elisitasi Kebutuhan

Setelah proses kajian pustaka dan wawancara, dilakukan proses analisis kebutuhan yang diawali dengan membuat dokumen hasil wawancara dan studi literatur sistem informasi gereja yang serupa. Dokumen hasil wawancara merupakan rangkuman transkrip dialog saat wawancara berlangsung. Kemudian, dihasilkan daftar hasil evaluasi sistem informasi organisasi HKBP Manyar iterasi pertama dan kebutuhan sistem. Bukti proses wawancara ditandai dengan dokumen perjanjian dengan stakeholders yang memuat hasil wawancara mengenai kebutuhan sistem. Selanjutnya dilakukan studi literatur sistem informasi web dengan melakukan pencarian web gereja-gereja lalu dirangkum fitur apa saja yang ada dan bisa dijadikan kebutuhan sistem. Didapat kebutuhan apa saja yang diperlukan melalui wawancara dan studi literatur.

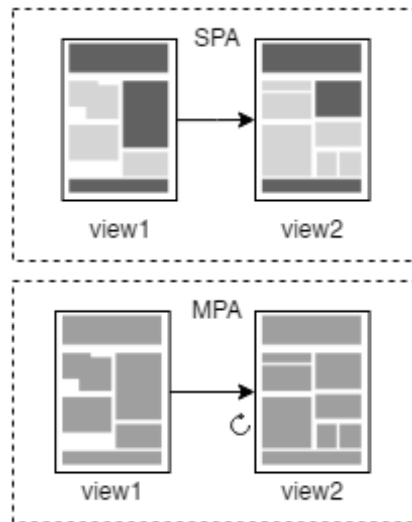
2. Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

Dokumen SKPL menjadi acuan dalam pengembangan hingga implementasi sistem. Dokumen SKPL terdiri dari penjelasan serta visualisasi diagram kebutuhan sistem seperti diagram kasus penggunaan, kebutuhan fungsional dan non fungsional sistem, dokumen spesifikasi kasus penggunaan, diagram aktivitas, diagram hubungan entitas (pemodelan data). Diagram kasus penggunaan menggambarkan kebutuhan utama yang dibutuhkan oleh sistem. Setiap kebutuhan utama memiliki kebutuhan pendukung yang dijelaskan pada dokumen spesifikasi kasus penggunaan. Dokumen spesifikasi kasus penggunaan menjelaskan tentang alur kejadian sistem bekerja dan persyaratan yang dibutuhkan untuk menjalankan sistem. Dari diagram kasus penggunaan dan dokumen spesifikasi kasus penggunaan, dihasilkan daftar kebutuhan fungsional dan kebutuhan non fungsional yang menjadi kebutuhan spesifik sistem. Alur kejadian pada sistem terdiri dari alur utama, alur alternatif, dan alur titik tambahan. Kemudian, alur kejadian sistem pada dokumen spesifikasi kasus penggunaan divisualisasikan dalam bentuk diagram aktivitas, diagram sekuen, dan pemodelan data. Pemodelan data digunakan untuk menggambarkan bagaimana proses penyaluran data yang terjadi pada sistem untuk memastikan sistem dapat bekerja dengan baik

3.3 Tahap Perancangan Desain

Pada tahap ini memberikan perincian pada perancangan desain yang akan dibuat. Hal pertama yang dibahas adalah perbedaan desain SPA dan MPA dari segi struktur desain. Selanjutnya, akan dilakukan pembuatan desain database dan antarmuka dari web Sistem Informasi HKBP Manyar. Untuk melakukan perancangan desain, dibutuhkan informasi SKPL yang diperoleh dari tahap analisis kebutuhan.

Pada bagian ini akan dijelaskan perbedaan SPA dan MPA pada tahap desain. Pada struktur yang dibangun antara SPA dan MPA memiliki perbedaan. Pada SPA halaman HTML dibuat satu saja dan menjadi halaman utama yang secara dinamis memperbarui konten berdasarkan interaksi pengguna dengan melakukan perubahan komponen. Sedangkan pada metode MPA terdiri dari banyak halaman HTML yang berbeda untuk setiap tampilan atau fitur. Setiap kali pengguna mengklik link, seluruh halaman akan dimuat ulang. Di Gambar 3.2 akan dijelaskan perbedaan antara perubahan tampilan yang terjadi antara SPA dan MPA.



Gambar 3. 3 Perbedaan Tampilan saat Terjadi Perubahan Tampilan

Struktur desain pada SPA memiliki komponen terpusat. Komponen adalah unit dasar dari aplikasi yang digunakan kembali di berbagai bagian aplikasi. Setiap komponen dapat berisi logika, tampilan, dan *state* yang spesifik untuk bagian tertentu dari aplikasi. Gambar 3.2 menampilkan terdapat komponen dengan kotak berwarna gelap pada tampilan satu dan tampilan dua SPA, dan merupakan komponen yang sama. Ketika tampilan berpindah maka komponen tersebut tidak akan dimuat ulang, perpindahan dari tampilan satu ke tampilan dua atau sebaliknya dan dilakukan berulang juga tidak akan membuat data dimuat kembali. Berbeda dengan MPA, saat dilakukan perpindahan tampilan, maka semua tampilan akan berubah karena server mengirimkan HTML yang baru dan menimbulkan halaman memuat hal yang sama berulang.

Selanjutnya adalah pembuatan desain database dan web antarmuka. Pembuatan desain database dibuat dengan membuat Conceptual Data Model (CDM). CDM yang dibuat dapat melihat entitas apa saja yang ada di Web HKBP Manyar dan relasi antar entitas yang ada. Selanjutnya adalah pembuatan desain web antarmuka. Pembuatan desain yang dilakukan adalah desain *wireframe*. Ini merupakan kerangka aplikasi dengan penempatan fitur dan tombol yang disesuaikan dengan baik. Pada akhir tahap desain, akan dilakukan pembuatan dokumen berupa Dokumen Deskripsi Perangkat Lunak. Dokumen ini berisi gambaran desain arsitektur dan desain antarmuka yang sudah dibuat. Selanjutnya fase akan dilanjutkan ke tahap implementasi.

3.4 Tahap Implementasi

Setelah tahap desain diselesaikan, langkah selanjutnya adalah tahap implementasi. Dalam implementasi sistem informasi HKBP Manyar, SPA digunakan sebagai landasan pengembangan. Pada bagian ini akan dijelaskan perbedaan dari penulisan kode menggunakan SPA dan MPA.

3.4.1 Pembuatan Komponen

Pada pembuatan kode SPA yang paling utama adalah pembuatan komponen. Sedangkan MPA tidak ada pembuatan komponen, semua halaman dibuat secara utuh. SPA menggunakan pendekatan berbasis komponen di mana setiap bagian UI dipecah menjadi komponen terpisah yang bisa digunakan kembali. Komponen yang dibuat juga bisa ditambahkan ke komponen lainnya tanpa memuat ulang komponen yang sama. Contoh pembuatan komponen pada SPA terdapat di Kode 3.1

```

1. //component/ ButtonCounter.vue
2. <script setup>
3. import { ref } from 'vue'
4.
5. const count = ref(0)
6. </script>
7.
8. <template>
9.   <button @click="count++">You clicked me {{ count }} times.</button>
10. </template>

```

Kode 3. 2 Contoh Pembuatan Komponen Pada SPA

```

1. //view/ Home.vue
2. <script setup>
3. import ButtonCounter from './ButtonCounter.vue'
4. </script>
5.
6. <template>
7. <h1>Here is a child component!</h1>
8. <ButtonCounter />
9. </template>

```

Kode 3. 1 Contoh Penggunaan Komponen di Komponen lain SPA

Di contoh pembuatan komponen SPA di bagian a Kode 3.1 merupakan komponen tombol sederhana. Di dalam tombol terdapat interpolasi atau *binding teks*. Interpolasi ini digunakan untuk menampilkan data dinamis pada template HTML. `{{ count }}` adalah interpolasi yang akan menampilkan nilai dari properti `count` yang ada dalam data atau *computed properties* di Vue instance atau komponen. Ketika nilai `count` berubah, Vue.js akan secara otomatis memperbarui teks di dalam elemen tersebut untuk mencerminkan nilai yang terbaru. Saat aplikasi pertama kali dimuat, tombol akan menampilkan teks “You clicked me 0 times.” Setiap kali tombol diklik, nilai `count` akan bertambah satu, dan teks pada tombol akan diperbarui secara otomatis untuk mencerminkan perubahan tersebut.

Dalam Vue.js, komponen dapat dipanggil di dalam komponen lain untuk menciptakan hierarki komponen yang lebih kompleks. Di bagian b Kode 3.1, *file* Home.vue mengimpor komponen ButtonCounter dari './ButtonCounter.vue'. Kemudian, di baris 8 komponen ButtonCounter dipanggil dengan menggunakan tag khusus `<ButtonCounter />`, yang memungkinkan Home.vue untuk menampilkan dan menggunakan fungsionalitas dari ButtonCounter sebagai bagian dari tampilan dan logika Home.vue.

3.4.2 Penggunaan Routes

Tahap implementasi lain yang membedakan SPA dengan MPA adalah penggunaan *routes*. Pembuatan *routes* pada SPA bertujuan untuk membuat navigasi, sehingga ketika ada interaksi dari pengguna, aplikasi dapat mengganti komponen sesuai navigasi yang sudah ditentukan. Dalam MPA, rute-rute diatur di sisi server. Setiap rute mengarah ke file HTML atau skrip server yang menghasilkan halaman HTML baru. Untuk penulisan code antara kedua metode dalam pembuatan *routes* dapat dilihat di Kode 3.2.

```

1. // Contoh Vue Router setup
2. import { createRouter, createWebHistory } from 'vue-router'
3. import HomePage from '@components/HomePage.vue'
4. import AboutPage from '@components/AboutPage.vue'
5.
6. const routes = [
7.   { path: '/', component: HomePage },
8.   { path: '/about', component: AboutPage }
9. ]
10.
11. const router = createRouter({
12.   history: createWebHistory(),
13.   routes
14. })
15.
16. export default router

```

Kode 3. 4 Pembuatan Routes di SPA

```

1. // Contoh routing di server-side PHP
2. switch ($request_uri) {
3.   case '/':
4.     require 'home.php';
5.     break;
6.   case '/about':
7.     require 'about.php';
8.     break;
9.   // Tambahkan rute lainnya
10. }

```

Kode 3. 3 Pembuatan Routes di MPA

Dalam pembuatan *routes* SPA, *routing* dilakukan di sisi klien menggunakan JavaScript. Dalam kode tersebut, rute didefinisikan dengan mengimpor komponen Vue (HomePage dan AboutPage) dan mengaitkannya dengan path tertentu ('/' dan '/about'). *Router* ini kemudian diinisialisasi dengan 'createRouter' dan menggunakan 'createWebHistory' untuk mengelola riwayat penelusuran. Ketika pengguna menavigasi antara rute, aplikasi tidak memuat ulang seluruh halaman, melainkan hanya mengubah konten yang diperlukan di dalam komponen.

Sedangkan pembuatan *routes* MPA, *routing* dilakukan di sisi server, seperti dalam contoh *routing server-side* PHP. Dalam kode tersebut, URL permintaan (\$request_uri) diperiksa, dan *file* PHP tertentu (home.php, about.php) dimuat berdasarkan *path* yang cocok. Setiap kali pengguna menavigasi ke rute yang berbeda, server mengirimkan halaman HTML yang baru, yang mengakibatkan pemuatan ulang seluruh halaman. Pendekatan ini lebih tradisional dan biasanya melibatkan lebih banyak permintaan ke server dibandingkan dengan SPA.

3.4.3 Penggunaan State Management

Perbandingan lainnya adalah SPA memiliki *state management* untuk mengelola data aplikasi secara global, memudahkan berbagi *state* antar komponen. Sedangkan MPA biasanya tidak memerlukan *state management* yang kompleks di sisi klien, karena setiap halaman adalah entitas yang terpisah. Implementasi *state management* ini adalah pembuatan *store* dengan menggunakan Pinia, contohnya seperti Kode 3. 2. Dan untuk contoh penggunaannya pada komponen UserProfile terdapat di Kode 3.3.

```

1. // Contoh Pinia store
2. import { defineStore } from 'pinia'
3. import { ref } from 'vue'
4.
5. export const useUserStore = defineStore('user', () => {
6.   const user = ref(null)
7.
8.   function setUser(newUser) {
9.     user.value = newUser
10.  }
11.
12.  async function fetchUser() {
13.    // Fetch user data and set user
14.    // Example:
15.    // const response = await fetch('/api/user')
16.    // const data = await response.json()
17.    // setUser(data)
18.  }
19.
20.  return {
21.    user,
22.    setUser,
23.    fetchUser
24.  }
25. })

```

Kode 3. 6 Pembuatan store useUserStore dengan Pinia

```

1. // components/UserProfile.vue
2. <template>
3.   <div>
4.     <h1>User Profile</h1>
5.     <div v-if="user">
6.       <p>Name: {{ user.name }}</p>
7.       <p>Email: {{ user.email }}</p>
8.     </div>
9.     <div v-else>
10.      <p>Loading user data...</p>
11.    </div>
12.  </div>
13. </template>
14.
15. <script>
16. import { useUserStore } from '@/stores/user'
17. import { onMounted } from 'vue'
18.
19. export default {
20.   setup() {
21.     const userStore = useUserStore()
22.     // Fetch user data when the component is mounted
23.     onMounted(() => {
24.       userStore.fetchUser()
25.     })
26.
27.     return {
28.       user: userStore.user
29.     }
30.   }
31. }
32. </script>

```

Kode 3. 5 Komponen *UserProfil*

Store useUserStore dibuat menggunakan Pinia. Store ini didefinisikan dengan defineStore dan mencakup *state*, *action*, serta *getter*. Di kode ini, *user* adalah *state* yang diinisialisasi dengan *null* menggunakan *ref*. Store ini juga memiliki dua fungsi *setUser* untuk

memperbarui *state user* dengan nilai baru, dan *fetchUser* untuk mengambil data pengguna dari API dan memperbarui *state user*. Fungsi *fetchUser* saat ini hanya mengandung contoh komentar yang menunjukkan bagaimana data pengguna dapat diambil dan diatur.

Pada komponen *UserProfile.vue*, *store useUserStore* digunakan untuk menampilkan profil pengguna. *Store* diimpor dan diinisialisasi di dalam *setup function*. Ketika komponen di-*mount*, *fetchUser* dipanggil untuk mengambil data pengguna. Data pengguna kemudian di-*bind* ke *template* menggunakan ‘*v-if*’ untuk menampilkan informasi pengguna jika data tersedia, atau pesan “*Loading user data...*” jika tidak. Dengan cara ini, *state* yang dikelola oleh *store useUserStore* dapat digunakan dan di-*update* di dalam komponen Vue, memungkinkan manajemen *state* yang terpusat dan reaktif di seluruh aplikasi.

Selanjutnya, pengembangan dengan konsep SPA diimplementasikan menggunakan kerangka kerja Vue. Vue memungkinkan pembangunan aplikasi yang mudah dengan komponen yang sederhana, fleksibel, dan mudah dipelihara dalam konteks SPA. SPA berfokus pada kecepatan muat halaman web yang lebih cepat dan penggunaan *resource* yang lebih sedikit sehingga dapat meningkatkan pengalaman pengguna.

Untuk penyimpanan data, dipilih database Firebase. Firebase menawarkan layanan *Backend as a Service* (BaaS) yang memudahkan pengelolaan data dan menyediakan berbagai fitur seperti autentikasi pengguna, penyimpanan data, dan *hosting*. Integrasi Firebase memastikan sistem informasi HKBP Manyar dapat dibangun dengan handal dan responsif. Setelah dilakukan tahap implementasi dilanjutkan tahap testing pada implementasi yang sudah dibuat.

3.5 Tahap Pengujian

Dalam tahap pengujian terdapat dua pengujian yang dilakukan. Pengujian pertama adalah pengujian performa aplikasi web SPA yang dilakukan oleh pengembang dan pengujian kedua merupakan pengujian sistem yang sudah dibuat. Pada pengujian pertama berfokus pada performa dari web yang sudah dibuat. Pengujian menggunakan Wireshark sebagai alat untuk menghitung jumlah paket data yang masuk dan keluar dari web server. Pengujian juga menggunakan fitur Developer Tools pada Google Chrome untuk mendapatkan hasil *resource* yang terpakai. Wireshark juga bisa menghitung waktu transfer data selama pengujian berjalan. Pengujian dilakukan dengan menggunakan web dengan metode MPA yang memiliki fungsi yang sama dengan sistem yang dikembangkan. Lalu dilanjutkan dengan pengujian dengan web yang sudah dikembangkan. Skenario yang dilakukan pada masing-masing pengujian akan disamakan supaya hasil pengujian mendekati akurat. Untuk pengujian *resource* dilakukan skenario yang lebih sedikit, tetapi dilakukan sebanyak dua kali untuk melihat penggunaan *resource* yang terpakai.

Pengujian sistem menggunakan metode *User Acceptance Test* (UAT). Pengujian akan dilakukan di sisi pengguna yang akan menggunakan sistem. Pengguna yang dimaksud adalah *stakeholders* terkait dalam tugas akhir ini yaitu sekretaris gereja dan jemaat. Pengujian didasarkan pada kriteria – kriteria yang tercantum dalam dokumen kontrak evaluasi yang disepakati oleh pengembang dan *stakeholders* terkait. Pengembang akan memberikan arahan pengujian dengan *stakeholders*, kemudian pengujian dilakukan secara mandiri berdasarkan skenario pengujian dalam dokumen pengujian. *Output* dari pengujian ini adalah berupa masukan dari peneliti mengenai aplikasi web yang digunakan dan dilakukan perencanaan kembali jika terdapat hal yang perlu diperbaiki atau dikembangkan.

3.6 Tahap Peluncuran

Pada tahap peluncuran aplikasi web sistem informasi HKBP Manyar akan menggunakan layanan hosting dari layanan firebase bernama *Hosting*. Alasan memilih layanan Firebase karena Firebase memiliki beberapa keunggulan utama yaitu, memiliki koneksi yang

aman, bisa melakukan *hosting* konten secara dinamis dan dapat mengirim konten dengan cepat. Dengan keunggulan yang dimiliki sangat cocok dengan pengembangan aplikasi web dengan teknik SPA. Dengan memilih layanan Firebase *Hosting*, aplikasi web sistem informasi HKBP Manyar akan memiliki waktu muat yang cepat, kestabilan yang tinggi, dan ketersediaan yang handal. Diharapkan bahwa menggunakan Firebase Hosting sebagai platform peluncuran akan memberikan keuntungan signifikan bagi aplikasi web sistem informasi HKBP Manyar.

3.7 Tahap Penulisan Laporan

Tahap terakhir yang akan dilakukan adalah membuat laporan pengerjaan. Seluruh proses perancangan dan pengembangan sistem informasi HKBP Manyar, dimulai dari kajian pustaka, akan didokumentasikan pada tahap ini, menghasilkan buku laporan tugas akhir.

3.8 Peralatan Pendukung

Pengembangan sistem informasi HKBP Manyar menggunakan lima alat bantu dan satu kerangka kerja. Alat-alat yang digunakan berupa perangkat lunak yang berguna dalam tahap perancangan desain, implementasi, pengujian dan peluncuran. Pada tahap perancangan desain menggunakan StarUML versi 5.0.1 dan Figma versi 116.15.7, pada tahap implementasi dan peluncuran menggunakan Visual Studio Code versi 1.86.2, Firebase versi 10.8.0, Github, dan Vue versi 3.2.39 sebagai kerangka kerja. Pada tahap pengujian menggunakan Wireshark versi 4.2.5 dan Google Chrome versi 126.

(Halaman ini sengaja dikosongkan)

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai langkah-langkah yang dilaksanakan dalam analisis kebutuhan, desain sistem, desain arsitektur, desain antarmuka, implementasi. Hasil dari pengujian ditampilkan dalam pengujian unit, pengujian integrasi, pengujian sistem, dan acceptance testing.

4.1 Analisa Kebutuhan

Analisis Kebutuhan Penerapan proses *waterfall* diawali dengan proses *requirement analysis*. Terdapat beberapa proses analisis kebutuhan yaitu proses elisitasi kebutuhan dan pembuatan Spesifikasi Kebutuhan.

4.1.1 Elisitasi Kebutuhan

Proses ini diawali melalui wawancara sekretaris gereja HKBP Manyar sebagai pemangku kepentingan dan satu anggota jemaat. Wawancara kebutuhan dilakukan dengan datang langsung ke narasumber. Sebelum melakukan wawancara ke pihak sekretaris gereja HKBP Manyar, dilakukan pertemuan pertama terlebih dahulu untuk menyusun jadwal wawancara dan sekretaris gereja meminta untuk dibuatkan surat sebagai bentuk formal dimulainya proyek pengembangan aplikasi web sistem informasi HKBP Manyar. Surat persetujuan untuk melakukan proyek dengan pihak pengurus HKBP Manyar dapat dilihat di Lampiran A.

Setelah dilakukan wawancara bersama sekretaris gereja dan salah satu jemaat, didapatkan beberapa hasil mengenai kebutuhan yang diinginkan pengguna. Hasil wawancara dengan sekretaris gereja bisa dilihat di Tabel 4.1 dan dengan anggota jemaat di Tabel 4.2.

Tabel 4. 1 Hasil Wawancara dengan Sekretaris Gereja HKBP Manyar

ID	Pertanyaan Wawancara	Respon Sekretaris Gereja
Q1.1	Bagaimana sistem informasi dapat membantu tugas sehari-hari Anda sebagai sekretaris?	Banyak program kerja yang dijalani oleh pengurus gereja dari masing-masing dewan dan ini berlangsung bisa setiap hari. Sebuah sistem informasi bisa sangat membantu dewan tersebut dalam menjalankan program kerja tersebut jika dapat diakses dan diolah setiap hari. Saat ini pencatatan informasi secara manual, dilakukan dalam bentuk penulisan pada kertas. Jika, ada sebuah sistem yang bisa merangkum semua kegiatan organisasi gereja bisa sangat membantu dari pengurus dan anggota gereja.
Q1.2	Apakah dalam pencatatan secara manual pernah mengalami kendala seperti kertas tersebut hilang?	Sebelumnya tidak pernah.
Q2	Apa saja tantangan atau hambatan yang sering dihadapi dalam pengelolaan informasi di organisasi?	Sering terjadi kendala kecil yang didapat saat melakukan pengolahan data karena sulitnya mengumpulkan informasi yang tersimpan di masa lalu.
Q3.1	Apakah ada proses atau kegiatan spesifik yang Anda rasa dapat ditingkatkan efisiensinya melalui sistem informasi?	Proses penyebaran informasi saat ini masih melakukan penyebaran dari warta jemaat yang diumumkan setiap minggu. Saat ini sudah dibuat grup yang berisikan pengurus dan anggota melalui aplikasi <i>WhatsApp</i> , biasanya penyebaran informasi juga dilakukan dari grup tersebut.
Q3.2	Dari penyebaran informasi yang dilakukan melalui <i>WhatsApp</i> apakah sudah dirasa cukup?	Belum, karena sering kali penyebaran informasi yang dilakukan bulan lalu sulit diakses kembali pada saat ini karena pengurus dan anggota kesulitan untuk mencari pesan tersebut. Dan terkadang informasi yang diberikan belum sempat diketahui oleh anggota karena banyaknya chat yang bertumpuk.
Q4.1	Bagaimana proses pelaporan dan dokumentasi saat ini, dan apa yang dapat ditingkatkan?	Mungkin bisa dalam pembuatan sistem informasi dalam bentuk aplikasi web.
Q4.2	Apakah ada referensi yang pernah dilihat dari sistem informasi dalam aplikasi web?	<i>Website</i> HKBP Tebet, berisikan informasi yang mencakup struktur dewan yang ada pada organisasi dan informasi yang dibutuhkan oleh anggota untuk mengetahui kegiatan yang ada pada organisasi
Q5.1	Apakah pernah mengetahui ada sistem informasi yang bisa dibuat dalam bentuk aplikasi web dan bisa mempermudah dalam menyebarluaskan informasi?	Ya, ini merupakan Solusi yang bagus, dimana informasi yang saat ini manual bisa diterjemahkan ke dalam bentuk web. Ini bisa menggait semua anggota supaya mengetahui informasi pada HKBP manyar dengan lebih mudah.
Q5.2	Apa saja fitur yang ingin ditampilkan dalam aplikasi web ini?	Struktur organisasi setiap dewan dan fungsinya masing-masing, Semua kegiatan yang ada dalam satu minggu, dan rekap aktivitas yang akan dijalankan dalam satu bulan kalender. Kalender program harus jelas ditampilkan, tata ibadah setiap minggu dan juga tata ibadah partangiangan.

