

**PENGGUNAAN METODE CLIENT SIDE RENDERING (CSR)
PADA APLIKASI MONITORING JARINGAN FTTH SECARA
REAL-TIME BERBASIS PETA DIGITAL**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan mendapatkan gelar Strata
Satu Program Studi Informatika**



**AHMAD MUSTAUFIQ
22EO10006**

**PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU KOMPUTER
UNIVERSITAS NAHDLATUL ULAMA AL GHAZALI
CILACAP
2026**

**PENGGUNAAN METODE CLIENT SIDE RENDERING (CSR)
PADA APLIKASI MONITORING JARINGAN FTTH SECARA
REAL-TIME BERBASIS PETA DIGITAL**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan mendapatkan gelar Strata
Satu Program Studi Informatika**



**AHMAD MUSTAUFIQ
22EO10006**

**PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU KOMPUTER
UNIVERSITAS NAHDLATUL ULAMA AL GHAZALI
CILACAP
2026**

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa penulisan skripsi dengan judul “Penggunaan Metode Client Side Rendering (CSR) Pada Aplikasi Monitoring Jaringan FTTH Secara Real-time Berbasis Peta Digital” adalah hasil karya saya dengan arahan dari pembimbing dan belum diajukan kepada pihak manapun. Sumber informasi yang dikutip dalam skripsi ini telah dicantumkan dalam Daftar Pustaka.

Demikian pernyataan ini dibuat dengan sebenarnya. Apabila di kemudian hari terdapat ketidaksesuaian dalam pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Cilacap, 20 Juli 2026



Ahmad Mustaufiq
22EO10006

LEMBAR PENGESAHAN

Skripsi Saudara

Nama : Ahmad Mustaufiq
NIM : 22EO10006
Fakultas/Prodi : Fakultas MIKOM / Informatika
Judul : Penggunaan Metode *Client Side Rendering* (CSR) Pada Aplikasi Monitoring Jaringan FTTH Secara *Real-time* Berbasis Peta Digital

Telah disidangkan oleh Dewan Penguji Fakultas Matematika dan Ilmu Komputer Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap pada hari / tanggal :

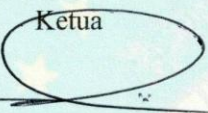
Selasa, 28 Juli 2026

Dan dapat diterima sebagai pemenuhan tugas akhir mahasiswa Program Strata 1 (S.1) Program Studi Informatika (S.Kom.) Fakultas Matematika dan Ilmu Komputer (FMIKOM) pada Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap.

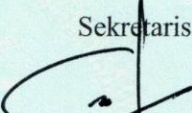
Cilacap, 05 Agustus 2026

Dewan Sidang

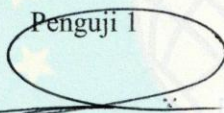
Ketua


Tri Anggoro, M.Kom.
NIDN. 0601128103

Sekretaris


Abdul Haq, SE., M.Cs.
NIDN. 0606067701

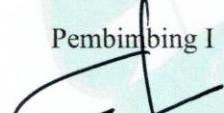
Penguji I


Tri Anggoro, M.Kom.
NIDN. 0601128103

Penguji 2


Eri Nur Shofi'i, S.H., M.A.
NIDN. 0602029501

Pembimbing I


Abdul Haq, SE., M.Cs.
NIDN. 0606067701

Pembimbing II


Dr. M. Khanif, M.Pd.B.I
NIDN. 0617058703

Mengetahui,
Dekan Fakultas Matematika dan Ilmu Komputer


M.T. Abdul Aziz Zein, M.Kom.
NIDN. 2125098601

HALAMAN NOTA KONSULTAN

NOTA KONSULTAN

Hal : Skripsi Saudara Ahmad Mustaufiq
Lampiran : -

Kepada
Yth. Dekan Fakultas Matematika dan Ilmu Komputer
Universitas Nahdlatul Ulama Al Ghazali Cilacap
di Cilacap

Assalamu'alaikum Wr. Wb.

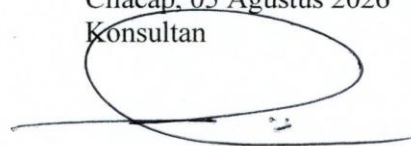
Setelah membaca, memeriksa dan melakukan perbaikan seperlunya maka skripsi saudara/i:

Nama : Ahmad Mustaufiq
NIM : 22EO10006
Prodi : Informatika
Judul : Penggunaan Metode *Client Side Rendering* (CSR) Pada Aplikasi Monitoring Jaringan FTTH Secara *Real-time* Berbasis Peta Digital

Dapat diajukan ke Fakultas Matematika dan Ilmu Komputer, Universitas Nahdlatul Ulama Al Ghazali Cilacap untuk memenuhi syarat memperoleh gelar Strata Satu (S1).

Wassalamu'alaikum Wr. Wb.

Cilacap, 05 Agustus 2026
Konsultan



Tri Anggoro, M.Kom.
NIDN. 0601128103

NOTA PEMBIMBING

Cilacap, 25 Juli 2026

Kepada Yth :
Fakultas Matematika dan Komputer (FMIKOM)
UNUGHA Cilacap
di Cilacap

Assalamu'alaikum Wr. Wb.

Setelah melakukan bimbingan, telaah, arahan dan koreksi tahap penulisan skripsi
saudari:

Nama : Ahmad Mustaufiq
NIM : 22EO10006
Fakultas : Matematika dan Ilmu Komputer
Prodi : Informasi
Judul : Penggunaan Metode *Client Side Rendering* (CSR) Pada Aplikasi
Monitoring Jaringan FTTH Secara *Real-time* Berbasis Peta Digital

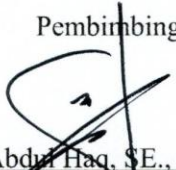
Kami berpendapat bahwa skripsi tersebut sudah dapat diajukan ke sidang skripsi.
Bersamaan ini kami kirimkan skripsi tersebut, semoga dapat segera disidangkan.

Atas perhatiannya kami ucapkan terima kasih.

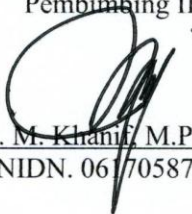
Wassalamu'alaikum Wr. Wb.

Mengetahui,

Pembimbing I


Abdul Haq, S.E., M.Cs.
NIDN. 0606067701

Pembimbing II


Dr. M. Khanif, M.Pd.B.I
NIDN. 0617058703

HALAMAN MOTO

“Kita memasuki era dimana gelar tidak menjamin kompetensi, lulusan tidak menjamin kesiapan berkarya dan bekerja, akreditasi tidak menjamin mutu, masuk kelas tidak menjamin belajar”

(Nadiem Makarim)

”مَنْ أَرَادَ الدُّنْيَا فَعَلَيْهِ بِالْعِلْمِ، وَمَنْ أَرَادَ الْآخِرَةَ فَعَلَيْهِ بِالْعِلْمِ، وَمَنْ أَرَادَهُمَا فَعَلَيْهِ بِالْعِلْمِ
(أَبُو عَبْدِ اللَّهِ مُحَمَّدُ بْنُ إِدْرِيسَ الشَّافِعِيُّ رَحِمَهُ اللَّهُ)

“Jadi orang itu harus seperti falsafahnya genteng, genteng itu dari tanah, tapi mau sabar, diinjak-injak, dibanting, dicetak, dijemur sampai dibakar, akhirnya bisa dimanfaatkan jadi atap rumah”

(KH. Ahmad Hisyam Syafa’at Blokagung)

HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT, Rabb semesta alam yang senantiasa memberikan karunia sehingga penulis mampu menyelesaikan penulisan skripsi ini. Karya ini saya persembahkan kepada :

1. Orang tua dan guru-guru yang selalu mendidik saya, memberikan do'a, dukungan, nasihat dan semangat yang tiada henti.
2. Keluarga dan sahabat yang selalu memberikan do'a dan semangat tiada henti di setiap detik langkahku.
3. Keluarga FMIKOM Angkatan 2022 yang selalu memberikan keceriaan, kebersamaan dan motivasi.
4. Keluarga program studi Informatika, Sistem Informasi, Matematika yang saya banggakan
5. Seluruh teman UNUGHA yang telah memberikan do'a, dukungan, dan semangat.