Tabel 4. 2 Hasil Wawancara dengan Anggota Jemaat HKBP Manyar

ID	Pertanyaan Wawancara	Respon Anggota Jemaat Gereja
Q1.1	Bagaimana Anda berinteraksi dengan informasi yang disediakan oleh gereja saat ini, terutama melalui media digital?	Saya sebagai anggota jemaat sementara karena saya merupakan mahasiswa yang merantau dan selama di Surabaya melakukan ibadah di HKBP Manyar. Interaksi saya dengan informasi yang ada di gereja biasanya saya dapat selama seminggu atau biasanya saat saya gereja. Informasi yang saya dapat berupa pengumuman yang disampaikan saat warta jemaat.
Q1.2	Bagaimana penyampaian warta sebelumnya?	Biasanya warta dibacakan saat agenda warta jemaat dan ada power point yang menampilkan warta selama pembacaan warta.
Q1.3	Apakah tidak ada kertas warta yang dibagikan kepada jemaat?	Dulu ada kertas warta yang dibagikan, biasanya sebulan sekali untuk rekap pengumuman per bulan. Namun saat ini tidak ada
Q2	Apakah ada harapan atau kebutuhan khusus dari jemaat terkait penyampaian informasi dan aktivitas gereja melalui platform digital?	Ya. Jika ada alat yang dapat mempermudah saya atau jemaat yang lain dalam memperoleh informasi yang ada di gereja manyar ini, itu lebih baik. Contohnya dengan pembuatan <i>website</i> gereja.
Q3	Apakah ada ide atau saran untuk fitur-fitur tertentu yang dapat disertakan dalam aplikasi web untuk memenuhi kebutuhan jemaat?	Sepertinya cukup dengan menampilkan informasi umum saja seperti jadwal ibadah gereja, bacaan firman harian, warta mingguan. Dan mungkin karena saya pernah menjadi pengurus dalam organisasi gereja juga yaitu Naposo dan Guru Sekolah Minggu yang dimana keduanya merupakan organisasi dibawah dewan Koinonia, mungkin lebih bagus pada aplikasi <i>web</i> nya nanti ada berita yang menyajikan aktivitas atau kegiatan yang sudah dilakukan pengurus tersebut.
Q4	Bagaimana preferensi Anda terkait desain antarmuka dan pengalaman pengguna (<i>user experience</i>) dalam menggunakan aplikasi web gereja?	<i>Website</i> nya mudah digunakan, tidak rumit, dalam hal tata letak juga yang rapi. Bisa menyajikan informasi yang terbaru. Dan halaman admin nya mungkin bisa dipermudah supaya pengurus gereja nantinya mudah menggunakannya. Pemilihan warna dan design <i>website</i> nya juga harus bagus.

Setelah mendapatkan informasi dari calon pengguna aplikasi, dilakukan analisis kebutuhan kembali dengan studi literatur sistem informasi gereja serupa di internet. Didapatkan hasil studi literatur di Tabel 4.3.

Tabel 4. 3 Hasil Studi Literatur Sistem Informasi Serupa

No	Nama Gereja	Alamat <i>website</i>	Informasi yang ada di web
1	Gereja GMS	https://gms.id/home.php	<ul style="list-style-type: none"> - Agenda yang akan datang - Lokasi - Tentang gereja - Pendaftaran komunitas gereja
2	HKBP Depok	https://hkbpdepok1.org/	<ul style="list-style-type: none"> - Jadwal Ibadah - Warta jemaat mingguan - Video renungan harian - Dewan gereja (Marturia, Koinonia, Diakonia) - Galeri kegiatan gereja
3	HKBP Bogor	https://www.hkbpbogor.org/	<ul style="list-style-type: none"> - Daftar pengurus gereja - Dokumentasi acara - Dewan gereja - Jadwal ibadah - Kontak gereja - Kode QR untuk persembahan
4	HKBP Rawamangun	https://www.hkbprawamangun.id/	<ul style="list-style-type: none"> - Pengurus ibadah - Bacaan renungan harian - Artikel - Fungsi Dewan, pengertian dewan, dan pengurus yang ada di dewan itu. - Data Sintua

4.1.2 Spesifikasi Kebutuhan Perangkat Lunak

Setelah proses wawancara, dilakukan proses analisa dan pembuatan kebutuhan fungsional dalam website sistem informasi HKBP Manyar. Proses pertama yaitu dengan melakukan analisa kebutuhan utama. Proses ini memberikan hasil dalam bentuk tabel kebutuhan seperti Tabel 4.4 mengenai kebutuhan utama.

Tabel 4. 4 Kebutuhan Utama

ID	Kebutuhan Utama
M01	Jemaat dapat melihat informasi pada halaman web
M02	Pengurus dapat melakukan penambahan, pembaruan, dan penghapusan informasi

Dari hasil tersebut dapat dipecah menjadi dua bagian mengenai kebutuhan fungsional dan non fungsional. Kebutuhan fungsional berfungsi dalam memberikan rancangan kebutuhan perangkat lunak seperti pembuatan fungsi, proses input output dan interaksi yang lainnya. Sedangkan kebutuhan non fungsional dirumuskan dengan meninjau kebutuhan utama yang sudah dibuat. Kebutuhan fungsional dan non fungsional dapat dijelaskan di Tabel 4.5

Tabel 4. 5 Daftar Kebutuhan Fungsional dan Non fungsional HKBP Manyar

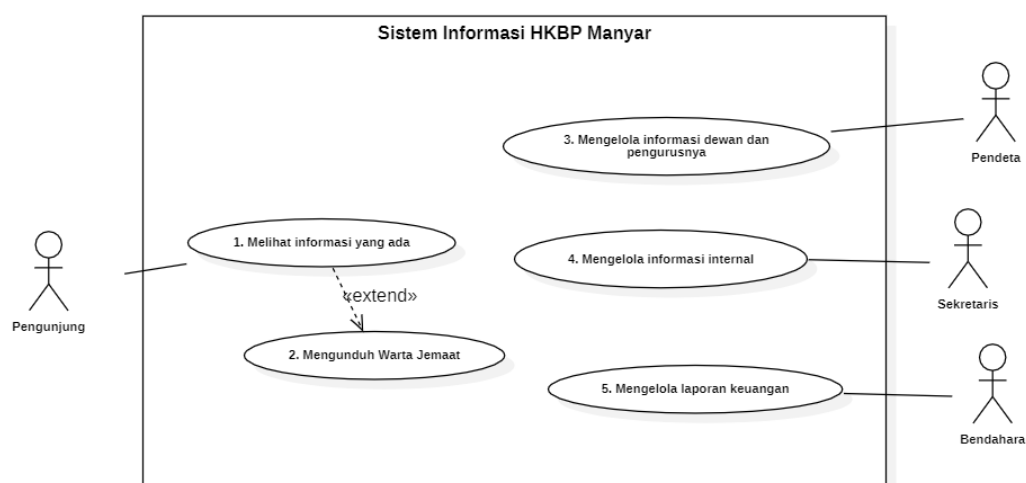
<i>Non fungsional</i>	
Analisa Kebutuhan	
SIHMAN dapat melakukan	
1.	Waktu muat halaman yang cepat.
2.	Penggunaan <i>resource</i> lebih hemat
3.	Fitur admin hanya bisa digunakan untuk pengguna yang sudah login
<i>Fungsional</i>	
Analisa Kebutuhan	
SIHMAN dapat melakukan	
1	Menampilkan profil gereja, visi, misi, dan sejarah gereja
2	Menampilkan informasi kontak gereja dan lokasi.
3	Menampilkan warta jemaat mingguan dalam bentuk teks dan PDF.
4	Menampilkan jadwal ibadah mingguan dan bulanan.
5	Menampilkan tata ibadah minggu pada sesi pagi, siang, dan sore dan tata ibadah rabu
6	Menampilkan agenda kegiatan gereja yang akan datang.
7	Menyediakan bacaan firman Tuhan yang diperbarui setiap hari
8	Informasi mengenai Dewan Marturia, Koinonia, dan Diakonia, termasuk pengurus dan fungsi mereka.
9	Menampilkan daftar pengurus gereja beserta kontak mereka.
10	Menyediakan informasi tentang berbagai komunitas gereja seperti Naposo dan Guru Sekolah Minggu.
11	Menampilkan laporan keuangan mingguan berupa data uang yang masuk dan keluar.
12	Menampilkan berita dan laporan kegiatan yang dilakukan oleh pengurus komunitas.
13	Menampilkan Formulir pendaftaran untuk bergabung dengan komunitas gereja.
14	Menyediakan dokumentasi dari acara-acara besar gereja.
15	Menampilkan link video streaming ibadah mingguan
16	Menyediakan kode QR untuk memudahkan jemaat memberikan persembahan secara digital.
17	Fitur untuk menambah, mengubah, dan menghapus akun admin.
18	Memberikan peran dan hak akses berbeda berdasarkan tingkat administrasi (pengurus seperti pendeta, sekretaris, atau bendahara).
19	Menampilkan halaman login dan logout
20	Hanya admin yang dapat mengelola informasi yang ada di halaman web
21	Ada perbedaan halaman antara pengguna yang tidak login dengan yang login.
22	Menambah, mengubah, menghapus, dan mempublikasikan warta jemaat mingguan.
23	Menambah, mengubah, dan menghapus jadwal ibadah mingguan dan bulanan.
24	Mengelola konten bacaan firman Tuhan harian.
25	Menambah, mengubah, dan menghapus berita kegiatan komunitas dan organisasi gereja.
26	Mengelola foto dan video di galeri kegiatan gereja.
27	Menambah, mengubah, dan menghapus agenda kegiatan gereja.
28	Mengelola dokumentasi acara gereja (foto, video, artikel).
29	Menambah, mengubah, dan menghapus informasi terkait komunitas gereja.
30	Menambah, mengubah, dan menghapus informasi dewan gereja beserta anggotanya.

Dari kebutuhan fungsional dan non fungsional yang sudah dirumuskan, dilakukan perancangan fungsi apa saja yang akan dikembangkan dalam aplikasi sistem informasi HKBP Manyar. Dalam spesifikasi kebutuhan perangkat lunak yang sudah dibuat fungsi yang akan dikembangkan dapat dilihat di Tabel 4.6.

Tabel 4. 6 Fungsi Aplikasi HKBP Manyar

ID	Fungsi
F01	Melihat informasi yang ada
F02	Mengunduh Warta Jemaat
F03	Mengelola informasi dewan dan pengurusnya
F04	Mengelola informasi internal
F05	Mengelola laporan keuangan

Dari fungsi aplikasi yang sudah dibuat terdapat aktor yang memiliki peran dalam melakukan fungsi tersebut. Terdapat dua aktor utama yaitu anggota jemaat dan admin yang merupakan pengurus organisasi. Pengurus organisasi ini merupakan pengurus meliputi pimpinan jemaat (pendeta), sekretaris, dan bendahara. Ketua Dewan Marturia, Diakonia, Koinonia, dan seksi-seksi yang ada di dalamnya tidak termaksud dalam admin. Untuk penjelasan tentang fungsi yang dilakukan aktor digambarkan di Gambar 4.1.



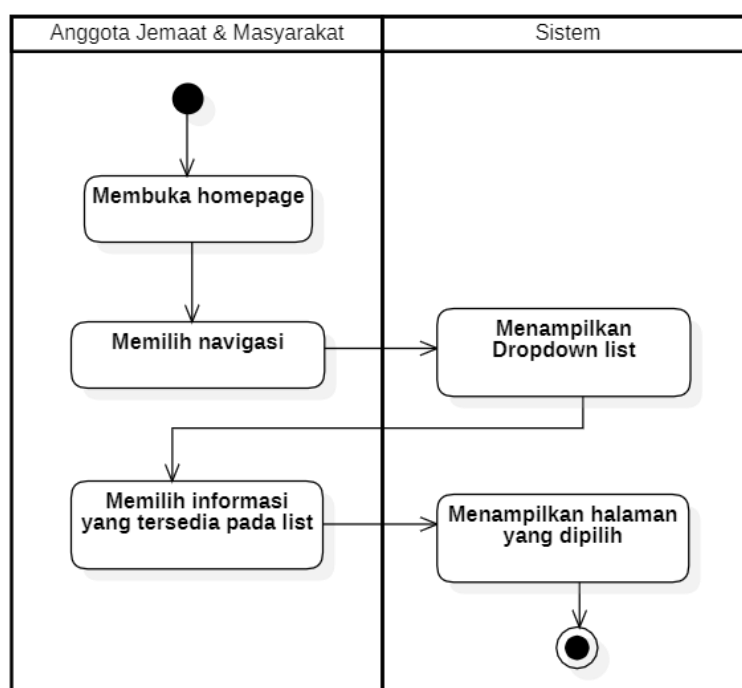
Gambar 4. 1 Use Case Diagram Sistem Informasi HKBP Manyar

Perincian pada kasus penggunaan dapat dilakukan dengan menggunakan pendekatan yang lebih ilmiah dan terstruktur. Perincian tersebut dilakukan untuk mendapatkan kebutuhan spesifikasi kasus. Rincian tersebut merupakan informasi tentang skenario, diagram aktivitas, dan diagram sekuens pada masing-masing fungsi. Spesifikasi kasus menjelaskan mengenai target, kondisi awal, kondisi akhir saat sukses, kondisi akhir saat gagal, actor utama, trigger, dan langkah dalam melakukan *use case*.

Pembuatannya sudah dilakukan dalam SKPL dan salah satu contoh rincian spesifikasi kasus adalah melihat informasi tentang HKBP Manyar, skenario terdapat di Tabel 4.7, diagram aktivitas di Gambar 4.2, dan diagram sekuens di Gambar 4.3. Penjelasan tentang semua skenario, diagram aktivitas, dan diagram sekuens terdapat di Lampiran B.

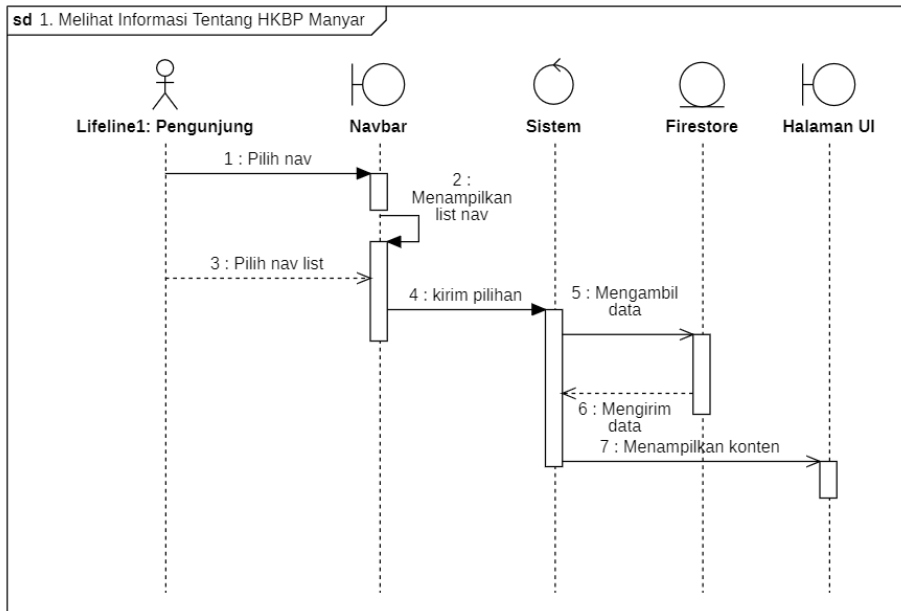
Tabel 4. 7 Skenario melihat informasi yang ada

Komponen	Deskripsi
Nama Use Case	Melihat Informasi yang ada
Nomor	UC-01
Deskripsi	Aktor dapat melihat informasi yang ada pada web
Aktor	Pengunjung
Kondisi Awal	Aktor berada di beranda HKBP Manyar
Kondisi Akhir	Sistem menampilkan informasi yang dibutuhkan aktor
Alur Normal	<ol style="list-style-type: none"> 1. Aktor masuk halaman beranda 2. Aktor memilih navigasi yang ada tersedia 3. Sistem menampilkan pilihan informasi yang tersedia 4. Aktor memilih informasi yang tersedia 5. Sistem menampilkan halaman informasi yang dipilih aktor
Alur Alternatif	-



Gambar 4. 2 Diagram Aktivitas Melihat Informasi HKBP Manyar

Dalam proses analisa kebutuhan terdapat rancangan untuk melakukan pengujian terhadap aplikasi. Analisa diperlukan sebelum proses implementasi supaya dapat menilai implementasi yang dibuat sudah berhasil atau tidak. Hal ini memiliki arti proses analisa kebutuhan memerlukan rancangan tolak ukur keberhasilan yang digunakan di tahap *User Acceptance Testing*. Rancangan *Unit Acceptance Testing* (UAT) merupakan pengujian yang digunakan sebagai tolak ukur keberhasilan *website*. Tabel 4.8 merupakan salah satu contoh rancangan pengujian aplikasi web. Dokumen tersebut menjelaskan mengenai data kebutuhan fungsional dan non fungsional dalam pengujian *website*, dan skenario pengujian pada setiap UAT. Skenario pengujian meliputi deskripsi pengujian, tahapan dalam pengujian, dan keterangan ekspektasi dalam pengujian. Rancangan UAT tertulis di Lampiran C.

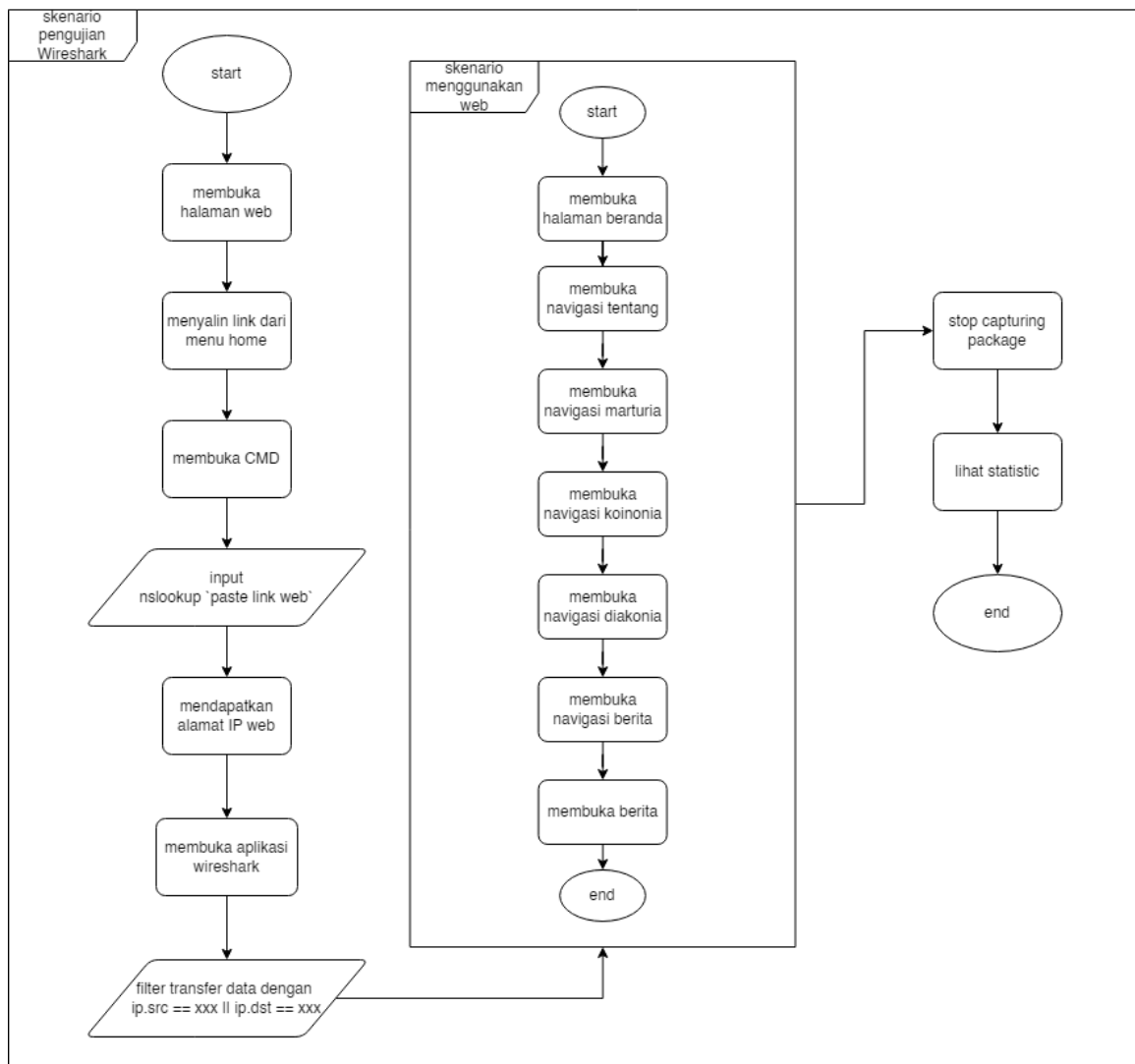


Gambar 4. 3 Diagram Sekuens Melihat Informasi HKBP Manyar

Tabel 4. 8 Contoh Dokumen User Acceptance Testing (UAT)

ID	Deskripsi Pengujian	Deskripsi Pengujian	Keterangan
UAT01	Jemaat dapat melihat informasi HKBP	<ul style="list-style-type: none"> Masuk pada halaman utama Klik beranda Lihat informasi yang dibutuhkan 	Jemaat dapat melihat informasi seputar HKBP Manyar
UAT02	Jemaat dapat melihat informasi dewan Koinonia	<ul style="list-style-type: none"> Masuk pada halaman utama Klik navigasi Koinonia Pada dropdown klik dewan Koinonia 	Jemaat dapat melihat informasi dewan Koinonia meliputi tanggung jawab, program kerja, dan struktur kepengurusan
UAT03	Jemaat dapat melihat informasi seksi yang berada di dewan Koinonia	<ul style="list-style-type: none"> Masuk pada halaman utama Klik navigasi Koinonia Pada dropdown klik seksi ada 	Jemaat dapat melihat informasi seksi yang berada di dewan Koinonia meliputi pengertian, program, dan pengurus.
UAT04	Jemaat dapat melihat informasi dewan Marturia	<ul style="list-style-type: none"> Masuk pada halaman utama Klik navigasi Marturia Pada dropdown klik dewan Marturia 	Jemaat dapat melihat informasi dewan Marturia meliputi tanggung jawab, program kerja, dan struktur kepengurusan

Pada tahap pengujian yang pertama yaitu pengujian performa aplikasi. Ini merupakan pengujian non fungsional, tetapi merupakan tujuan penting dalam tugas akhir ini. Pengujian performa dilakukan dengan menggunakan aplikasi Wireshark. Pengujian dilakukan menggunakan dua aplikasi web serupa dengan pengembangan MPA dan aplikasi web HKBP Manyar yang dibangun dengan SPA. Dilakukan perbandingan sebanyak dua kali dari web MPA satu dengan web HKBP manyar dan web MPA dua dengan web HKBP Manyar. Supaya hasil perbandingan valid maka menu yang dibuka oleh web MPA dan SPA harus sama. Skenario ini akan menjadi skenario dengan ID UAT 23. Skenario dari pengujian ini dapat dilihat di Gambar 4.4.