KATA PENGANTAR

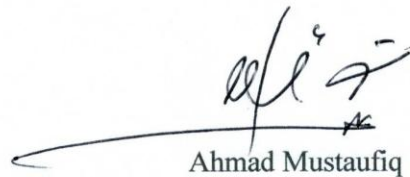
Puji syukur kepada Allah SWT atas segala limpahan nikmat dan karuniaNya, sehingga penulis dapat menyelesaikan penulisan skripsi. Sholawat dan salam senantiasa penulis haturkan kepada Nabi Muhammad SAW sebagai pembimbing seluruh umat manusia.

Skripsi ini tidak akan selesai tanpa adanya bantuan dari banyak pihak, karena itu penulis menyampaikan terima kasih kepada :

1. Bapak Mochamad T. A. Aziz Zein, S.Si., M.Kom, Dekan FMIKOM UNUGHA
2. Bapak Abdul Haq, SE., M.Cs., pembimbing I yang telah dengan sabar memberikan bimbingan dan petunjuk dalam menyelesaikan skripsi ini.
3. Bapak Dr. M. Khanif, M.Pd.B.I, pembimbing II yang telah dengan sabar memberikan bimbingan dan petunjuk dalam menyelesaikan skripsi ini.
4. Ibu Riski Aspriyani, M.Pd, sebagai Pembimbing Akademis
5. Semua pihak PT. Wifian Solution
6. Bapak-Ibu dosen Program Studi Informatika FMIKOM UNUGHA
7. Semua pihak yang tidak dapat kami sebutkan satu persatu

Semoga Allah SWT membalas jerih payah dan pengorbanan yang telah diberikan dengan balasan yang lebih baik. Amiin. Penulis berharap semoga karya kecil ini bermanfaat bagi pembaca.

Cilacap, 20 Juli 2026



Ahmad Mustaufiq

ABSTRAK

AHMAD MUSTAUFIQ. PENGGUNAAN METODE CLIENT SIDE RENDERING (CSR) PADA APLIKASI MONITORING JARINGAN FTTH SECARA REAL-TIME BERBASIS PETA DIGITAL. Dibimbing oleh ABDUL HAQ, SE., M.Cs. dan Dr. M. KHANIF, M.Pd.B.I.

Perkembangan infrastruktur jaringan *Fiber to the Home* (FTTH) menuntut adanya sistem monitoring yang mampu menyajikan informasi jaringan secara cepat, akurat, dan *real-time*. Namun, proses monitoring yang masih dilakukan melalui beberapa aplikasi terpisah dan belum dilengkapi visualisasi lokasi pelanggan menyebabkan identifikasi gangguan menjadi kurang efisien. Selain itu, mekanisme pembaruan data yang masih mengandalkan pemuatan ulang halaman membuat respons aplikasi menjadi kurang optimal. Oleh karena itu, penelitian ini bertujuan mengembangkan aplikasi monitoring jaringan FTTH berbasis peta digital dengan menerapkan metode *Client Side Rendering* (CSR) agar penyajian data menjadi lebih responsif, interaktif, dan efisien.

Penelitian ini menggunakan metode pengembangan sistem yang meliputi identifikasi masalah, studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan analisis hasil. Aplikasi dikembangkan menggunakan React.js untuk membangun antarmuka yang dinamis melalui pendekatan CSR, Node.js sebagai pengolah data dan layanan *backend*, WebSocket untuk mengirimkan pembaruan data secara *real-time* tanpa memuat ulang halaman, Redis sebagai media cache guna meningkatkan kecepatan akses data, Leaflet.js untuk memvisualisasikan lokasi jaringan pada peta digital, serta MikroTik API sebagai sumber data monitoring perangkat dan pengguna jaringan.