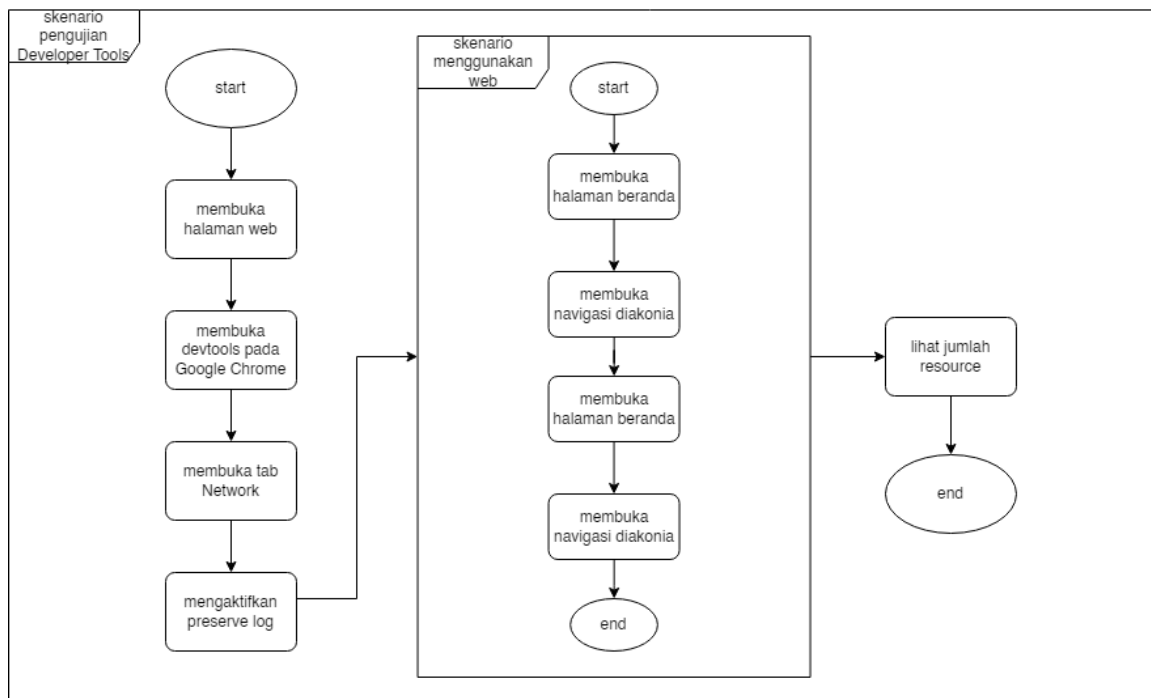


Gambar 4. 4 Alur Pengujian Performa Web Menggunakan Wireshark

Pengujian performa aplikasi menjadi fokus utama dari Tugas Akhir ini. Hasil uji yang didapatkan dari masing-masing pengujian akan dibandingkan dalam sebuah tabel perbandingan. Pengujian juga dilakukan menggunakan developer tools pada Google Chrome. Alur pengujian menggunakan developer tools memiliki sedikit perbedaan dengan pengujian menggunakan Wireshark. Alur pengujian menggunakan developer tools dapat dilihat di Gambar 4.5.

Dari alur pengujian performa menggunakan Developer Tools dapat diketahui data berupa jumlah *request*, data transfer, dan *resource* yang digunakan. Dalam pengujian, kotak preserve log diaktifkan supaya saat halaman diarahkan ke link pada navigasi lain, tidak menghapus catatan pengujian. Sehingga, saat pengujian dilakukan berulang bisa mendapatkan total hasil *resource* yang terpakai.

Dari konsep SPA seharusnya komponen yang sudah dimuat tidak akan dimuat ulang yang membuat penggunaan *resource* menjadi lebih hemat. Untuk pengujian *resource* dibuat menjadi skenario ID UAT 24.



Gambar 4. 5 Alur Pengujian Performa Web Menggunakan Developer Tools

Dalam dokumentasi hasil pengujian, website memerlukan data berupa traceability matrix untuk melacak hubungan antara prasyarat pengguna dan skenario pengujian UAT yang relevan. *Traceability matrix* ini memberi gambaran jelas mengenai hubungan antara kebutuhan fungsional dan pengujian UAT. *Traceability* ini juga dapat meminimalisir resiko suatu pengujian yang tidak dapat diidentifikasi atau tidak ditemukan. Tabel 4. 10 merupakan rancangan dari *traceability matrix* dari tahap analisis kebutuhan.

4.2 Perancangan Desain

Langkah pertama dalam perancangan adalah desain database. Untuk memastikan agar proses penyaluran data pada sistem tetap aman dan sesuai, data divisualisasikan dalam pemodelan data. Sistem ini menggunakan basis data Firebase sebagai salah satu basis data berjenis NoSQL yang tidak memerlukan adanya relasi antar entitas. Sehingga, basis data lebih fleksibel terhadap perubahan struktur data pada sistem. Pada SQL, setiap tabel data disebut dengan entitas. Namun, dalam NoSQL, setiap tabel data disebut dengan koleksi. Daftar koleksi yang dibutuhkan dalam sistem aplikasi web HKBP Manyar dirangkum dalam Tabel Koleksi seperti di Tabel 4.9. Daftar koleksi tersebut disusun untuk memastikan semua data yang diperlukan telah diidentifikasi dan dikelola dengan baik.

Tabel 4. 9 Data Koleksi pada Aplikasi Web HKBP Manyar

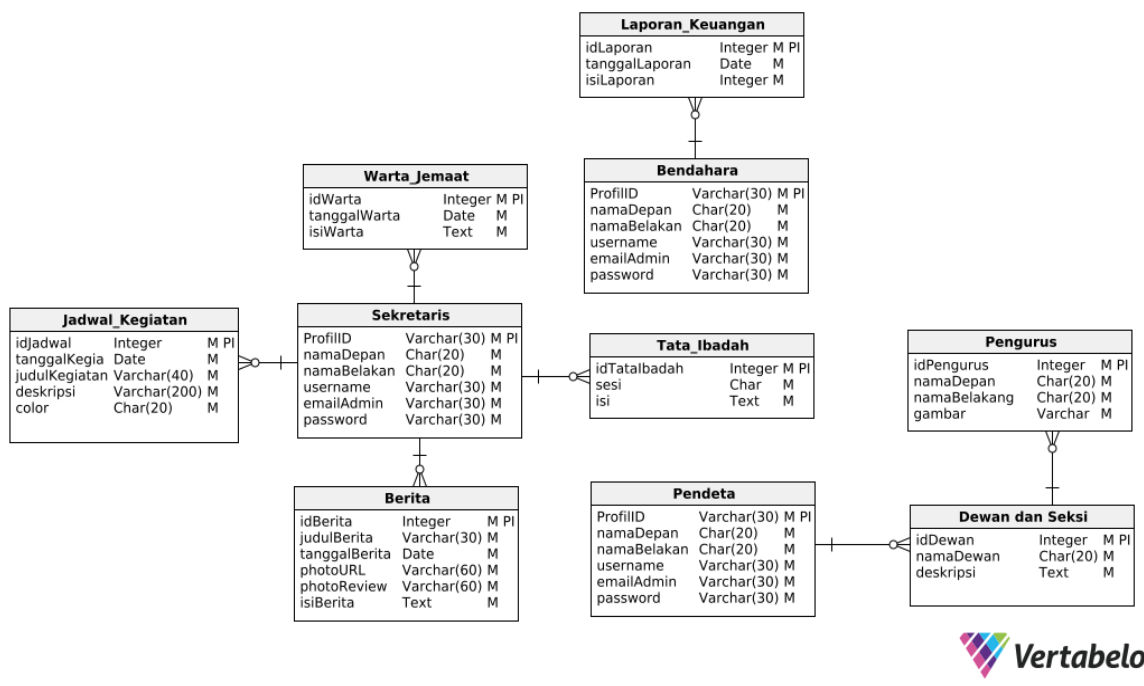
ID	Koleksi	Deskripsi
C01	User	Koleksi untuk menyimpan data pengguna yang merupakan admin
C02	CourosellItem	Koleksi untuk menyimpan data gambar pada halaman utama.
C03	Event	Koleksi untuk menyimpan data detail jadwal kegiatan
C04	Keuangan	Koleksi untuk menyimpan data keuangan
C05	Seksi	Koleksi untuk menyimpan data dewan dan seksi berupa konten dan pengurus dewan atau seksi
C06	WartaJemaat	Koleksi untuk menyimpan data Warta jemaat
C07	Blog	Koleksi untuk menyimpan data berita

Tabel 4. 10 *Traceability Matrix* UAT

ID Kebutuhan	Deskripsi Kebutuhan	ID Testing	Dekripsi User Acceptance Testing
F01	Melihat informasi tentang HKBP Manyar	UAT01	Jemaat dapat melihat informasi HKBP
		UAT02	Jemaat dapat melihat informasi dewan Koinonia
		UAT03	Jemaat dapat melihat informasi seksi yang berada di dewan Koinonia
		UAT04	Jemaat dapat melihat informasi dewan Marturia
		UAT05	Jemaat dapat melihat informasi seksi yang berada di dewan Marturia
		UAT06	Jemaat dapat melihat informasi dewan Diakonia
		UAT07	Jemaat dapat melihat informasi seksi yang berada di dewan Diakonia
		UAT08	Jemaat dapat melihat informasi keuangan gereja
		UAT10	Jemaat dapat melihat tata ibadah
		UAT11	Jemaat dapat melihat jadwal kegiatan
		UAT12	Jemaat dapat melihat berita dan artikel
		UAT13	Jemaat dapat mencari atau memfilter berita
		F02	Mengunduh warta jemaat
F03	Mengelola informasi dewan dan pengurusnya	UAT14	Pendeta dapat mengubah informasi dewan dan pengurusnya
F04	Mengelola informasi internal	UAT15	Sekretaris dapat mengelola tata ibadah
		UAT16	Sekretaris dapat mengelola warta jemaat
		UAT17	Sekretaris dapat mengelola jadwal kegiatan
		UAT18	Sekretaris dapat menambah berita
		UAT19	Sekretaris dapat mengedit berita
		UAT20	Sekretaris dapat melihat preview berita
F05	Mengelola laporan keuangan	UAT21	Bendahara dapat mengelola laporan keuangan
NF01	Waktu muat halaman yang cepat.	UAT23	Menganalisis dengan Wireshark
NF02	Penggunaan <i>resource</i> lebih hemat	UAT24	Melakukan skenario menggunakan <i>website</i> sebanyak dua kali dan di analisis dengan Wireshark
NF03	Fitur admin hanya bisa digunakan untuk pengguna yang sudah login	UAT24	Pendeta bisa login sebagai role pendeta
		UAT25	Sekretaris bisa login sebagai role Sekretaris
		UAT26	Bendahara bisa login sebagai role bendahara

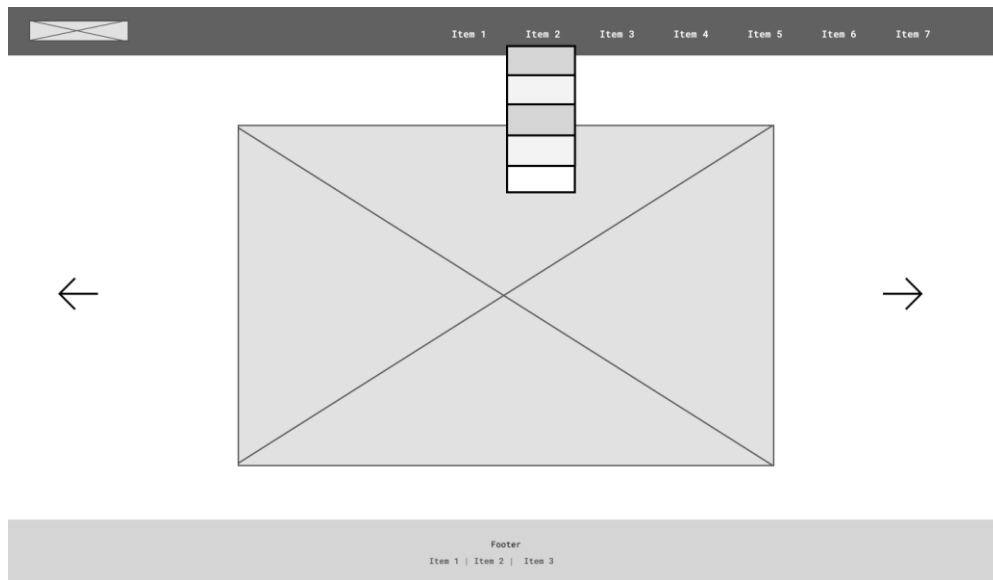
Sistem ini menggunakan pemodelan data bukan *Entity Relationship Diagram* (ERD) seperti pada basis data SQL pada umumnya sebagai visualisasi data. Namun, untuk menjaga dan mengetahui lebih rinci terkait proses pertukaran data pada sistem aplikasi web HKBP Manyar ini, pemodelan data menggunakan kombinasi antara model data ERD dengan diagram model NoSQL seperti di Gambar 4.6. Jenis model data ERD yang digunakan adalah *Conceptual Data Model* (CDM) yang merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data. CDM dibuat sudah dalam bentuk koleksi tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke database. Diagram di Gambar 4.6 memiliki berbagai tabel data.

Penggunaan Firebase sebagai database dapat memudahkan pengelolaan data menggunakan fitur *Firestore Database*. Penyimpanan ke *database* ini berupa data dengan tipe *string*, *number*, *boolean*, *map*, *array*, *null*, dan *timestamp*. Pada aplikasi Web HKBP Manyar juga harus dapat menyimpan data berupa gambar maupun file PDF. Sehingga, digunakan fitur berupa Storage yang disediakan oleh Firebase supaya dapat menyimpan data gambar dan file tersebut. Setelah disimpan dalam Storage gambar atau file tersebut akan mempunyai link supaya dapat diakses dan Firestore akan menyimpan link tersebut supaya dapat diakses. Integrasi Firebase memastikan sistem informasi HKBP Manyar dapat dibangun dengan handal dan responsif.



Gambar 4. 6 Pemodelan *Database* Sistem Informasi HKBP Manyar

Perancangan desain selanjutnya adalah perancangan antarmuka. Pembuatan antarmuka yang dibuat merupakan sebuah *wireframe*. *Wireframe* adalah panduan visual tentang tampilan suatu produk. Tujuan utama mereka adalah membuat aplikasi atau situs yang memberikan pengalaman kohesif dan dirancang dengan baik. Pembuatan *Wireframe* dapat membantu dalam tahap implementasi atau kode. Contoh pembuatan antarmuka adalah antarmuka Layar Utama tampilan nya bisa dilihat di Gambar 4.7. Spesifikasi antarmuka dapat dilihat di Gambar dan spesifikasi elemen-elemen pada layar dapat dilihat di Tabel 4.11.



Gambar 4. 7 Tampilan Layar Utama *Home*

Tabel 4. 11 Spesifikasi Elemen-elemen Layar *Home*

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Teks HKBP Manyar	<i>Button</i>	Mengarahkan ke halaman utama	<i>ButtonClick</i>
2	Nav Tentang	<i>Button</i>	Mengarahkan ke halaman tentang	<i>ButtonClick</i>
3	Nav Diakonia	<i>Button</i>	Membuka dropdown nav Diakonia	<i>ButtonClick</i>
4	Nav Koinonia	<i>Button</i>	Membuka dropdown nav Koinonia	<i>ButtonClick</i>
5	Nav Marturia	<i>Button</i>	Membuka dropdown nav Marturia	<i>ButtonClick</i>
6	Nav Berita	<i>Button</i>	Mengarahkan ke halaman berita	<i>ButtonClick</i>
7	Dropdown	<i>Dropdown</i>	Menampilkan pilihan halaman pada navigasi yang dipilih	<i>ButtonClick</i>
8	Courosel	<i>Image</i>	Menampilkan informasi penting	<i>ButtonClick</i>
9	Right and Left icon	<i>icon</i>	Menggeser courosel	<i>ButtonClick</i>
10	Footer	<i>Teks, Button</i>	Menampilkan informasi Alamat dan link yang bisa diakses	<i>Teks and ButtonClick</i>

Tampilan antarmuka dan juga spesifikasi elemen-elemen layar dari komponen halaman yang dibuat selengkapnya terdapat di Lampiran D. Dari pembuatan perancangan antarmuka dan spesifikasi elemen-elemen pada layar akan menjadi pedoman dalam melakukan implementasi untuk menentukan letak komponen-komponen.

4.3 Implementasi

Implementasi merupakan fase pembuatan kode aplikasi web Sistem Informasi HKBP Manyar. Aplikasi web ini menggunakan pendekatan *Single Page Application* (SPA) dengan Vue.js sebagai kerangka kerja utama untuk membangun antarmuka yang dinamis dan responsif. Firebase digunakan sebagai *Backend as a Service* (BaaS) untuk mengelola data dengan *Firestore* sebagai database NoSQL dan *Firebase Storage* untuk menyimpan gambar serta *file* PDF. *Pinia* dipilih sebagai pengelola *state* (*store*) dalam aplikasi ini, memungkinkan manajemen *state* yang efisien dan terstruktur. Dengan menggunakan kombinasi Vue.js, Firebase, dan *Pinia*, Sistem Informasi HKBP Manyar dapat dibangun dengan handal, responsif, dan mudah dipelihara, serta memberikan pengalaman pengguna yang optimal.

4.3.1 Pembuatan Satu Halaman HTML

Pembuatan satu halaman HTML menjadi dasar dari pengembangan SPA. Saat meng-*install* Vue.js pertama kali file ini akan disiapkan secara otomatis. Isi dari kode HTML ini hanya berupa pemanggilan *script.js* yang nantinya akan menginisiasi komponen-komponen yang ada untuk dipanggil. Pada pembuatan kode ini menggunakan *framework* *Vuetify*. *Vuetify* merupakan *framework* yang menyediakan *library* yang memungkinkan untuk membuat aplikasi dengan tampilan yang lebih indah. *Vuetify* memiliki lebih dari 80 komponen berbasis *Material Design* yang mana komponen tersebut dapat membantu supaya lebih cepat dalam mendesain aplikasi. Kode 4.1 memberikan contoh kode HTML yang dibuat pertama kali saat *Vuetify* di *install*.

```
1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <meta charset="UTF-8" />
5.     <link rel="icon" href="/src/assets/pic/HKBP.png" />
6.     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7.     <title>HKBP Manyar</title>
8.   </head>
9.
10.  <body>
11.    <div id="app"></div>
12.    <script type="module" src="/src/main.js"></script>
13.  </body>
14. </html>
```

Kode 4. 1 Dokumen HTML

Dalam implementasi kode HTML SPA, penggunaan *tag script* di baris 12 adalah kunci untuk memulai aplikasi Vue.js. Dokumen *main.js* ini bertindak sebagai titik masuk utama untuk aplikasi SPA, di mana aplikasi Vue diinisialisasi dan di-*mount* ke elemen HTML dengan id *app* di baris 11. Dengan struktur ini, HTML berfungsi sebagai kerangka dasar, sedangkan logika dan tampilan aplikasi diatur sepenuhnya oleh Vue.js melalui JavaScript *modular*.

4.3.2 Pembuatan Main Js

Halaman HTML pertama kali menjalankan skrip *main.js*. Pada dokumen JavaScript ini, *lifecycle hook mount* akan dipanggil sehingga komponen yang ada di *main.js* akan dijalankan. Komponen yang ada adalah *App.vue*. Kode 4.2 menampilkan halaman *main.js* dalam pembuatan *App.vue*.


```

1. import { registerPlugins } from "@plugins";
2.
3. import App from "./App.vue";
4.
5. import { createApp } from "vue";
6.
7. const app = createApp(App);
8.
9. registerPlugins(app);
10.
11. app.mount("#app");

```

Kode 4. 2 Dokumen Main.Js

4.3.3 Pembuatan App.vue sebagai komponen pertama

Komponen yang pertama kali hidup adalah App.vue. Pada pengembangan SPA, App.vue akan menjalankan 'router-view' supaya komponen-komponen yang ada dapat dinavigasikan. Kode App.vue dapat dilihat di Kode 4.3.

```

1. <template>
2.   <v-app>
3.     <AppNavigation v-if="!isLoginPage" />
4.     <router-view v-slot="{ Component }">
5.       <KeepAlive>
6.         <component :is="Component" />
7.       </KeepAlive>
8.     </router-view>
9.     <AppFooter v-if="!isLoginPage" />
10.   </v-app>
11. </template>

```

Kode 4. 3 Dokumen App.vue

Kode App.vue ini mendefinisikan struktur utama aplikasi dengan menggunakan Vuetify dan Vue Router. Bagian *v-app* membungkus seluruh aplikasi sebagai *root Vuetify component*. Komponen AppNavigation dan AppFooter hanya ditampilkan jika pengguna tidak berada di halaman *login*, yang dikontrol oleh properti *isLoginPage*. *Tag router-view* bertindak sebagai *placeholder* untuk komponen yang sesuai dengan rute saat ini. Komponen dinamis ini dibungkus dengan *KeepAlive*, yang memungkinkan komponen yang tidak aktif tetap hidup di memori, meningkatkan kinerja dengan menghindari *re-rendering* yang tidak perlu. Contoh dari *re-rendering* yang mungkin terjadi ketika komponen konten berita dibuka maka akan *re-trigger* untuk melakukan pengambilan data. *KeepAlive* membuat data yang sudah diambil tidak akan diambil ulang ketika ada *trigger* kembali aktif.

4.3.4 Pembuatan Router

Router dibuat supaya tampilan yang ada bisa dinavigasikan ke tujuan tertentu. Misalkan saat tombol yang ada di AppNavigation di klik maka tampilan akan diubah ke halaman yang sudah di tentukan. Pada pembuatan navigasi web HKBP Manyar diperlukan banyak komponen. Contoh pembuatan *routes* dapat dilihat di kode 4.4.

```

1. const router = createRouter({
2.   history: createWebHistory(import.meta.env.BASE_URL),
3.
4.   routes: [
5.     {
6.       path: "/",
7.       name: "beranda",
8.       component: Beranda,
9.       meta: {
10.        title: "Beranda",
11.      },
12.    },
13.    // skip code
262.   {
263.     path: "/profile",
264.     name: "profile",
265.     component: () => import("@/pages/admin/ProfilePage.vue"),
266.     meta: {
267.       title: "Profile",
268.       requiresAuth: true,
269.     },
270.   },
271.   {
272.     path: "/*",
273.     name: "not-found",
274.     component: () => import("@/pages/NotFoundPage.vue"),
275.     meta: {
276.       title: "Error",
277.     },
278.   },
279. ],
280. });
281.
282. router.beforeEach((to, from, next) => {
283.   document.title = `${to.meta.title} | HKBP Manyar`;
284.   next();
285. });
286.
287. // Auth Guard
288. router.beforeEach(async (to, from, next) => {
289.   const user = auth.currentUser;
290.
291.   if (to.matched.some((res) => res.meta.requiresAuth)) {
292.     if (user) {
293.       const userDoc = await getDoc(doc(db, "users", user.uid));
294.       const userData = userDoc.data();
295.       const userRole = userData?.role;
296.
297.       if (to.matched.some((res) => res.meta.role)) {
298.         if (to.meta.role === userRole) {
299.           return next();
300.         } else {
301.           return next({ name: "beranda" });
302.         }
303.       }
304.       return next();
305.     } else {
306.       return next({ name: "beranda" });
307.     }
308.   }
309.   return next();
310. });
311.
312. export default router;

```

Kode 4. 4 Dokumen *Router*

Di kode 4.4 menunjukkan konfigurasi beberapa rute, di baris 4-279 terdapat rute beranda, arsip warta, profil, dan halaman *not found*. Setiap rute memiliki *meta field* yang dapat digunakan untuk menyimpan informasi tambahan, seperti judul halaman, persyaratan autentikasi, dan *role* admin. Di baris 282-285, terdapat *beforeEach hook* yang digunakan untuk mengubah judul dokumen berdasarkan *meta title* dari rute yang dituju. Kemudian, ada *beforeEach hook* lain di baris 288-310 yang bertindak sebagai Auth Guard. Auth Guard ini memeriksa apakah rute yang dituju memerlukan autentikasi dan apakah pengguna saat ini telah terautentikasi. Jika rute memerlukan autentikasi dan pengguna belum *login*, pengguna akan diarahkan kembali ke beranda. Selain itu, jika rute memerlukan *role* tertentu dan *role* pengguna tidak sesuai, pengguna juga akan diarahkan kembali ke beranda. Hal ini memastikan bahwa hanya pengguna dengan izin yang sesuai yang dapat mengakses rute tertentu, menjaga keamanan dan akses yang tepat dalam aplikasi.

4.3.5 Pembuatan Komponen

Komponen dalam proyek ini dibagi menjadi dua jenis, yaitu komponen halaman utama yang ditampilkan secara keseluruhan, disimpan dalam folder '*pages*', dan komponen pendukung yang dapat dipanggil oleh komponen di folder '*pages*', disimpan dalam folder '*components*'. Penjelasan lengkap mengenai semua komponen di Folder '*pages*' terdapat di Lampiran E. Sebagai contoh, komponen halaman utama yang dijelaskan di Kode 4.5.

```
1. <template>
2.   <v-app>
3.     <CarouselCom />
4.     <JadwalIbadah />
5.     <TataIbadah />
6.     <KeuanganCom />
7.     <JadwalKegiatan />
8.   </v-app>
9. </template>
```

Kode 4. 5 Dokumen BerandaPage.vue

Di dalam komponen halaman utama BerandaPage.vue berfungsi untuk menampilkan halaman beranda. Halaman beranda menggunakan beberapa komponen Vue lain seperti, CouroselItem, JadwalIbadah, TataIbadah, KeuanganCom, dan JadwalKegiatan. Komponen yang dibuat memiliki fungsi yang berbeda-beda, contoh pembuatan komponen yaitu komponen CouroselCom yang memiliki fungsi menampilkan carousel atau slideshow gambar pada Home. Kode CouroselCom terdapat di Kode 4.6.

```

1. <template>
2.   <v-app>
3.     <v-main>
4.       <v-container fluid class="pa-0">
5.         <div
6.           v-if="authStore.profileRole === 'sekretaris'"
7.           class="d-flex flex-row-reverse"
8.         >
9.           <v-sheet class="pa-2 mr-12 mb-n6" :width="220">
10.            <div class="pa-3 mr-6">
11.              <span>Edit mode</span>
12.              <v-switch
13.                v-model="isEditMode"
14.                :label="`Switch: ${isEditMode}`"
15.                color="primary"
16.                hide-details
17.              ></v-switch>
18.            </div>
19.          </v-sheet>
20.        </div>
21.        <v-carousel
22.          height="90vh"
23.          show-arrows="hover"
24.          cycle
25.          hide-delimiter-background
26.          :interval="interval"
27.          :show-arrows-on-hover="true"
28.        >
29.          <v-carousel-item
30.            v-for="item in items"
31.            :key="item.id"
32.            :src="item.imageUrl"
33.            cover
34.          >
35.            <div class="title">{{ item.title }}</div>
36.            <v-icon v-if="isEditMode" @click="openUploadDialog(item.id)"
37.              >mdi-pencil</v-icon>
38.          >
39.        </v-carousel-item>
40.      </v-carousel>
41.      <v-dialog v-model="uploadDialog" max-width="500px">
42.        <v-card>
43.          <v-card-title>Upload Foto</v-card-title>
44.          <v-card-text>
45.            <v-file-input
46.              v-model="selectedFile"
47.              label="Pilih file"
48.              accept="image/*"
49.            ></v-file-input>
50.          </v-card-text>
51.          <v-card-actions>
52.            <v-btn color="primary" @click="uploadImage" :loading="isLoading"
53.              >Upload</v-btn>
54.            >
55.            <v-btn text @click="uploadDialog = false">Batal</v-btn>
56.          </v-card-actions>
57.        </v-card>
58.      </v-dialog>
59.    </v-container>
60.  </v-main>
61. </v-app>
62. </template>

```

Kode 4. 6 File CourouselCom

Terdapat komponen yang dapat digunakan pada komponen lainnya. Penjelasan tentang komponen yang dibuat di file `components` dijelaskan di Tabel 4.10 beserta dengan fungsi komponen tersebut.

Tabel 4. 12 Komponen pada File *Components* dan Fungsinya

ID	Nama Komponen	Fungsi
C01	AppFooter.vue	Menampilkan <i>footer</i> pada aplikasi.
C02	AppNavigation	Menampilkan navigasi pada aplikasi.
C03	CouroselCom.vue	Menampilkan carousel atau <i>slideshow</i> gambar pada <i>Home</i>
C04	ConfirmDeleteDialog.vue	Menampilkan dialog konfirmasi penghapusan item
C05	JadwalIbadah.vue	Menampilkan jadwal ibadah pada <i>Home</i>
C06	JadwalKegiatan.vue	Menampilkan jadwal kegiatan dalam bentuk kalender pada <i>Home</i>
C07	Kuangan.vue	Menampilkan rekapitulasi keuangan pada <i>Home</i>
C08	Loading.vue	Menampilkan animasi <i>loading</i> ketika terdapat proses <i>write</i> ke <i>database</i>
C09	Maintenance.vue	Menampilkan tampilan sedang terjadi <i>maintenance</i>
C10	ProfilTool.vue	Menampilkan profil yang ada di <i>Navigation</i> ketika sudah <i>login</i>
C11	SeksiContent.vue	Menampilkan tampilan informasi untuk dewan dan seksi berupa teks dan gambar pengurus
C12	Tatalibadah.vue	Menampilkan tata ibadah pada <i>Home</i>

4.3.6 Pembuatan *Stores*

Pada tahap pembuatan *store*, digunakan Pinia sebagai pengelola *state*. Dengan menggunakan *store*, data dapat disimpan sebagai *state* dan dapat memiliki *action* yang berisi logika atau pengambilan data dari *database*. Sebagai contoh, berikut adalah *store* untuk admin yang dibuat dan ditunjukkan di kode 4.7

Store ini memiliki penyimpanan *state* dijelaskan di baris 11-21 yang berisikan data *user*, *error*, *profileRole*, *profileEmail*, *profileFirstName*, *profileLastName*, *profileUserName*, *profilId*, dan *profilInitials*. Di dalam *store* ini juga terdapat *action* yang bisa dilihat di baris 22-93 yang berguna untuk mengatur hal yang berkaitan dengan admin. Baris 22-33 berfungsi untuk mendapatkan data pengguna saat ini dari *Firestore* dan memperbarui *state* dengan data profil, baris 34-36 berfungsi untuk mengatur *role* profil di *state*, baris 37-48 berfungsi untuk mengatur inisial profil di *state*, baris 49-62 berfungsi untuk *login* menggunakan *e-mail* dan *password*, serta memperbarui *state* dengan informasi pengguna, baris 63-72 berfungsi untuk *logout* dan menghapus informasi pengguna dari *state*, baris 73-83 berfungsi untuk menghapus informasi profil dari *state*, dan baris 85-93 sebagai pemantauan perubahan status autentikasi untuk memperbarui *state* berdasarkan status *login* pengguna.

Penjelasan tentang *store* yang dibuat dalam pengembangan web Sistem HKBP Manyar terdapat di Tabel 4.11 beserta fungsinya

Tabel 4. 13 *Store* yang Dapat Digunakan

ID	Nama Store	Fungsi
S01	admin.js	Mengatur <i>state</i> dan <i>action</i> pada admin yang <i>login</i>
S01	beranda.js	Mengatur <i>state</i> dan <i>action</i> pada menu <i>Home</i>
S03	blog.js	Mengatur <i>state</i> dan <i>action</i> pada menu berita
S04	seksi.js	Mengatur <i>state</i> dan <i>action</i> pada menu dewan dan seksi di dalamnya

```

1. import { defineStore } from "pinia";
2. import { auth, db } from "@/firebase/firebase";
3. import {
4.   signInWithEmailAndPassword,
5.   signOut,
6.   onAuthStateChanged,
7. } from "firebase/auth";
8. import { doc, getDoc } from "firebase/firestore";
9.
10. export const useAuthStore = defineStore("auth", {
11.   state: () => ({
12.     user: null,
13.     // skip code
21.   }),
22.   actions: {
23.     async getCurrentUser() {
24.       if (!auth.currentUser) return;
25.       const dataBase = doc(db, "users", auth.currentUser.uid);
26.       const dbResults = await getDoc(dataBase);
27.       if (dbResults.exists()) {
28.         this.setProfileInfo(dbResults.data());
29.         this.setProfileInitials();
30.         const role = dbResults.data().role;
31.         this.setProfileRole(role);
32.       }
33.     },
34.     setProfileRole(payload) {
35.       this.profileRole = payload;
36.     },
37.     setProfileInfo(data) {
38.       this.profileId = auth.currentUser.uid;
39.       this.profileEmail = data.email;
40.       this.profileFirstName = data.firstName;
41.       this.profileLastName = data.lastName;
42.       this.profileUsername = data.username;
43.     },
44.     setProfileInitials() {
45.       this.profileInitials =
46.         this.profileFirstName.charAt(0).toUpperCase() +
47.         this.profileLastName.charAt(0).toUpperCase();
48.     },
49.     async login(email, password) {
50.       // skip code
62.     },
63.     async logout() {
64.       // skip code
72.     },
73.     clearProfile() {
74.       // skip code
81.     },
82.   },
83. });
84.
85. onAuthStateChanged(auth, (user) => {
86.   const authStore = useAuthStore();
87.   if (user) {
88.     authStore.getCurrentUser();
89.     authStore.user = user;
90.   } else {
91.     authStore.clearProfile();
92.   }
93. });

```

Kode 4.7 Store Admin.js

4.3.7 Pembuatan Firebase

Kode ini bertujuan untuk menginisialisasi aplikasi Firebase dan mengkonfigurasi berbagai layanan Firebase yang akan digunakan dalam proyek, yaitu Firestore, Authentication, dan Storage. Penulisan kode dapat dilihat di Kode 4.7.

```
1. import { initializeApp } from 'firebase/app';
2. import { getFirestore } from 'firebase/firestore';
3. import { getAuth } from 'firebase/auth';
4. import { getStorage } from 'firebase/storage';
5.
6. const firebaseConfig = {
7.   apiKey: "xxx",
8.   authDomain: "xxx",
9.   databaseURL: "xxx",
10.  projectId: "xxx",
11.  storageBucket: "xxx",
12.  messagingSenderId: "xxx",
13.  appId: "xxx",
14. };
15.
16. const app = initializeApp(firebaseConfig);
17. const db = getFirestore(app);
18. const auth = getAuth(app);
19. const storage = getStorage(app);
20.
21. export {
22.   db,
23.   auth,
24.   storage
25. };
```

Kode 4. 8 File Firebase.js

Kode ini mengimpor dan menginisialisasi layanan Firebase untuk digunakan dalam aplikasi. Pertama, Firebase App diinisialisasi dengan konfigurasi yang diberikan di baris 6-14. Selanjutnya, tiga layanan Firebase diambil yaitu Firestore untuk database (baris 17), Authentication untuk autentikasi pengguna (baris 18), dan Storage untuk penyimpanan file (baris 19). Objek-objek ini kemudian diekspor agar dapat digunakan di bagian komponen yang memerlukan akses data pada Firebase. Dengan cara ini, aplikasi dapat terhubung dan menggunakan layanan-layanan Firebase yang diperlukan.

Secara keseluruhan, kode ini berfungsi untuk mengatur dan menghubungkan aplikasi dengan berbagai layanan Firebase, sehingga memungkinkan penggunaan database NoSQL (Firestore), sistem autentikasi pengguna, dan penyimpanan file dalam aplikasi. Dalam penulisan data ke dalam Firestore maupun Storage harus dibuat *rules* supaya keamanan aplikasi terjamin. Hanya admin saja yang dapat melakukan penulisan ke *database*, cara tersebut dapat dilakukan dengan penulisan *rules* di Firestore Database terdapat di Kode 4.8 dan penulisan rules Storage di Kode 4.9.

```
1. //Firestore Database
2. rules_version = '2';
3. service cloud.firestore {
4.   match /databases/{database}/documents {
5.     match /{document=**} {
6.       allow read;
7.       allow write: if request.auth.uid != null;
8.     }
9.   }
10. }
```

Kode 4. 9 Rules Pada Firestore Database

```

1. //Storage
2. service firebase.storage {
3.   match /b/{bucket}/o {
4.     match /{allPaths=**} {
5.       allow read;
6.       allow write: if request.auth != null;
7.     }
8.   }
9. }

```

Kode 4. 10 *Rules* pada Storage

Rules yang dibuat untuk Firestore Database dan Storage mengatur hak akses baca dan tulis untuk memastikan keamanan data. Pada Firestore Database, semua dokumen di dalam *database* dapat dibaca oleh semua pengguna, namun hanya pengguna yang terautentikasi yang diizinkan untuk menulis data. Demikian juga Firebase Storage, semua jalur di *bucket storage* dapat dibaca oleh semua pengguna, tetapi hanya pengguna yang terautentikasi yang diizinkan untuk menulis data. Aturan ini memastikan bahwa hanya pengguna terverifikasi yang dapat memodifikasi data, sementara data tetap dapat diakses secara publik untuk dibaca.

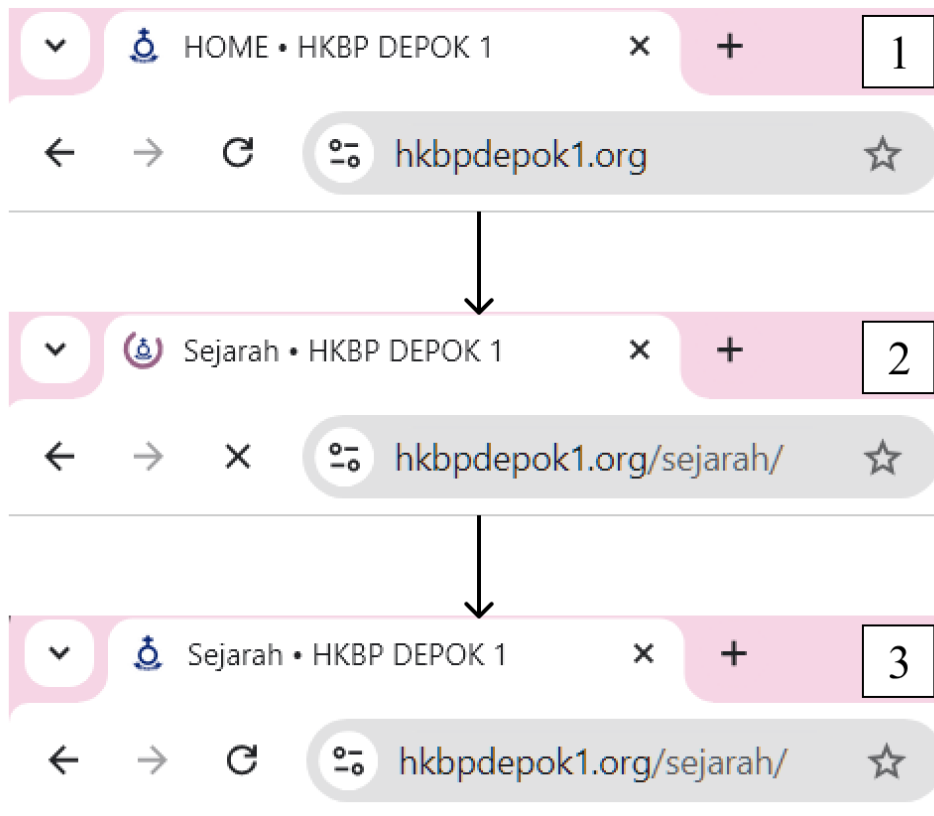
4.4 Tahap Pengujian

Proses pengujian dilakukan dua tahap yaitu pengujian kebutuhan fungsional dan non fungsional. Pada pengembangan proyek ini memiliki fokus pada pengujian performa aplikasi web yang dibuat oleh karena itu pengujian pertama yang dilakukan yaitu performa aplikasi yang ada di kebutuhan non fungsional.

Untuk menunjukkan perbaikan kinerja aplikasi SPA yang telah dibuat, aplikasi dibandingkan dengan sistem informasi HKBP lain yang dibangun dengan MPA. Pengujian menggunakan dua metrik yaitu total waktu dan ukuran data yang ditransfer (dalam *byte*) dan jumlah *resource* yang terpakai selama pengujian. Metrik pertama diukur menggunakan software Wireshark dan metrik kedua menggunakan Developer Tools pada google chrome. Tujuan pengujian ini adalah untuk membandingkan data transfer dan banyaknya *resource* yang terpakai jika menu pada halaman web diakses dua kali antara sistem HKBP yang menggunakan MPA dengan aplikasi SPA yang sudah dibangun. Dari pengujian ini dapat mengetahui sistem mana yang mempunyai data transfer lebih kecil serta sistem mana yang lebih cepat dan yang hemat dalam penggunaan *resource*.

Aplikasi web dengan metode MPA yang digunakan sebagai pembanding adalah *website* HKBP Depok. Untuk menentukan bahwa HKBP Depok menggunakan metode MPA, dilakukan analisis karakteristik MPA. Karakteristik utama yang dianalisis adalah MPA melakukan *reload* seluruh halaman setiap kali navigasi diklik. *Reload page* ini menjadi indikator utama bahwa *website* tersebut merupakan MPA. Gambar 4.4 menunjukkan bahwa *website* HKBP Depok melakukan *reload* halaman saat navigasi diklik.

Website HKBP Depok dipilih menjadi pembanding karena memiliki kesamaan dalam fungsi yang ada di aplikasi web SPA yang sudah dikembangkan. Kesamaan fungsi yang ada adalah dapat menampilkan informasi tentang Dewan Koinonia, Dewan Marturia, Dewan Diakonia, beserta dengan semua seksi yang ada di dalamnya, dan menampilkan berita.



Gambar 4. 8 Website HKBP Depok Melakukan *Reload Page*

Pengujian pertama dilakukan menggunakan Wireshark dengan skenario yang terdapat di Gambar 4.4. Langkah pertama adalah mencari alamat IP dari masing-masing *website*. *Website* HKBP Depok memiliki alamat “hkbpdepok1.org” dan *website* yang dikembangkan memiliki alamat “<https://sihman-c569d.web.app/>”. Untuk mendapatkan alamat IP dari masing-masing *website*, gunakan Command Prompt (CMD) dengan mengetikkan perintah “nslookup [alamat_web]”. Alamat IP dari web HKBP Depok adalah 202.157.186.128, dan alamat IP untuk web HKBP Manyar adalah 2620:0:890::100. Alamat IP tersebut kemudian digunakan sebagai filter pada Wireshark dengan perintah filter `ip.src==202.157.186.128 || ip.dst==202.157.186.128` untuk HKBP Depok dan `ipv6.src==2620:0:890::100 || ipv6.dst==2620:0:890::100` untuk HKBP Manyar. Filter ini bertujuan untuk menyaring data yang ditampilkan oleh Wireshark, sehingga hanya menampilkan data yang bersumber dari dan menuju alamat IP yang difilter. Langkah berikutnya adalah melakukan skenario penggunaan *website* pada masing-masing web. Untuk mendapatkan hasil dari skenario pengujian web, proses penghitungan transaksi dihentikan dengan mengeklik “stop capturing packets”. Selanjutnya, buka menu “Conversations” di *Toolbar* “Statistics” untuk melihat hasil pengujian. Hasil pengujian dengan Wireshark disimpan dalam *file* dengan format pcap (*.pcap).

Setelah pengujian menggunakan Wireshark, pengujian diulang dengan Developer Tools menggunakan skenario yang terdapat di Gambar 4.5. Langkah pertama adalah membuka salah satu alat pada Google Chrome yaitu Developer Tools. Setelah itu, buka bagian *Network* dan selanjutnya lakukan skenario pengujian *website*. Hasil pengujian dengan Developer Tools disimpan di *file* dengan format har (*.har). Hasil pengujian menggunakan Wireshark dituliskan di Tabel 4.14 dan Tabel 4.15, sedangkan untuk hasil pengujian menggunakan Developer Tools dapat dituliskan di Tabel 4.16 dan Tabel 4.17.

Tabel 4. 16 Hasil Pengujian dengan Wireshark pada HKBP Manyar

Data kinerja yang didapat	Nilai
Total <i>Package</i>	120
<i>Package Outgoing</i>	61
<i>Package Incoming</i>	59
Total Data	39 kB
<i>Data Outgoing</i>	23 kB
<i>Data Incoming</i>	15 kB
Kecepatan	
<i>Speed Outgoing</i>	2648 bits/s
<i>Speed Incoming</i>	1749 bits/s
Durasi	70.4 s

Tabel 4. 15 Hasil Pengujian dengan Wireshark pada HKBP Depok

Data kinerja yang didapat	Nilai
Total <i>Package</i>	2,812
<i>Package Outgoing</i>	851
<i>Package Incoming</i>	1961
Total Data	3 MB
<i>Data Outgoing</i>	55 kB
<i>Data Incoming</i>	3 MB
Kecepatan	
<i>Speed Outgoing</i>	2801 bits/s
<i>Speed Incoming</i>	137 kB/s
Durasi	158 s

Tabel 4. 14 Hasil Pengujian dengan Developer Tools pada HKBP Manyar

Data kinerja yang didapat	Nilai
Fase Pertama	
<i>Request</i>	87
<i>Transferred</i>	259 kB
<i>Resources</i>	7.4 MB
Fase Kedua	
<i>Request</i>	87
<i>Transferred</i>	259 kB
<i>Resources</i>	7.4 MB

Tabel 4. 17 Hasil Pengujian dengan Developer Tools pada HKBP Depok

Data kinerja yang didapat	Nilai
Fase Pertama	
<i>Request</i>	1064
<i>Transferred</i>	5.6 MB
<i>Resources</i>	60.4 MB
Fase Kedua	
<i>Request</i>	2299
<i>Transferred</i>	13.8 MB
<i>Resources</i>	119 MB

Hasil dokumentasi dari pengujian ini dicantumkan di Lampiran F.

Selanjutnya merupakan hasil pengujian dari fungsional aplikasi. Pengujian ini dilakukan bersama dengan pemangku kepentingan untuk memastikan semua fungsi berjalan dengan baik. Pengujian menggunakan Black Box Testing. Untuk hasil pengujian fungsi dapat dilihat di Tabel 4.18.

Tabel 4. 18 Hasil Pengujian Fungsional Aplikasi

ID	Fungsi	Hasil Pengujian
F01	Melihat informasi yang ada	Berhasil
F02	Mengunduh Warta Jemaat	Berhasil
F03	Mengelola informasi dewan dan pengurusnya	Berhasil
F04	Mengelola informasi internal	Berhasil
F05	Mengelola laporan keuangan	Berhasil

Terdapat 26 pengujian sistem yang memiliki hasil berhasil dan memberikan kesimpulan bahwa sistem sudah berjalan dengan baik. Hasil pengujian tersebut diteruskan dengan melakukan pengujian *traceability matrix*. Hasil *traceability matrix* di Tabel 4. 19 merupakan hasil pengujian sistem juga dibuat untuk memetakan hubungan antara pengujian sistem dengan *use case*. Tabel tersebut menjelaskan bahwa data *use case* sudah terhubung dengan data deskripsi pengujian sistem dan memiliki hasil pengujian berhasil.

4.5 Tahap Peluncuran

Pada tahap ini, aplikasi sudah berhasil di-*deploy* menggunakan fitur *hosting* pada Firebase yang bernama Firebase Hosting. Dengan menggunakan Firebase Hosting, aplikasi dapat diakses secara online melalui URL yang telah ditentukan. URL yang diberikan oleh fitur Firebase ini adalah “https://sihman-c569d.web.app/”. Firebase Hosting memungkinkan aplikasi untuk diakses dengan cepat dan aman, serta menyediakan fitur-fitur yang mendukung performa dan keandalan aplikasi. Setelah proses *deployment* selesai, aplikasi siap digunakan oleh pengguna sesuai dengan fungsionalitas yang telah dirancang dan diuji sebelumnya.

Tabel 4. 19 Traceability Matrix Sistem Testing

ID Kebutuhan	Deskripsi Kebutuhan	ID Testing	Deskripsi User Acceptance Testing	Hasil Pengujian
F01	Melihat informasi tentang HKBP Manyar	UAT01	Jemaat dapat melihat informasi HKBP	Berhasil
		UAT02	Jemaat dapat melihat informasi dewan Koinonia	Berhasil
		UAT03	Jemaat dapat melihat informasi seksi yang berada di dewan Koinonia	Berhasil
		UAT04	Jemaat dapat melihat informasi dewan Marturia	Berhasil
		UAT05	Jemaat dapat melihat informasi seksi yang berada di dewan Marturia	Berhasil
		UAT06	Jemaat dapat melihat informasi dewan Diakonia	Berhasil
		UAT07	Jemaat dapat melihat informasi seksi yang berada di dewan Diakonia	Berhasil
		UAT08	Jemaat dapat melihat informasi keuangan gereja	Berhasil
		UAT10	Jemaat dapat melihat tata ibadah	Berhasil
		UAT11	Jemaat dapat melihat jadwal kegiatan	Berhasil
		UAT12	Jemaat dapat melihat berita dan artikel	Berhasil
		UAT13	Jemaat dapat mencari atau memfilter berita	Berhasil
		F02	Mengunduh warta jemaat	UAT09
F03	Mengelola informasi dewan dan pengurusnya	UAT14	Pendeta dapat mengubah informasi dewan dan pengurusnya	Berhasil
F04	Mengelola informasi internal	UAT15	Sekretaris dapat mengelola tata ibadah	Berhasil
		UAT16	Sekretaris dapat mengelola warta jemaat	Berhasil
		UAT17	Sekretaris dapat mengelola jadwal kegiatan	Berhasil
		UAT18	Sekretaris dapat menambah berita	Berhasil
		UAT19	Sekretaris dapat mengedit berita	Berhasil
		UAT20	Sekretaris dapat melihat preview berita	Berhasil
F05	Mengelola laporan keuangan	UAT21	Bendahara dapat mengelola laporan keuangan	Berhasil
NF01	Waktu muat halaman yang cepat dan hemat <i>resource</i>	UAT23	Menganalisis dengan Wireshark	Berhasil
NF02	Fitur admin hanya bisa digunakan untuk pengguna yang sudah login	UAT24	Pendeta bisa login sebagai role pendeta	Berhasil
		UAT25	Sekretaris bisa login sebagai role Sekretaris	Berhasil
		UAT26	Bendahara bisa login sebagai role bendahara	Berhasil

4.6 Pembahasan

Hasil pengujian yang didapat dari pengujian menggunakan Wireshark dan Developer Tools akan dilakukan analisis. Proses analisis pertama adalah analisis pengujian Wireshark menggunakan data dari pada Tabel 4.14 dan Tabel 4.15. Analisis ini melakukan dua perbandingan kinerja sistem informasi web HKBP Manyar yang menggunakan SPA dengan sistem informasi HKBP Depok yang menggunakan MPA. Perbandingan kinerja kedua sistem dapat dilihat di Tabel 4.20.

Tabel 4. 20 Perbandingan MPA dan SPA dari Pengujian Wireshark

Data kinerja untuk menyelesaikan seluruh tahapan	MPA	SPA	Perbandingan Kinerja (MPA : SPA)
Total <i>Package</i>	2812	120	23 : 1
<i>Incoming Package</i>	1961	59	33 : 1
<i>Outgoing Package</i>	851	61	16 : 1
Total Data transfer (<i>kilobyte</i>)	3.055	38	80 : 1
<i>Data Incoming</i>	3.000	15	200 : 1
<i>Data Outgoing</i>	55	23	2 : 1
Waktu Transfer (detik)	157.94	70.38	2 : 1
Rata-rata Kecepatan Transfer (kbps)	1,470	2.19	

Jumlah paket *incoming* dan *outgoing* menunjukkan jumlah permintaan dan respons yang dikirimkan antara klien dan server. Pada MPA, jumlah paket jauh lebih banyak karena setiap kali halaman diakses atau dinavigasi, permintaan baru dikirim dan respons penuh diterima, yang menyebabkan peningkatan jumlah paket. Sebaliknya, SPA mengurangi jumlah paket dengan memuat data secara dinamis sesuai kebutuhan, yang mengurangi frekuensi permintaan ke server. Jumlah paket ini akan mempengaruhi jumlah data transfer, semakin banyaknya package yang ada maka akan memperbanyak jumlah data yang dilakukan.

Data *incoming* mengacu pada data yang diterima oleh klien dari server. Dalam kasus ini, aplikasi MPA menerima jauh lebih banyak data dibandingkan dengan SPA. MPA mengirimkan seluruh halaman web setiap kali pengguna navigasi, termasuk semua *resource* yang diperlukan (HTML, CSS, JavaScript, dan konten). Sebaliknya, SPA biasanya memuat sebagian besar *resource* saat pertama kali diakses dan hanya mengambil data tambahan yang dibutuhkan selama interaksi pengguna, sehingga mengurangi volume data *incoming* secara signifikan.

Data *outgoing* mengacu pada data yang dikirim dari klien ke server, seperti permintaan HTTP, pengiriman formulir, dan interaksi lainnya. Dalam kasus ini, MPA menghasilkan lebih banyak data *outgoing* dibandingkan dengan SPA. Hal ini bisa disebabkan oleh pengiriman permintaan HTTP yang lebih sering saat setiap halaman dimuat ulang di MPA, sementara SPA biasanya mengoptimalkan permintaan data melalui AJAX atau metode lain yang lebih efisien.

Total data transfer adalah jumlah dari data *incoming* dan *outgoing*. Hasil ini menunjukkan bahwa MPA memerlukan data transfer yang jauh lebih besar dibandingkan dengan SPA. SPA lebih efisien dalam hal jumlah data yang ditransfer karena hanya mengambil data yang diperlukan setelah halaman pertama dimuat, sedangkan MPA memerlukan pengambilan data penuh setiap kali halaman berubah.

Waktu transfer adalah total waktu yang diperlukan untuk mentransfer semua data selama pengujian. MPA membutuhkan waktu yang lebih lama untuk mentransfer data dibandingkan dengan SPA. Ini bisa disebabkan oleh ukuran data yang lebih besar dan frekuensi permintaan

yang lebih tinggi di MPA, sementara SPA lebih cepat karena optimisasi pada data transfer setelah muatan awal.

Kecepatan transfer rata-rata dihitung berdasarkan total data yang ditransfer dibagi dengan waktu transfer. Hasilnya menunjukkan bahwa meskipun MPA memiliki kecepatan transfer rata-rata yang lebih tinggi, ini tidak berarti lebih efisien. Kecepatan yang lebih tinggi dalam kasus MPA disebabkan oleh *volume* data yang besar, sementara SPA, meskipun memiliki kecepatan transfer yang lebih rendah, lebih efisien karena *volume* data yang jauh lebih kecil dan waktu transfer yang lebih singkat.

Data yang ditransfer ke server dari *browser* (atau data *outgoing*) pada web HKBP Depok lebih besar dua kali lipat lebih banyak dibanding data *outgoing* pada aplikasi SPA. Namun perbedaannya tidak seekstrem pada data *incoming*, yang mencapai 200 kali lipat. Hal ini karena data *outgoing* yang berasal dari *browser* tidak banyak berbeda antara web HKBP Depok dan aplikasi SPA. Selebihnya perbedaan muncul karena jumlah *byte overhead* yang dikirim oleh *browser* karena jumlah permintaan yang berbeda terkait jumlah langkah aktivitas yang berbeda.

Pembahasan selanjutnya yaitu pengujian menggunakan alat Developer Tools. Pada hasil pengujian terdapat data yang berasal dari fase pertama dan fase kedua. Fase pertama merupakan data yang didapat ketika melakukan skenario pengujian web pertama kali, sedangkan fase kedua merupakan data yang didapat ketika melakukan pengulangan kembali pengujian web tanpa menghapus hasil data pertama. Hasil data dari pengujian HKBP Depok dan HKBP Manyar dapat dilihat di Tabel 4.21 dan Tabel 4.22. Dari hasil tersebut dibuat perbandingan antara request, data transfer, dan *resource* yang digunakan pada setiap fase.

Tabel 4. 22 Perbandingan Pengujian MPA dengan Developer Tools

Data kinerja HKBP Depok	Fase pertama	Fase kedua	Perbandingan (Fase 1 : Fase 2)
<i>Request</i>	1064	2299	1 : 2
<i>Transferred</i>	5.6 MB	13.8 MB	1 : 2
<i>Resources</i>	60.4 MB	119 MB	1 : 2

Tabel 4. 21 Perbandingan Pengujian SPA dengan Developer Tools

Data kinerja HKBP Manyar	Fase pertama	Fase kedua	Perbandingan (Fase 1 : Fase 2)
<i>Request</i>	87	87	1 : 1
<i>Transferred</i>	259 kB	259 kB	1 : 1
<i>Resources</i>	7.4 MB	7.4 MB	1 : 1

Dari hasil pengujian menggunakan Developer Tools, ketika halaman yang sudah diakses akan diakses kembali, pada MPA akan memuat ulang datanya kembali. Sedangkan pada SPA hanya butuh satu kali saja. Sehingga bisa hemat *resource*. Dari jumlah *resource* yang digunakan terjadi ketimpangan yang sangat besar. Ini bisa dijelaskan dari penggunaan komponen pada SPA. Dalam SPA komponen yang dipakai terus menerus seperti *header* dan *footer* hanya perlu dimuat sekali. Dalam pengujian ini dilakukan dengan membuka banyak tampilan halaman, sehingga membuat MPA akan terus me-*reload* ulang *header* dan *footer* tersebut sehingga menyebabkan *resource* yang digunakan menjadi sangat besar. Alasan lain adalah dari isi konten masing-masing web. Web HKBP Depok memiliki konten yang sangat besar, sebagai contoh gambar yang terdapat di web ini menggunakan format *file* .png dan jpg. Sedangkan pada web

HKBP Manyar menggunakan format .webp yang dapat mengurangi *resource* saat mengunduh gambar, tetapi tidak menurunkan kualitas gambar tersebut.

Dalam web yang dikembangkan memiliki tampilan yang sedikit berbeda antara pengunjung biasa dengan pengunjung yang sudah *login*. Pada pengunjung yang sudah login terdapat *icon* profil yang sedang login dan dapat mengedit informasi yang ada sesuai *role*. Pengunjung yang sudah *login* akan mendapat id admin dengan tiga *role* berbeda, yaitu pendeta, sekretaris, dan bendahara. Pada fitur yang bisa diakses sekretaris, hanya sekretaris saja yang bisa melakukan fitur tersebut, begitu juga dengan *role* lain. Contoh perbandingan tampilan admin yang *login* sebagai pendeta dan *login* sebagai sekretaris pada tampilan berita dapat dilihat di Lampiran H. Di tampilan *role* sekretaris memiliki akses untuk menambah berita, mengedit berita, dan menghapus berita, sedangkan *role* pendeta tidak memiliki akses tersebut.

Pengembangan aplikasi ini sudah berada ditahap selesai. Semua fungsi dapat berjalan dengan baik. Namun, isi dari konten yang ada pada aplikasi web ini belum diisi dengan informasi yang terbaru karena masih menggunakan data *dummy* atau sementara karena kurangnya informasi dari pihak sekretaris gereja. Penggunaan aplikasi ini sebagai sistem informasi HKBP Manyar akan dilanjutkan ke tahap diskusi bersama *stakeholder*.

(Halaman ini sengaja dikosongkan)

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pembuatan sistem Informasi HKBP Manyar sebagai wadah penyebaran informasi yang ada. Pembuatan sistem informasi sudah sampai di tahap peluncuran dan sudah bisa dipakai oleh pengguna. Admin sudah bisa melakukan penyebaran informasi melalui aplikasi Web ini dan pengguna lain juga sudah bisa melihat isi dari konten yang disediakan. Pembuatan aplikasi web ini masih menggunakan desain seadanya dari komponen yang tersedia pada framework yang digunakan sehingga masih diperlukan pengembangan lebih dari segi desain.

Pada akhir dari pembahasan tentang pengujian aplikasi web dengan konsep *Single Page Application* (SPA) dan *Multi Page Application* (MPA), menunjukkan bahwa SPA lebih efisien dibandingkan MPA dalam hal penggunaan data dan *resource*. SPA mengirimkan dan menerima lebih sedikit data, memerlukan lebih sedikit *request*, dan menggunakan lebih sedikit *resource* secara keseluruhan. Efisiensi ini terutama disebabkan oleh sifat SPA yang hanya memuat halaman sekali dan kemudian memperbarui konten secara dinamis, berbeda dengan MPA yang memuat ulang seluruh halaman setiap kali pengguna melakukan navigasi. Penggunaan SPA dapat menghemat *bandwidth* dan waktu, memberikan pengalaman pengguna yang lebih baik dengan responsivitas yang lebih tinggi dan pengurangan waktu *loading*. Hal ini sejalan dengan penelitian yang menyatakan bahwa SPA dapat meningkatkan efisiensi sistem dan kenyamanan pengguna.

5.2 Saran

Dari implementasi yang sudah dilaksanakan terdapat saran mengenai Tugas Akhir ini. Harapannya dengan adanya saran yang diberikan, Tugas Akhir ini menjadi lebih baik dari sebelumnya. Saran yang dapat diberikan dari implementasi web HKBP Manyar harus meningkatkan performa supaya lebih baik lagi. Saran lain, supaya sistem informasi HKBP Manyar memiliki fitur yang lebih bagus lagi seperti pendataan anggota gereja.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- Alshamrani, A., & Bahattab, A. (2015). *A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model*. www.IJCSI.org
- Annur M, & Delianti V. (2020). Rancang Bangun Sistem Informasi Manajemen Organisasi Himpunan Mahasiswa Rokan Hulu. *Jurnal Vocational Teknik Elektronika Dan Informatika*, 8(1). <http://ejournal.unp.ac.id/index.php/voteknika/index>
- Chandra, K. B., Firstian, B., Fikri, M., & Rosyani, P. (2020). KELUARGA MAHASISWA TEGAL (KMT) CIPUTAT BERBASIS WEB. *Jurnal Kreativitas Mahasiswa Informatika*, 1, 126–132.
- Dokumentasi Firebase. (2024a). *Firestore: Pengantar Firestore Hosting*. https://firebase.google.com/docs/hosting?authuser=0&hl=id#implementation_path
- Dokumentasi Firebase. (2024b). *Firestore: Pengantar Firestore Realtime Database*. <https://firebase.google.com/docs/database?hl=id>
- Dokumentasi Vue.js. (2024). *Perkenalan: Apa itu Vue.js?* <https://docs.vuejs.id/v2/guide/#>
- Fajarianditya Nugroho, M., Agung Prakoso, S., & Tridian Galih, S. (2022). *FRONTEND ADMIN APLIKASI GEREJAKU BERBASIS WEB MENGGUNAKAN FRAMEWORK VUE JS PADA GEREJA ANUGRAH*.
- Fink, G., & Flatow, I. (2014). Pro Single Page Application. *Apress*, 1(1), 3–13.
- Haviluddin. (2011). Memahami Penggunaan UML (Unified Modelling Language). *Jurnal Informatika Mulawarman*.
- Hutahaean, J. (2014). *KONSEP SISTEM INFORMASI*.
- Johan, M. C., Handoyo, E. D., & Santoso, S. (2023). Prototipe Aplikasi Bimbingan Tugas Akhir Mahasiswa Berbasis Web. *Jurnal Teknik Informatika Dan Sistem Informasi*, 9(1). <https://doi.org/10.28932/jutisi.v9i1.5726>
- Madhurih. (2023). *Software Requirement Specification (SRS) Format*. <https://www.geeksforgeeks.org/software-requirement-specification-srs-format/>
- Pramudita, R., Arifin, R. W., Nurul Alfian, A., & Safitri, N. (2021). PENGGUNAAN APLIKASI FIGMA DALAM MEMBANGUN UI/UX YANG INTERAKTIF PADA PROGRAM STUDI TEKNIK INFORMATIKA STMIK TASIKMALAYA. *Shilka Dina Anwariya*, 3(1)
- Putra Mahendri, R., Dwinggo Samala, A., Negeri Padang, U., Jl Hamka Kampus UNP, I., & Tawar Padang, A. (2023). Penerapan Teknologi Single Page Application (SPA) Pada Aplikasi Lelang Barang Secondhand Berbasis Website. *Jurnal Vocational Teknik Elektronika Dan Informatika*, 11(3). <http://ejournal.unp.ac.id/index.php/voteknika/>

- Richard Sharpe, Ed Warnicke, & Ulf Lamping. (2024). *Wireshark User's Guide*. https://www.wireshark.org/docs/wsug_html/
- Ristyabudi, A., & Thamrin, H. (2016). *Penerapan Single Page Application Pada Proses Pengisian Online Data Rencana Studi Mahasiswa*. <https://journals.ums.ac.id/index.php/khif/article/view/1727>
- Santoso, M. F. (2021). Teknik Single Page Application (SPA) Layout Web dengan Menggunakan React JS dan Bootstrap. *JURNAL KHATULISTIWA INFORMATIKA* , 9(2), 107–114.
- Scott, E. A. (2015). *SPA : Design and Architecture*.
- Sommerville, I. (2011). *Software engineering*. Pearson.
- Srinam. (2024). *Use Case Diagrams - Unified Modeling Language (UML)*. <https://www.geeksforgeeks.org/use-case-diagram/>
- Suryaningsih, S., Riandika, Y., Hasanah, A., & Anggraito, S. (2020). Aplikasi Wakaf Indonesia Berbasis Blockchain. *Edumatic: Jurnal Pendidikan Informatika*, 4(2), 20–29. <https://doi.org/10.29408/edumatic.v4i2.2402>
- Susanto A. (2017). *Sistem Informasi Manajemen*.
- Wibowo, A. T., & Wiguna, A. S. (2019). Pemanfaatan Teknologi Single Page Application (SPA) dalam Pembuatan Aplikasi Feedback Dosen dari Mahasiswa Sebagai Bentuk Pengawasan Lembaga Terhadap Kinerja Dosen di Bidang Pengajaran. *SMARTICS Journal*, 5(1), 34–43. <https://doi.org/10.21067/smartics.v5i1.3327>

LAMPIRAN-LAMPIRAN ATAU APPENDIKS

A. Surat Permohonan



INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
DEPARTEMEN INFORMATIKA

Lampiran : Sebelas (11) lembar
Perihal : Permohonan Pengambilan Data Penelitian

Yth.

**Sekretaris Gereja HKBP Manyar
Surabaya**

Dengan hormat,

Saya mahasiswa dari Institut Teknologi Sepuluh Nopember ingin mengajukan permohonan kepada Bapak/Ibu dengan biodata sebagai berikut:

Nama : Amsal Herbert

NRP : 5025201182

Program Studi : Teknik Informatika / Semester 8

Untuk melaksanakan pengambilan data penelitian Tugas Akhir di Gereja HKBP Manyar. Tugas Akhir ini berupa proyek pengembangan Sistem Informasi HKBP Manyar dalam bentuk aplikasi website. Data yang akan diambil berupa jawaban dari wawancara yang akan dilakukan serta masukan dari diskusi bersama pengurus dalam pengembangan Sistem Informasi HKBP Manyar.

Adapun proyek pengembangan website tersebut dilakukan selama tiga bulan dimulai dari 27 April 2024 sampai 27 Juni 2024.

Demikian permohonan ini saya sampaikan, atas perhatian dan kerja sama Bapak/Ibu saya ucapkan terima kasih.

Hormat saya,

Amsal Herbert

B. Spesifikasi Kasus, Diagram Aktivitas, dan Diagram *Sequence*

a. *Use Case*

Use Case 01: Melihat Informasi yang Ada

Komponen	Deskripsi
Nama Use Case	Melihat Informasi yang ada
Nomor	UC-01
Deskripsi	Aktor dapat melihat informasi yang ada pada web
Aktor	Pengunjung
Kondisi Awal	Aktor berada di beranda HKBP Manyar
Kondisi Akhir	Sistem menampilkan informasi yang dibutuhkan aktor
Alur Normal	<ol style="list-style-type: none"> 1. Aktor masuk halaman beranda 2. Aktor memilih navigasi yang ada tersedia 3. Sistem menampilkan pilihan informasi yang tersedia 4. Aktor memilih informasi yang tersedia 5. Sistem menampilkan halaman informasi yang dipilih aktor
Alur Alternatif	-

Use Case 02: Mengunduh Warta Jemaat

Komponen	Deskripsi
Nama Use Case	Mengunduh warta jemaat
Nomor	UC-02
Deskripsi	Aktor dapat mengunduh warta jemaat
Aktor	Pengunjung
Kondisi Awal	Aktor berada di halaman HKBP Manyar
Kondisi Akhir	Sistem mengirimkan file warta jemaat ke anggota
Alur Normal	<ol style="list-style-type: none"> 1. Aktor menuju komponen warta jemaat 2. Sistem menampilkan link download 3. Aktor memilih tombol download 4. Sistem mengirimkan file warta jemaat ke anggota
Alur Alternatif	

Use Case 03: Mengelola Informasi Dewan dan Pengurusnya

Komponen	Deskripsi
Nama Use Case	Mengelola Informasi Dewan dan Pengurusnya
Nomor	UC-03
Deskripsi	Aktor dapat mengelola informasi yang ada di dewan Koinonia, Marturia, dan Diakonia
Aktor	Pendeta
Kondisi Awal	Aktor memilih Kelola daftar pengurus
Kondisi Akhir	Sistem berhasil menyimpan hasil pengelolaan informasi
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih dewan yang ingin dikelola 2. Sistem menampilkan list informasi umum dan seksi yang ada 3. Aktor memilih informasi umum dewan 4. Sistem menampilkan form edit informasi 5. Aktor mengolah informasi umum dewan 1. Sistem menyimpan hasil pengelolaan informasi
Alur Alternatif	<ol style="list-style-type: none"> 3.1 Aktor memilih mengelola informasi seksi yang dipilih <ol style="list-style-type: none"> 3.1.1 Sistem menampilkan informasi umum seksi 3.1.2 Aktor mengolah informasi umum seksi 3.1.3 Sistem menyimpan hasil pengelolaan informasi

Use Case 04: Mengelola Informasi Internal

Komponen	Deskripsi
Nama Use Case	Mengelola Informasi Internal
Nomor	UC-04
Deskripsi	Aktor dapat mengelola informasi yang ada. Mengelola informasi bisa dalam bentuk menambah, menghapus, dan memperbarui
Aktor	Sekretaris
Kondisi Awal	Aktor memilih menu pada navigasi
Kondisi Akhir	Sistem menyimpan informasi yang diolah
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih navigasi 2. Sistem menampilkan halaman atau komponen 3. Aktor mengaktifkan edit mode 4. Sistem menampilkan menu tambah, hapus, dan edit 5. Aktor memilih informasi yang akan diolah 6. Sistem menyimpan informasi yang sudah diolah
Alur Alternatif	

Use Case 05: Mengelola Laporan Keuangan

Komponen	Deskripsi
Nama Use Case	Mengelola Laporan Keuangan
Nomor	UC-06
Deskripsi	Aktor dapat mengelola laporan keuangan dengan menambah dan menghapus dokumen laporan keuangan.
Aktor	Bendahara
Kondisi Awal	Aktor memilih navigasi mengelola laporan keuangan
Kondisi Akhir	Sistem memperbarui laporan keuangan
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih navigasi laporan keuangan 2. Sistem menampilkan halaman mengelola laporan keuangan 3. Aktor memilih menambahkan laporan keuangan 4. Sistem memperbarui laporan keuangan
Alur Alternatif	<ol style="list-style-type: none"> 3.1. Aktor memilih menghapus laporan keuangan <ol style="list-style-type: none"> 3.1.1. Sistem menghapus laporan keuangan yang dipilih 3.1.2. Sistem memperbarui laporan keuangan

b. Diagram Aktivitas

Diagram Aktivitas 01: Melihat Informasi yang Ada

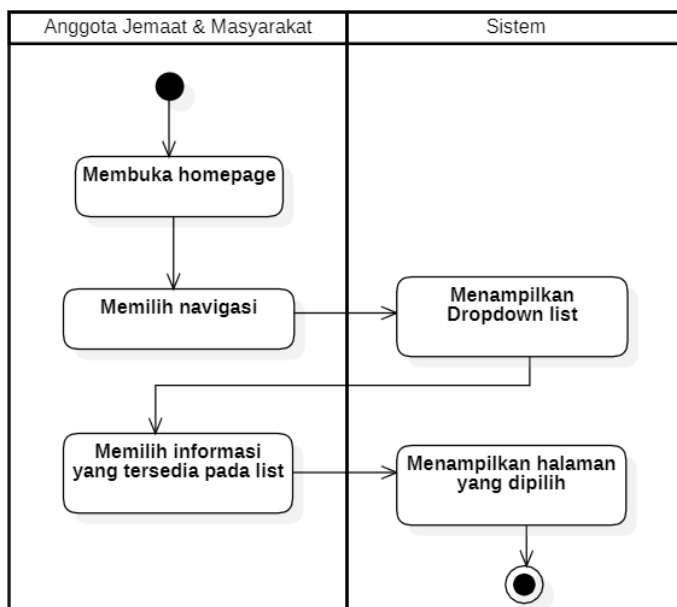


Diagram Aktivitas 02: Mengunduh Warta Jemaat

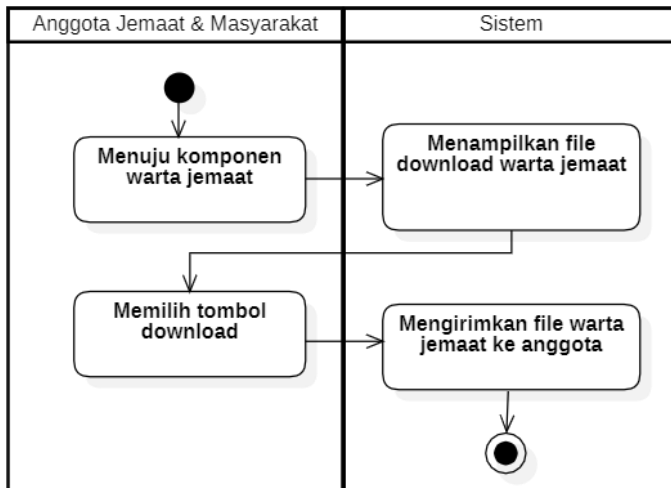


Diagram Aktivitas 03: Mengelola Informasi Dewan dan Pengurusnya

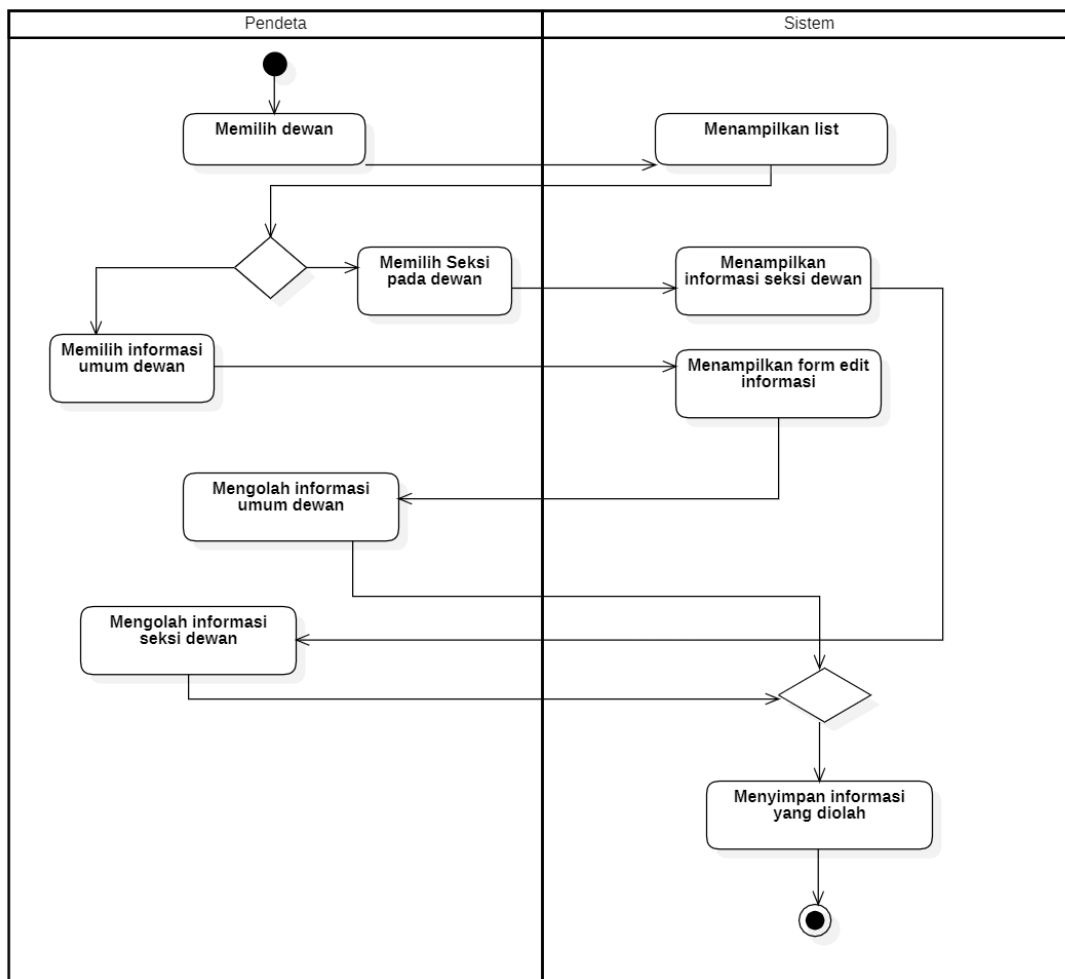


Diagram Aktivitas 04: Mengelola Informasi Internal

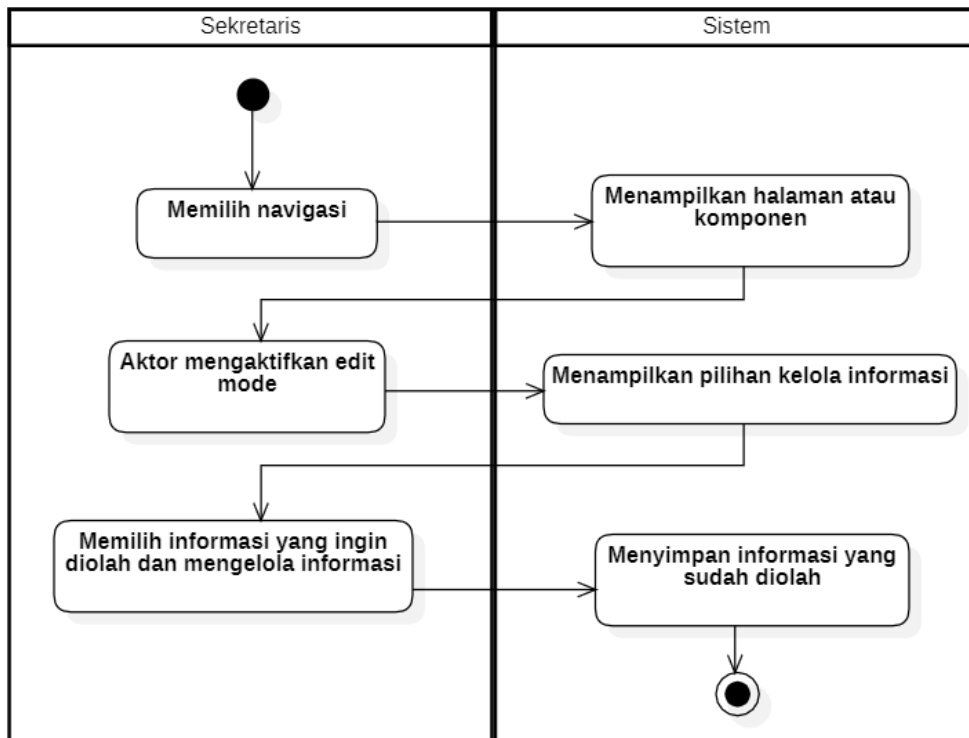
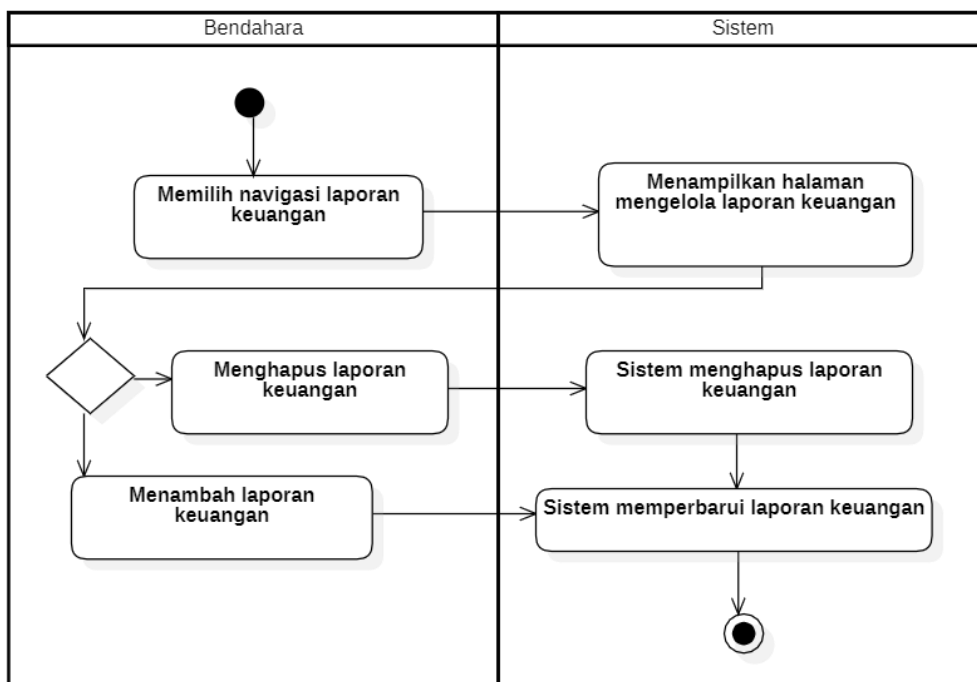


Diagram Aktivitas 05: Mengelola Laporan Keuangan



c. Diagram Sequence

Diagram Sequence 01: Melihat Informasi yang Ada

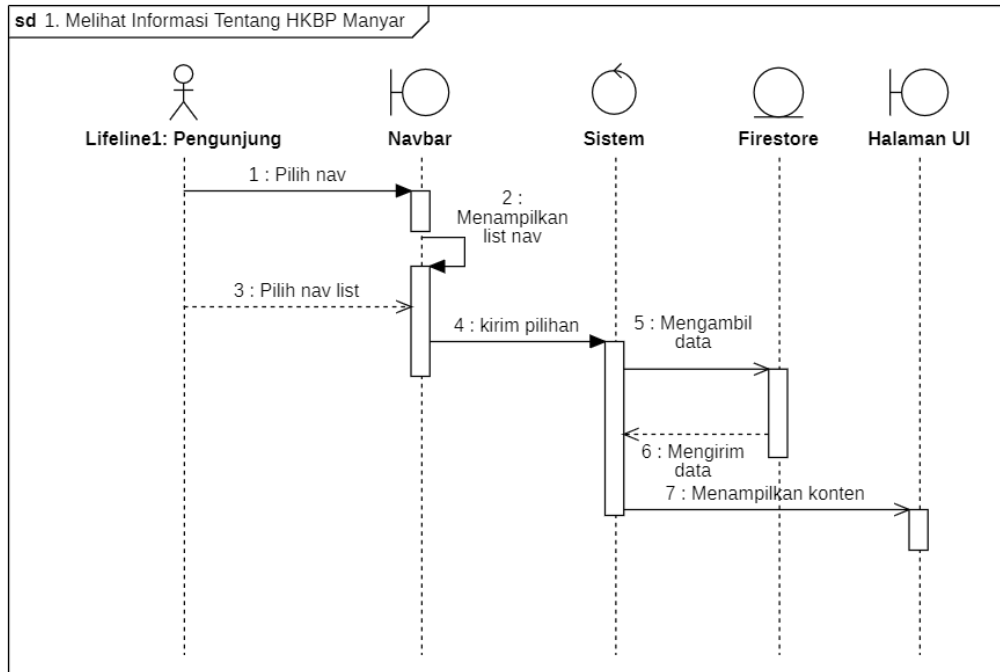


Diagram Sequence 02: Mengunduh Warta Jemaat

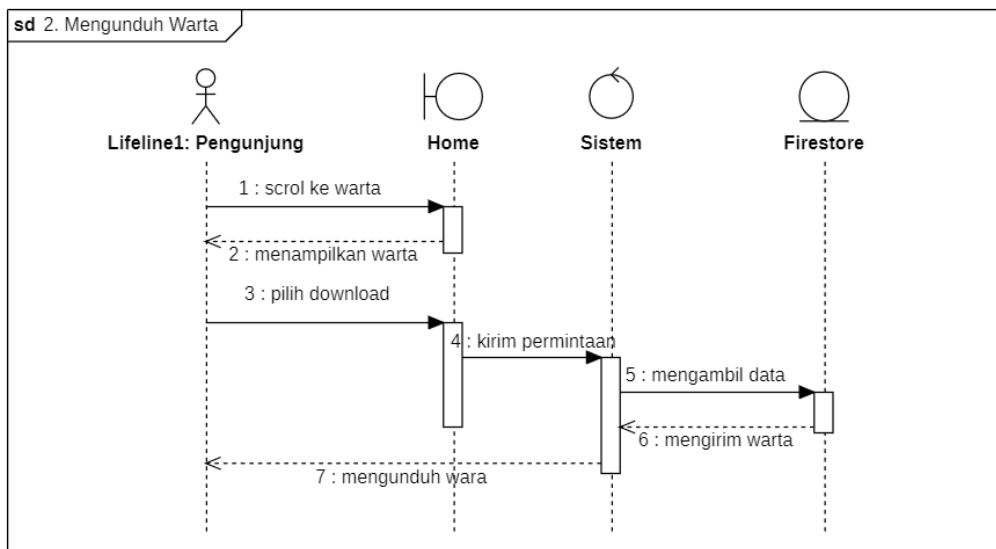


Diagram *Sequence* 03: Mengelola Informasi Dewan dan Pengurusnya

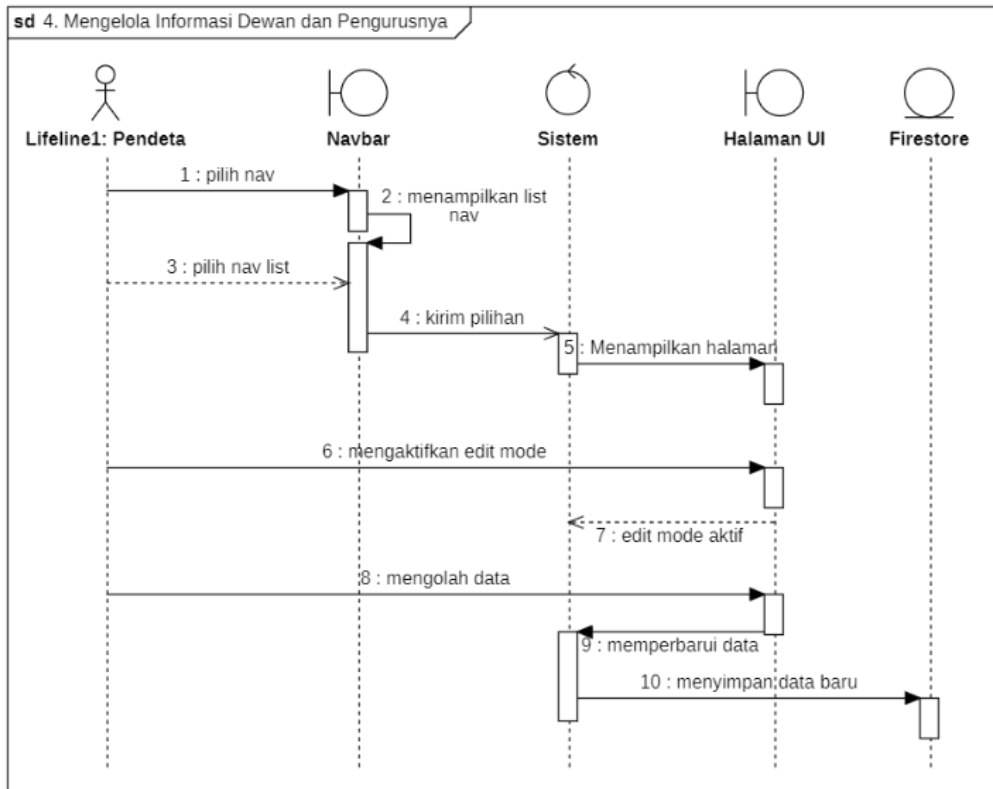


Diagram *Sequence* 04: Mengelola Informasi Internal

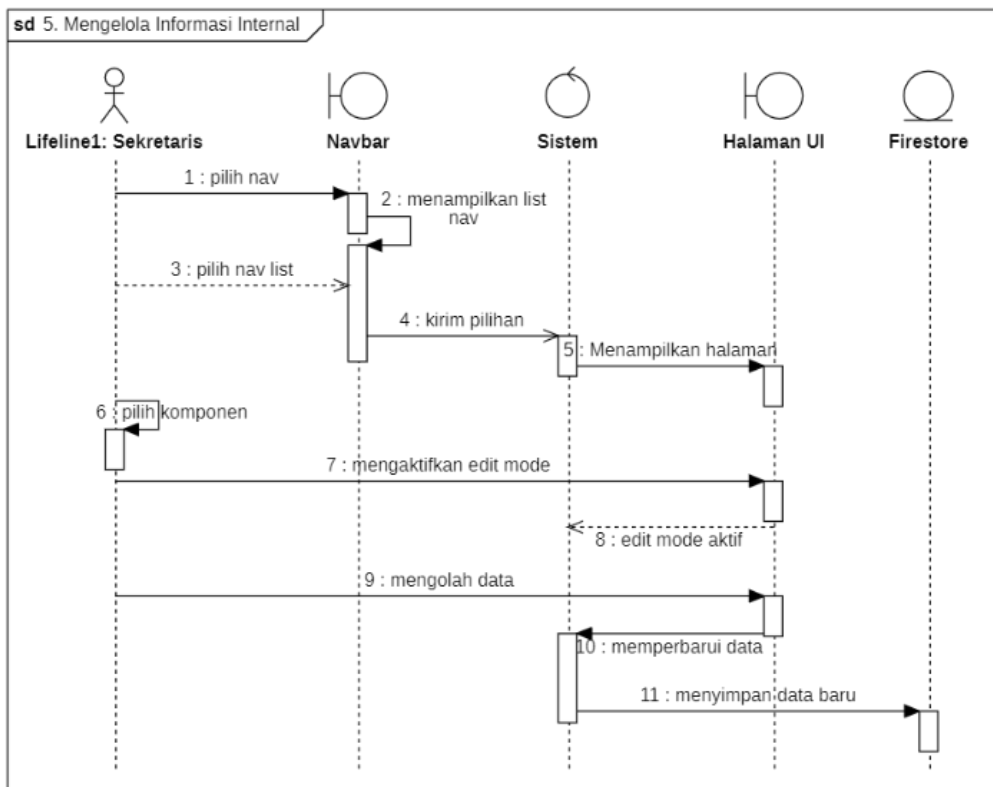
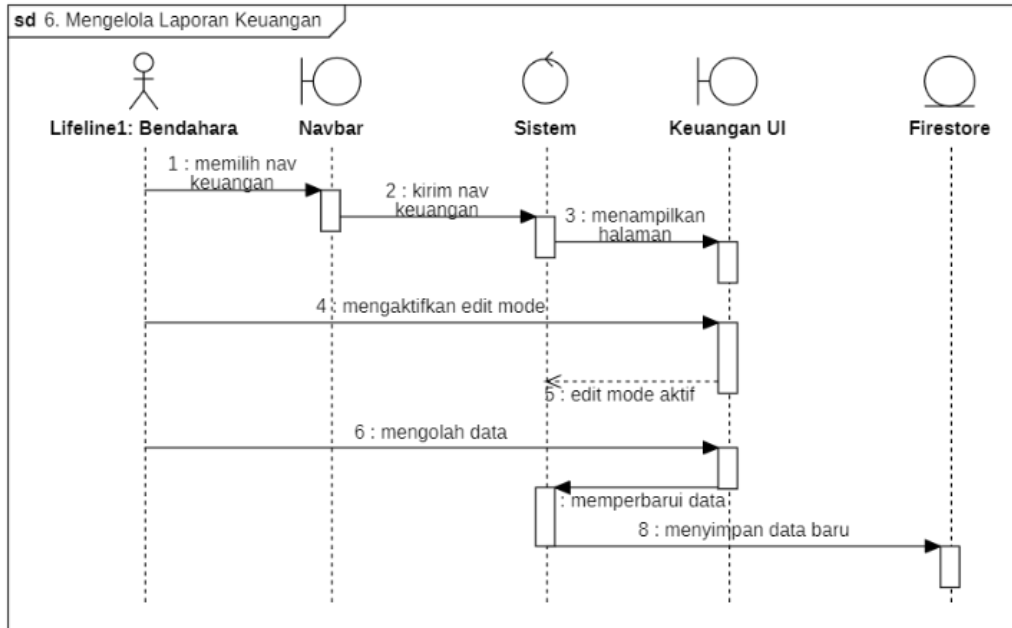


Diagram *Sequence* 05: Mengelola Laporan Keuangan



C. Rancangan User Acceptance Test

ID	Deskripsi Pengujian	Deskripsi Pengujian	Keterangan
UAT01	Jemaat dapat melihat informasi HKBP	<ul style="list-style-type: none"> Masuk pada halaman utama Klik beranda Lihat informasi yang dibutuhkan 	Jemaat dapat melihat informasi seputar HKBP Manyar
UAT02	Jemaat dapat melihat informasi dewan Koinonia	<ul style="list-style-type: none"> Masuk pada halaman utama Klik navigasi Koinonia Pada dropdown klik dewan Koinonia 	Jemaat dapat melihat informasi dewan Koinonia meliputi tanggung jawab, program kerja, dan stuktur kepengurusan
UAT03	Jemaat dapat melihat informasi seksi yang berada di dewan Koinonia	<ul style="list-style-type: none"> Masuk pada halaman utama Klik navigasi Koinonia Pada dropdown klik seksi ada 	Jemaat dapat melihat informasi seksi yang berada di dewan Koinonia meliputi pengertian, program, dan pengurus.
UAT04	Jemaat dapat melihat informasi dewan Marturia	<ul style="list-style-type: none"> Masuk pada halaman utama Klik navigasi Marturia Pada dropdown klik dewan Marturia 	Jemaat dapat melihat informasi dewan Marturia meliputi tanggung jawab, program kerja, dan stuktur kepengurusan
UAT05	Jemaat dapat melihat informasi seksi yang berada di dewan Marturia	<ul style="list-style-type: none"> Masuk pada halaman utama Klik navigasi Marturia Pada dropdown klik seksi ada 	Jemaat dapat melihat informasi seksi yang berada di dewan Marturia meliputi pengertian, program, dan pengurus.

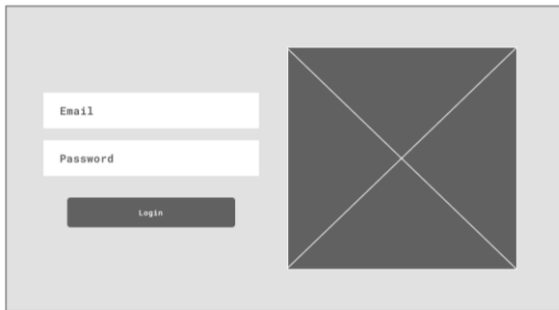
UAT06	Jemaat dapat melihat informasi dewan Diakonia	<ul style="list-style-type: none"> • Masuk pada halaman utama • Klik navigasi Diakonia • Pada dropdown klik dewan Diakonia 	Jemaat dapat melihat informasi dewan Diakonia meliputi tanggung jawab, program kerja, dan struktur kepengurusan
UAT07	Jemaat dapat melihat informasi seksi yang berada di dewan Diakonia	<ul style="list-style-type: none"> • Masuk pada halaman utama • Klik navigasi Diakonia • Pada dropdown klik seksi ada 	Jemaat dapat melihat informasi seksi yang berada di dewan Diakonia meliputi pengertian, program, dan pengurus.
UAT08	Jemaat dapat melihat informasi keuangan gereja	<ul style="list-style-type: none"> • Masuk pada halaman utama • Klik navigasi Tentang • Pada dropdown klik keuangan 	Jemaat dapat melihat informasi keuangan gereja
UAT09	Jemaat dapat mengunduh warta jemaat	<ul style="list-style-type: none"> • Masuk pada halaman utama • Gulir halaman utama kebawah • Pada bagian warta klik download warta minggu ini 	Jemaat dapat mengunduh warta jemaat
UAT10	Jemaat dapat melihat tata ibadah	<ul style="list-style-type: none"> • Masuk pada halaman utama • Gulir halaman utama kebawah • Pada bagian tata ibadah terdapat pilihan tata ibadah sesuai sesi • Ubah sesi tata ibadah dengan pilihan dropdown 	Jemaat dapat melihat tata ibadah dan memilih tata ibadah pada sesi gereja yang tersedia
UAT11	Jemaat dapat melihat jadwal kegiatan	<ul style="list-style-type: none"> • Masuk pada halaman utama • Gulir halaman utama kebawah • Pada bagian agenda berbentuk kalender lihat agenda • Klik hari yang ingin dilihat 	Jemaat dapat melihat jadwal kegiatan
UAT12	Jemaat dapat melihat berita dan artikel	<ul style="list-style-type: none"> • Masuk pada halaman utama • Klik navigasi berita • Pada card berita yang ada, klik salah satu untuk membaca berita 	Jemaat dapat melihat berita dan artikel
UAT13	Jemaat dapat mencari atau memfilter berita	<ul style="list-style-type: none"> • Masuk pada halaman utama • Klik navigasi berita • Pada menu search, klik judul berita • Pada filter, pilih tag yang ada untuk memfilter konten berita 	Jemaat dapat mencari atau memfilter berita
UAT14	Pendeta dapat mengubah informasi dewan dan pengurusnya	<ul style="list-style-type: none"> • Masuk pada halaman utama • Klik navigasi salah satu dewan • Klik salah satu dropdown yang muncul • Nyalakan toggle edit halaman • Edit informasi yang ingin diubah • Klik tombol simpan 	Pendeta dapat mengubah informasi dewan dan pengurusnya
UAT15	Sekretaris dapat	<ul style="list-style-type: none"> • Masuk pada halaman utama 	Sekretaris dapat mengelola

	mengelola tata ibadah	<ul style="list-style-type: none"> • Gulir halaman utama kebawah • Pada bagian tata ibadah klik toggle edit info • Pilih sesi tata ibadah pada dropdown • Update tata ibadah terbaru 	tata ibadah
UAT16	Sekretaris dapat mengelola warta jemaat	<ul style="list-style-type: none"> • Masuk pada halaman utama • Gulir halaman utama kebawah • Pada bagian warta jemaat klik toggle edit info • Update warta jemaat terbaru 	Sekretaris dapat mengelola warta jemaat
UAT17	Sekretaris dapat mengelola jadwal kegiatan	<ul style="list-style-type: none"> • Masuk pada halaman utama • Gulir halaman utama kebawah • Pada bagian jadwal kegiatan klik toggle edit info • Klik tanggal yang ingin diedit • Pada menu komponen yang terbuka, edit info kegiatan 	Sekretaris dapat mengelola jadwal kegiatan
UAT18	Sekretaris dapat menambah berita	<ul style="list-style-type: none"> • Masuk pada halaman utama • Klik navigasi berita • Pada bagian jadwal kegiatan klik toggle edit info • Klik icon tambah • Masukan informasi pada form • Klik publish 	Sekretaris dapat menambah berita
UAT19	Sekretaris dapat mengedit berita	<ul style="list-style-type: none"> • Masuk pada halaman utama • Klik navigasi berita • Pada bagian jadwal kegiatan klik toggle edit info • Klik icon pencil atau trash pada card berita • Jika klik icon trash akan muncul konfirmasi hapus, jika edit lakukan pengisian edit form • Klik update 	Sekretaris dapat mengedit berita
UAT20	Sekretaris dapat melihat preview berita	<ul style="list-style-type: none"> • Masuk ke form tambah berita atau edit berita • Klik preview 	Sekretaris dapat melihat preview berita
UAT21	Bendahara dapat mengelola laporan keuangan	<ul style="list-style-type: none"> • Masuk pada halaman utama • Klik navigasi tentang • Pada dropdown klik keuangan • Pada bagian laporan keuangan klik toggle edit info • Update laporan keuangan 	Bendahara dapat mengelola laporan keuangan
UAT22	Melakukan uji kecepatan dengan wireshark	<ul style="list-style-type: none"> • Masuk ke web yang diuji • Cari alamat IP nya • Lakukan pengujian dengan memfilter data transfer sesuai IP 	Melakukan uji kecepatan dengan wireshark

UAT23	Melakukan uji penggunaan resource dengan Developer Tools	<ul style="list-style-type: none"> • Masuk ke web • Buka menu network pada Developer Tools • Lakukan pengujian 	Melakukan uji penggunaan resource dengan Developer Tools
UAT24	Pendeta bisa login sebagai role pendeta	<ul style="list-style-type: none"> • Pada halaman utama tambah `/login` untuk masuk ke login • Masukkan email dan password • Klik masuk 	Pendeta bisa login sebagai role pendeta
UAT25	Sekretaris bisa login sebagai role Sekretaris	<ul style="list-style-type: none"> • Pada halaman utama tambah `/login` untuk masuk ke login • Masukkan email dan password • Klik masuk 	Sekretaris bisa login sebagai role Sekretaris
UAT26	Bendahara bisa login sebagai role bendahara	<ul style="list-style-type: none"> • Pada halaman utama tambah `/login` untuk masuk ke login • Masukkan email dan password • Klik masuk 	bendahara bisa login sebagai role bendahara

D. Rancangan Antarmuka

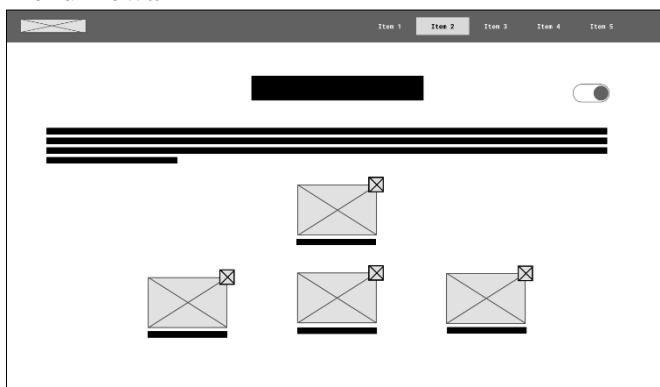
a. Menu Home



Spesifikasi Elemen-elemen Layar

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Email	<i>textfield</i>	Memasukan email admin terdaftar	<i>String</i>
2	password	<i>textfield</i>	Memasukan password admin terdaftar	<i>String</i>
3	login	<i>button</i>	Fungsi untuk masuk ke aplikasi	<i>ButtonClick</i>

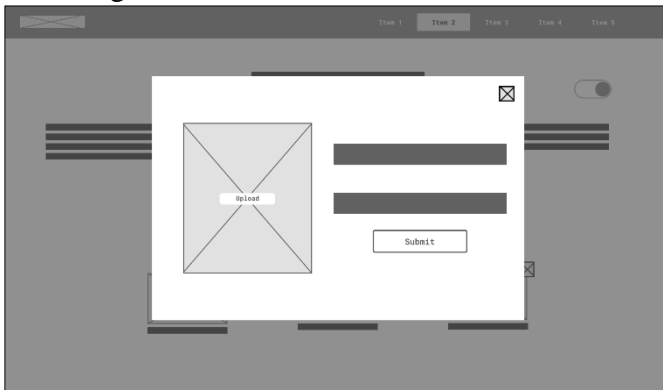
b. Menu Dewan



Spesifikasi Elemen-elemen Layar

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Teks Nama Dewan	<i>Teks</i>	Menampilkan nama dewan	<i>String</i>
2	Teks Info	<i>Teks</i>	Menampilkan informasi dewan	<i>String</i>
3	Gambar Pengurus	<i>Image</i>	Menampilkan gambar pengurus	<i>Image</i>
4	Nama Pengurus	<i>Teks</i>	Menampilkan nama pengurus	<i>String</i>
5	Toggle Edit	<i>Toggle</i>	Ditampilkan khusus admin dengan role pendeta berfungsi untuk mengaktifkan menu edit	<i>Click</i>
6	Icon Edit	<i>Icon</i>	Membuka menu edit pengurus	<i>Click</i>

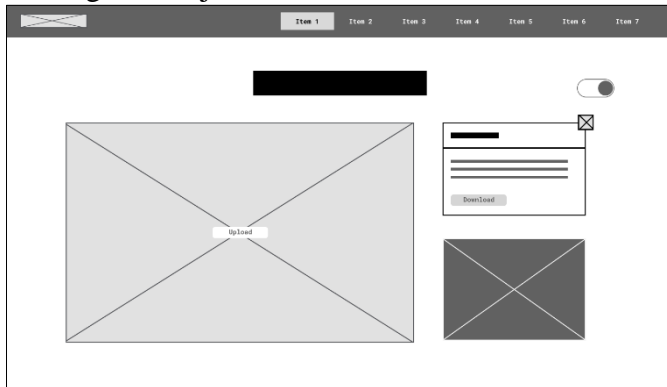
c. Edit Pengurus



Spesifikasi Elemen-elemen Layar

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	<i>Close</i>	<i>Icon</i>	Keluar dari <i>Dialog edit</i>	<i>Click</i>
2	<i>Upload image</i>	<i>Button</i>	Menambahkan foto	<i>Click</i>
3	<i>Form nama depan</i>	<i>Teksfield</i>	Memasukan nama depan	<i>String</i>
4	<i>Form nama belakang</i>	<i>Teksfield</i>	Memasukan nama belakang	<i>String</i>
5	<i>Submit</i>	<i>Button</i>	Mengirim <i>form</i>	<i>Click</i>

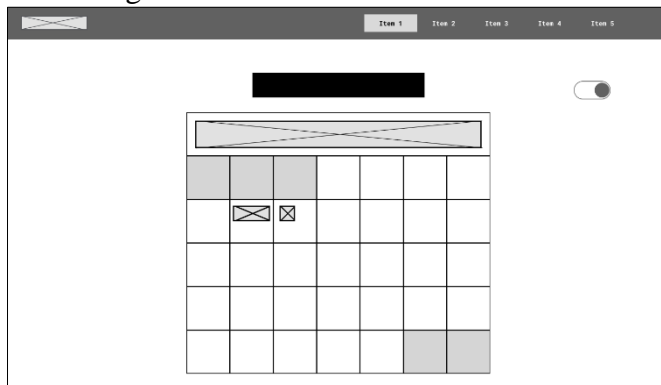
d. Keuangan Gereja



Spesifikasi Elemen-elemen Layar

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Judul	Teks	Menampilkan judul komponen	String
2	Toogle Edit	Toggle	Ditampilkan khusus admin dengan role bendahara berfungsi untuk mengaktifkan menu edit	Click
3	Gambar laporan keuangan	Image	Menampilkan gambar laporan keuangan mingguan	Image
4	Button upload	Button	Memperbarui laporan keuangan	Click
5	Hasil rekap	Teks	Menampilkan total pengeluaran dan pemasukan mingguan.	String
6	Tombol download	Button	Mengunduh laporan mingguan	Click
7	Edit Icon	Icon	Memperbarui hasil rekap	Click
8	Gambar QR code	Image	Menampilkan QR code untuk persembahan digital	Image

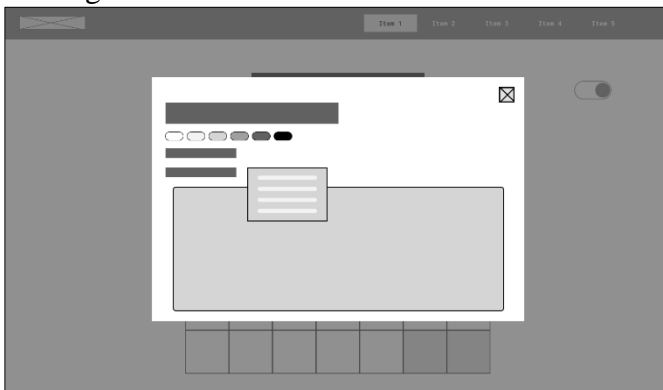
e. Jadwal Kegiatan



Spesifikasi Elemen-elemen Layar

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Judul	<i>Teks</i>	Menampilkan judul komponen	<i>String</i>
2	Toogle Edit	<i>Toggle</i>	Ditampilkan khusus admin dengan role sekretaris berfungsi untuk mengaktifkan menu edit	<i>Click</i>
3	Calender view	<i>Table</i>	Menampilkan tahun, bulan, dan tanggal	<i>Table</i>
4	Sel hari	<i>Teks</i>	Menampilkan hari	<i>Teks</i>
5	Tag kegiatan	<i>Tag</i>	Memberitahu tanggal yang memiliki agenda	<i>Tag</i>
6	View Button	<i>Button</i>	Membuka dialog detail agenda	<i>Click</i>
7	Edit Button	<i>Button</i>	Memperbarui agenda	<i>Click</i>

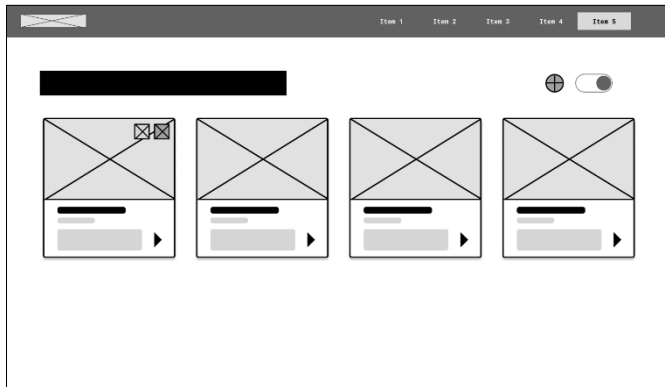
f. Edit Agenda



Spesifikasi Elemen-elemen Layar

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Judul	<i>Textfield</i>	Memasukan judul agenda	<i>String</i>
2	<i>Close</i>	<i>Icon</i>	Keluar dari dialog edit agenda	<i>Click</i>
3	Tag Warna	<i>Button</i>	Mengganti warna tag	<i>Click</i>
4	<i>Date</i>	<i>Date</i>	Memasukan tanggal	<i>String</i>
5	Pilih Seksi	<i>Button</i>	Membuka <i>dropdown</i> dewan atau seksi yang menyelenggarakan agenda	<i>Click</i>
6	<i>Dropdown</i>	<i>Button</i>	Menampilkan dewan atau seksi	<i>Click</i>
7	Deskripsi	<i>Textfield</i>	Memasukan deskripsi agenda	<i>String</i>

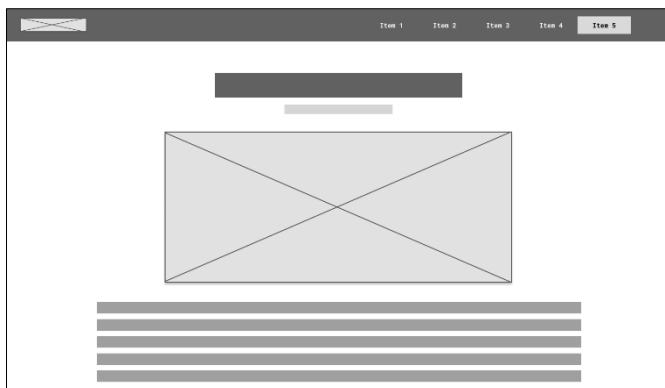
g. Berita



Spesifikasi Elemen-elemen Layar

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Judul	Teks	Menampilkan judul berita dan artikel	String
2	Toogle Edit	Toggle	Ditampilkan khusus admin dengan role sekretaris berfungsi untuk mengaktifkan menu edit	Click
3	Tambah berita	Button	Membuka halaman tambah berita	Click
4	Card Berita	Image and Text	Menampilkan cover berita, judul berita, tanggal berita, dan caption berita	Image and Text
5	Arrow Icon	Button	Membuka halaman berita	Click
6	Edit Icon	Icon	Membuka halaman edit berita	Click
7	Delete Icon	Icon	Menghapus berita	Click

h. Detail berita



Spesifikasi Elemen-elemen Layar

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Judul	<i>Teks</i>	Menampilkan judul berita	<i>String</i>
2	Tanggal	<i>Teks</i>	Menampilkan tanggal berita dibuat	<i>String</i>
3	<i>Cover</i>	<i>Image</i>	Menampilkan <i>cover</i> berita	<i>Image</i>
4	Konten	<i>Teks and image</i>	Menampilkan isi berita, dapat berupa teks dan gambar	<i>Teks and image</i>

i. Tambah atau edit berita



Spesifikasi Elemen-elemen Layar

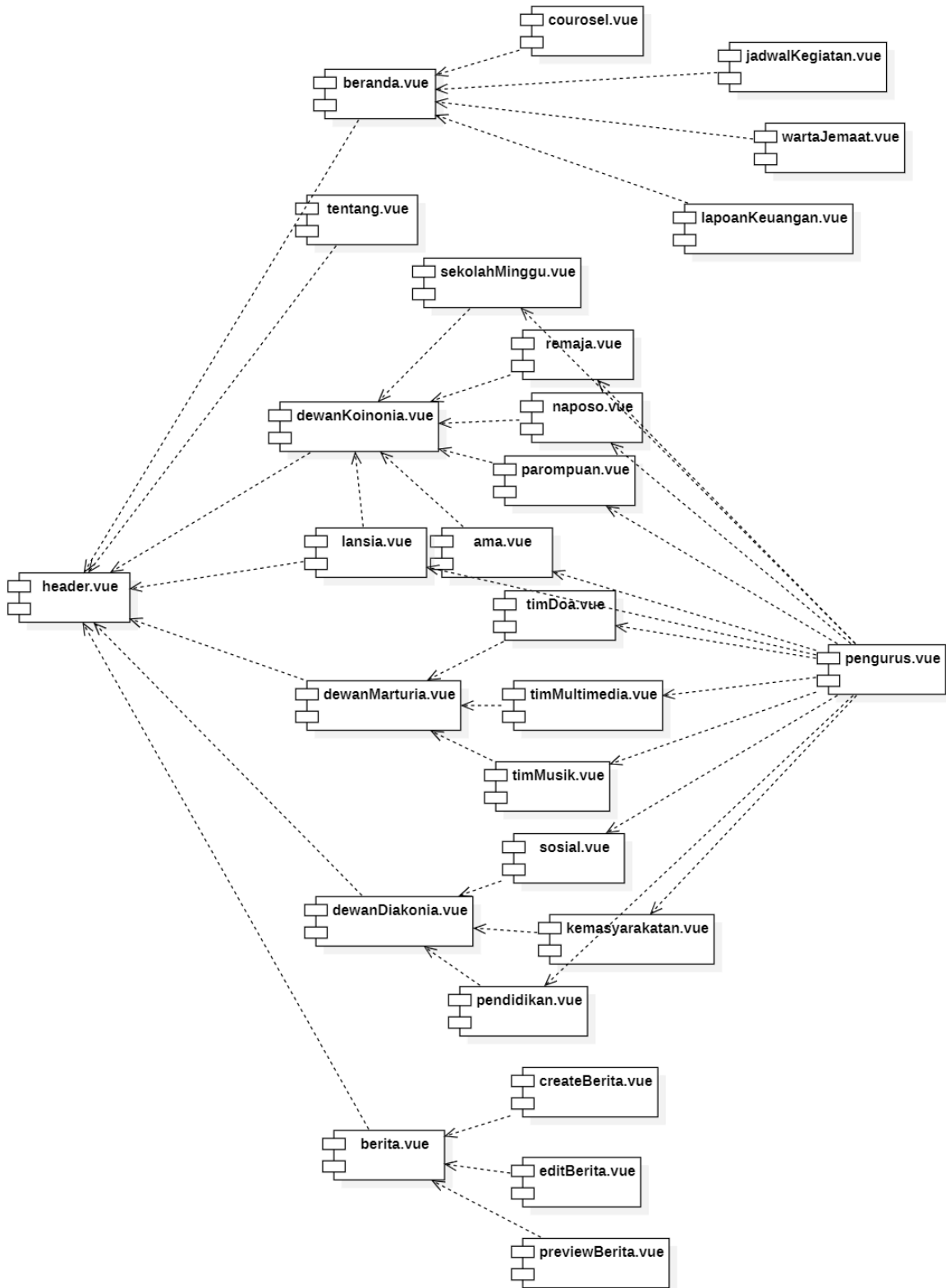
No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Jenis Masukan/ Keluaran
1	Judul	<i>Teks</i>	Menampilkan judul tambah berita atau edit berita	<i>String</i>
2	<i>Judul form</i>	<i>Teksfild</i>	Memasukan judul berita	<i>String</i>
3	<i>Button upload</i>	<i>Button</i>	Mengunggah foto <i>cover</i>	<i>Click</i>
4	<i>Toolbar</i>	<i>Icon</i>	Menjalankan <i>toolbar</i>	<i>Click</i>
5	Deskripsi	<i>Tectfield</i>	Memasukan deskripsi berita	<i>String</i>

E. Komponen Page

Komponen Page dan fungsi

ID	Komponen Pages	Fungsi
P01	Header.vue	Komponen yang mengatur navigasi
P02	Beranda.vue	Komponen yang mengatur tampilan beranda
P03	Courosel.vue	Komponen yang mengatur courosel
P04	jadwalKegiatan.vue	Komponen yang mengatur jadwal kegiatan
P05	wartaJemaat.vue	Komponen yang mengatur warta jemaat
P06	laporanKeuangan.vue	Komponen yang mengatur laporan keuangan
P07	Tentang.vue	Komponen yang mengatur tampilan tentang
P08	dewanKoinonia.vue	Komponen yang mengatur tampilan dewan koinonia general
P09	sekolahMinggu.vue	Komponen yang mengatur tampilan seksi sekolah minggu
P10	Remaja.vue	Komponen yang mengatur tampilan seksi rema
P11	Naposo.vue	Komponen yang mengatur tampilan seksi naposo
P12	Parampuan.vue	Komponen yang mengatur tampilan seksi parompuan
P13	Ama.vue	Komponen yang mengatur tampilan seksi ama
P14	Lansia.vue	Komponen yang mengatur tampilan seksi lansia
P15	dewanMarturia.vue	Komponen yang mengatur tampilan dewan marturia general
P16	timDoa.vue	Komponen yang mengatur tampilan seksi doa
P17	timMusik.vue	Komponen yang mengatur tampilan seksi musik
P18	timMultimedia.vue	Komponen yang mengatur tampilan seksi multimedia
P19	DewanDiakonia.vue	Komponen yang mengatur tampilan dewan diakonia general
P20	Sosial.vue	Komponen yang mengatur tampilan seksi sosial
P21	Kemasyarakatan.vue	Komponen yang mengatur tampilan seksi kemasyarakatan
P22	Pendidikan.vue	Komponen yang mengatur tampilan seksi pendidikan
P23	Pengurus.vue	Komponen yang mengatur tampilan pengurus
P24	Berita.vue	Komponen yang mengatur tampilan berita
P25	createBerita.vue	Komponen yang mengatur tampilan tambah berita
P26	editBerita.vue	Komponen yang mengatur tampilan edit berita

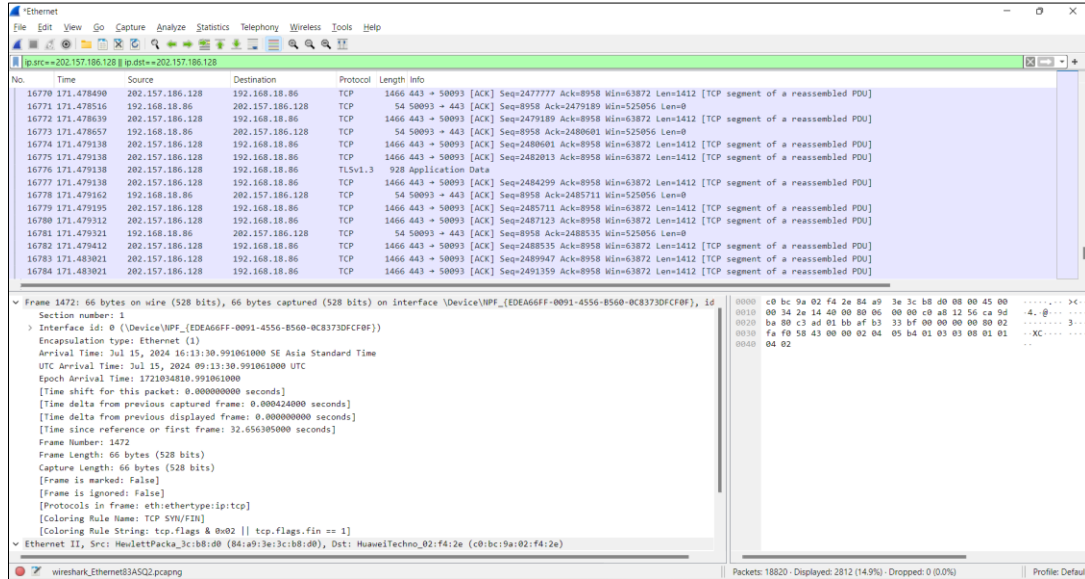
Pembuatan arah *Routes* Pada Komponen *Pages*



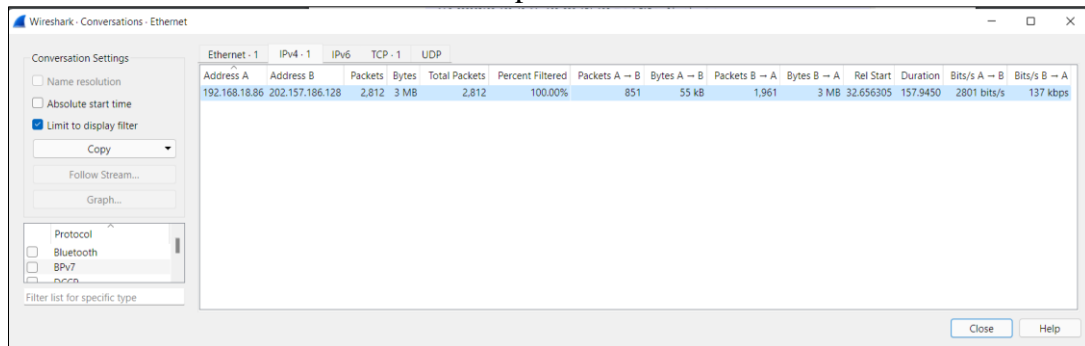
F. Dokumentasi Hasil Testing

a. Pengujian Wireshark

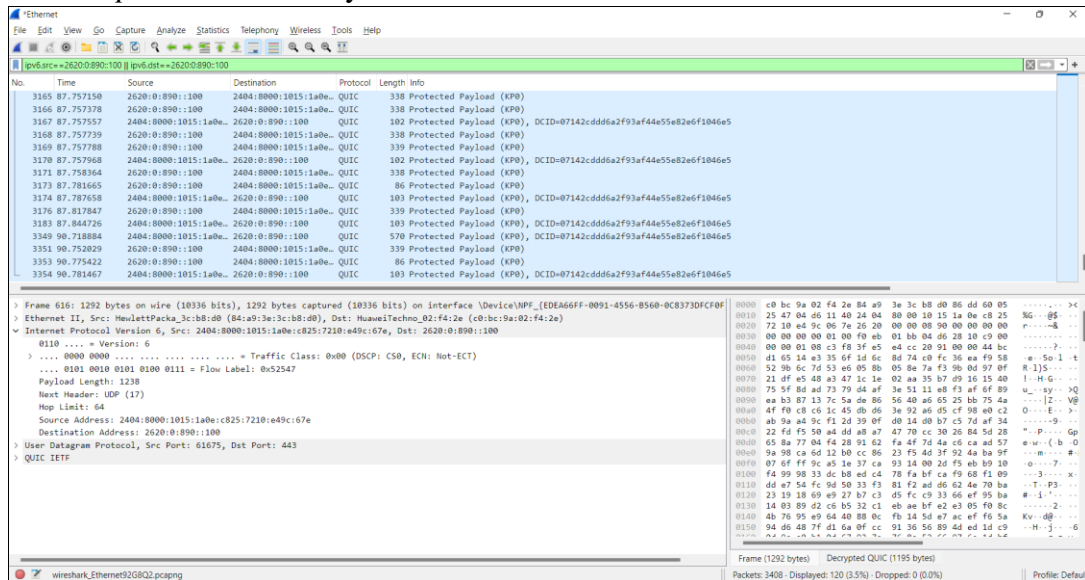
Hasil Capture HKBP Depok



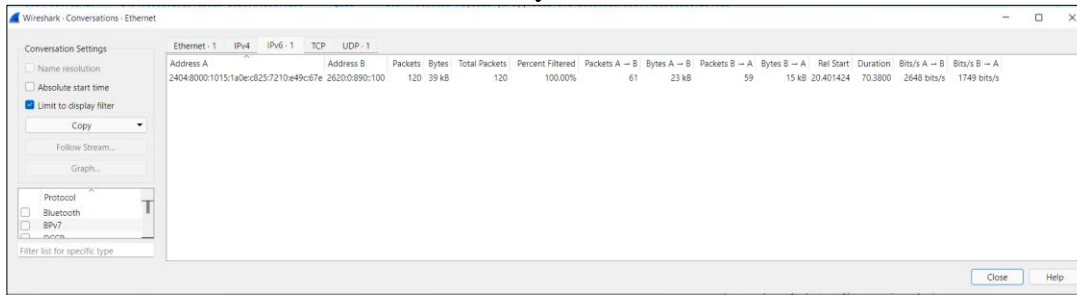
Hasil Conversations Statistic HKBP Depok



Hasil Capture HKBP Manyar

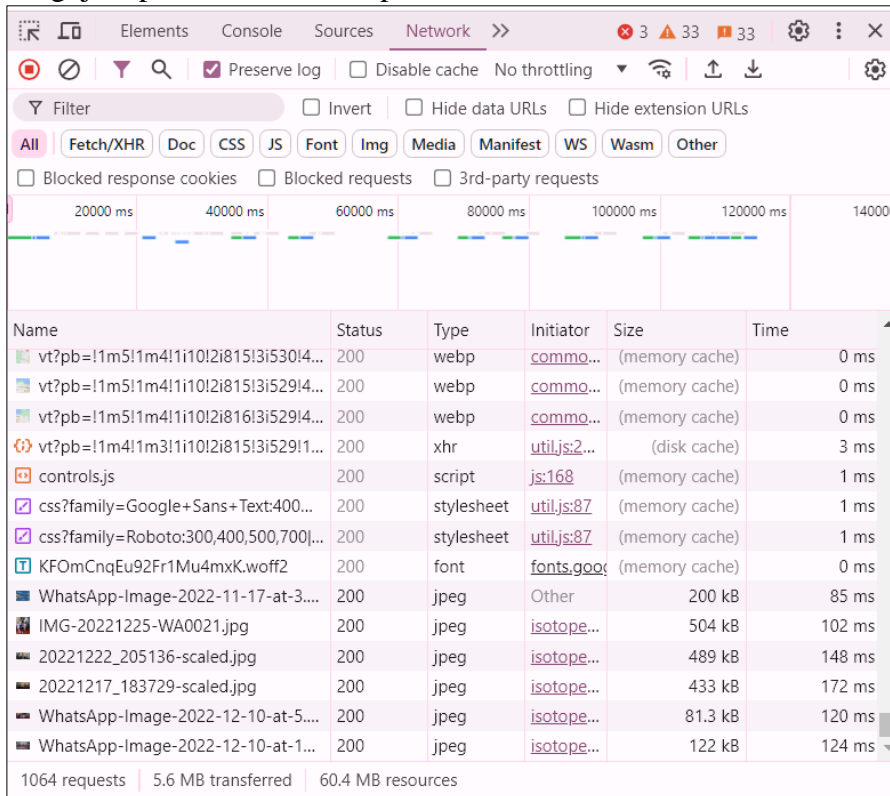


Hasil Conversations Statistic HKBP Manyar

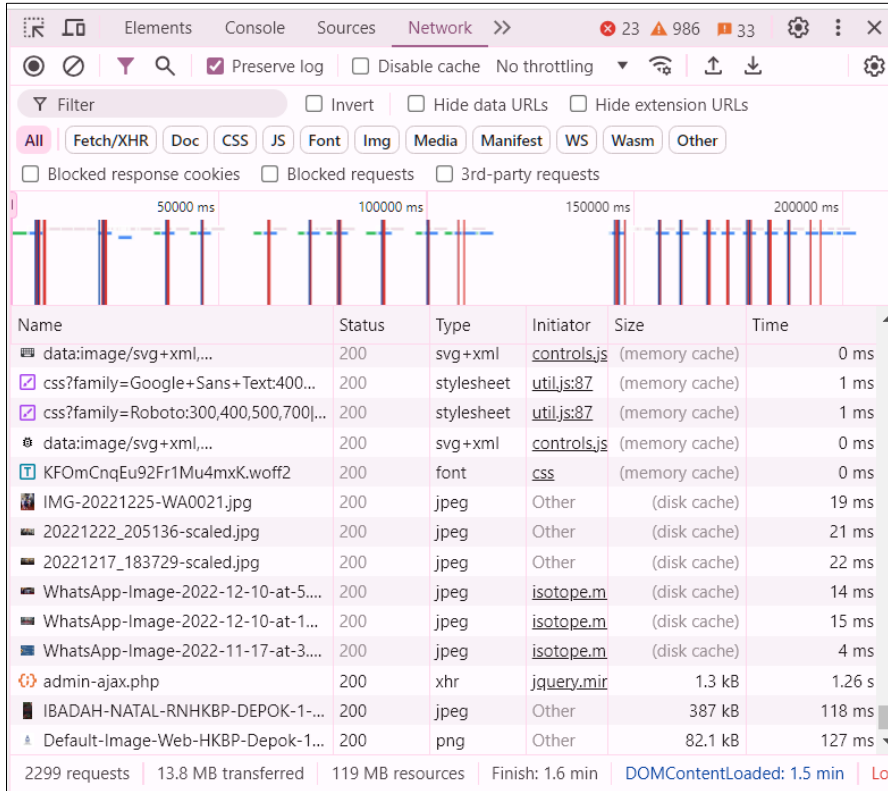


b. Pengujian Developer Tool

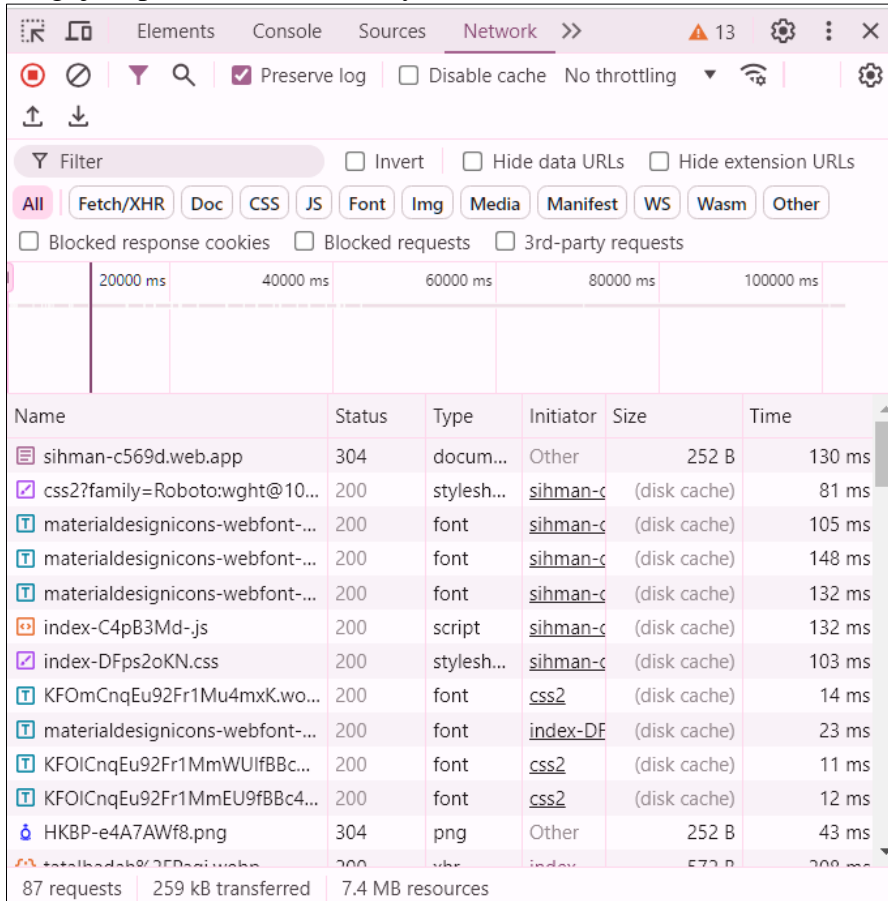
Pengujian pertama HKBP Depok



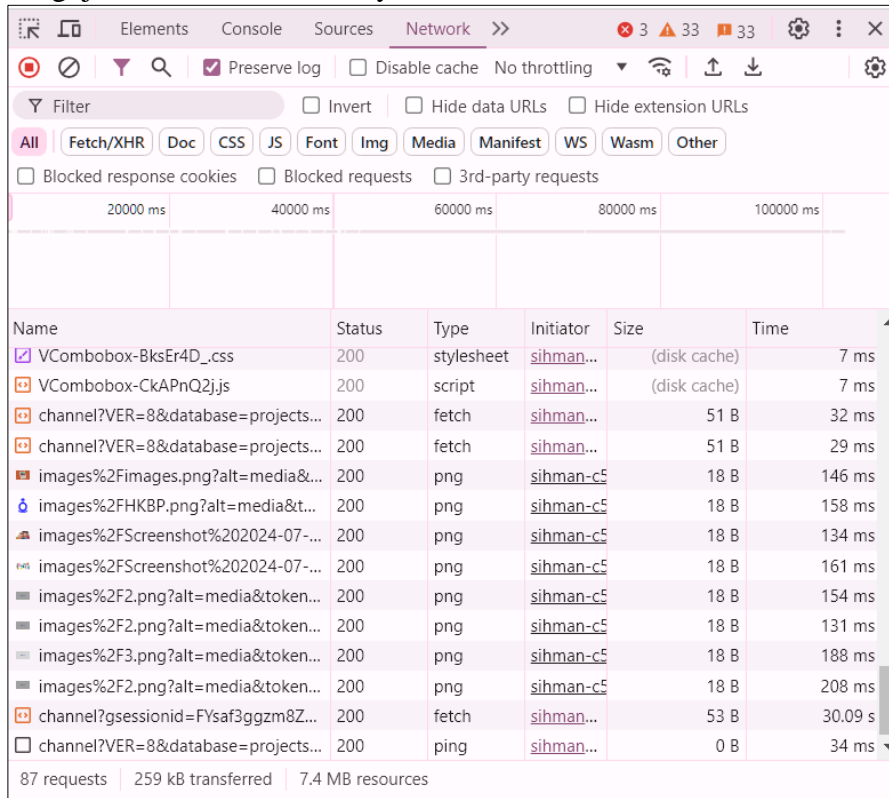
Pengujian kedua HKBP Depok



Pengujian pertama HKBP Manyar



Pengujian kedua HKBP Manyar



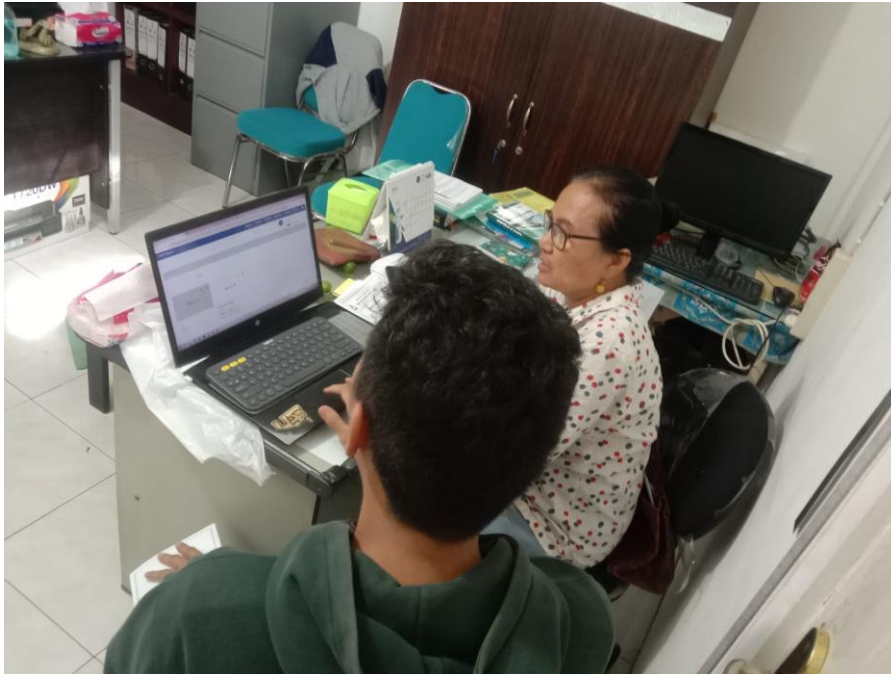
The screenshot shows the Network tab in Chrome DevTools. The top bar indicates 3 errors and 33 warnings. The filter is set to 'All'. The table below lists the requests:

Name	Status	Type	Initiator	Size	Time
VCombobox-8ksEr4D_.css	200	stylesheet	sihman...	(disk cache)	7 ms
VCombobox-CkAPnQ2jjs	200	script	sihman...	(disk cache)	7 ms
channel?VER=8&database=projects...	200	fetch	sihman...	51 B	32 ms
channel?VER=8&database=projects...	200	fetch	sihman...	51 B	29 ms
images%2Fimages.png?alt=media&...	200	png	sihman-c5	18 B	146 ms
images%2FHKBP.png?alt=media&t...	200	png	sihman-c5	18 B	158 ms
images%2FScreenshot%202024-07-...	200	png	sihman-c5	18 B	134 ms
images%2FScreenshot%202024-07-...	200	png	sihman-c5	18 B	161 ms
images%2F2.png?alt=media&token...	200	png	sihman-c5	18 B	154 ms
images%2F2.png?alt=media&token...	200	png	sihman-c5	18 B	131 ms
images%2F3.png?alt=media&token...	200	png	sihman-c5	18 B	188 ms
images%2F2.png?alt=media&token...	200	png	sihman-c5	18 B	208 ms
channel?gsessionid=FYsaf3ggzm8Z...	200	fetch	sihman...	53 B	30.09 s
channel?VER=8&database=projects...	200	ping	sihman...	0 B	34 ms

Summary: 87 requests | 259 kB transferred | 7.4 MB resources

G. Dokumentasi Pengujian Fungsional Bersama Sekretaris Gereja





H. Perbandingan Tampilan Berita pada Admin yang berbeda Login sebagai Pendeta

Login sebagai sekretaris

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Penulis dilahirkan di Bekasi, 10 Maret 2002, merupakan anak ketiga dari 4 bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Strada Bekasi, SDN 07 Duren Sawit Jakarta Timur, SMPN 109 Jakarta Timur dan SMAS Unggul Del. Setelah lulus dari SMA tahun 2020, Penulis mengikuti SBMPTN dan diterima di Departemen Teknik Informatika FTEIC - ITS pada tahun 2020 dan terdaftar dengan NRP 5025201182.

Di Departemen Teknik Informatika Penulis sempat aktif di beberapa kegiatan organisasi seperti HMTC ITS sebagai Staff Ahli Media dan Informasi 2023 dan bergabung dengan kepanitiaan untuk Schematics sebagai Perkamzin.