Hasil penelitian menunjukkan bahwa aplikasi berhasil mengintegrasikan monitoring pengguna PPPoE, perangkat FTTH, penggunaan bandwidth, dan visualisasi lokasi jaringan dalam satu platform berbasis web. Pengujian fungsional menggunakan metode *Black Box Testing* menunjukkan seluruh fitur berjalan sesuai kebutuhan. Sementara itu, pengujian performa menggunakan Google Lighthouse menunjukkan bahwa halaman Dashboard memperoleh Performance sebesar 91, Accessibility sebesar 80, Best Practices sebesar 100, dan SEO sebesar 92, sedangkan halaman Monitoring PPPoE memperoleh Performance sebesar 90, Accessibility sebesar 75, Best Practices sebesar 100, dan SEO sebesar 91. Selain itu, nilai First Contentful Paint (FCP) sebesar 1,0 detik pada kedua halaman, Largest Contentful Paint (LCP) sebesar 1,3 detik pada Dashboard dan 1,1 detik pada halaman PPPoE, Speed Index masing-masing 1,3 detik dan 1,1 detik, Total Blocking Time (TBT) sebesar 150 ms dan 60 ms, serta Cumulative Layout Shift (CLS) sebesar 0,036 dan 0,141 menunjukkan bahwa implementasi Client Side Rendering (CSR) mampu menghasilkan aplikasi monitoring yang responsif dan efektif dalam mendukung pemantauan jaringan FTTH secara *real-time*.

Kata kunci : *Monitoring jaringan FTTH, peta digital, Client Side Rendering (CSR), sistem monitoring real-time, aplikasi web.*

ABSTRACT

AHMAD MUSTAUFIQ. PENGGUNAAN METODE CLIENT SIDE RENDERING (CSR) PADA APLIKASI MONITORING JARINGAN FTTH SECARA REAL-TIME BERBASIS PETA DIGITAL. Dibimbing oleh ABDUL HAQ, SE., M.Cs. dan Dr. M. KHANIF, M.Pd.B.I.

The rapid expansion of Fiber to the Home (FTTH) network infrastructure requires a monitoring system capable of providing fast, accurate, and real-time network information. However, network monitoring is often performed using multiple separate applications without integrated geographical visualization, making fault identification and network management less efficient. In addition, data update mechanisms that rely on full-page reloading reduce application responsiveness. Therefore, this study aims to develop a digital map-based FTTH network monitoring application by implementing the Client Side Rendering (CSR) approach to provide a more responsive, interactive, and efficient monitoring system.

This research employed a system development methodology consisting of problem identification, literature review, requirements analysis, system design, implementation, testing, and result analysis. The application was developed using React.js to create a dynamic user interface through the Client Side Rendering approach, Node.js as the backend for processing requests and managing application services, WebSocket to deliver real-time data updates without requiring page reloads, Redis as a caching mechanism to improve data access performance, Leaflet.js to visualize network infrastructure and customer locations on an interactive digital map, and the MikroTik API to collect monitoring data from network devices.

Performance evaluation using Google Lighthouse showed that the Dashboard page achieved a Performance score of 91, Accessibility of 80, Best Practices of 100, and SEO of 92, while the PPPoE Monitoring page achieved a Performance score of 90, Accessibility of 75, Best Practices of 100, and SEO of 91. Furthermore, the application recorded a First Contentful Paint (FCP) of 1.0 second on both pages, Largest Contentful Paint (LCP) of 1.3 seconds on the Dashboard and 1.1 seconds on the PPPoE page, Speed Index values of 1.3 and 1.1 seconds, Total Blocking Time (TBT) of 150 ms and 60 ms, and Cumulative Layout Shift (CLS) values of 0.036 and 0.141, indicating that the implementation of Client Side Rendering (CSR) provides a responsive and effective web-based monitoring application for real-time FTTH network monitoring.

Keywords : FTTH network monitoring, digital map, Client Side Rendering (CSR), real-time monitoring system, web-based monitoring.

DAFTAR ISI

HALAMAN PERNYATAAN KEASLIAN SKRIPSI	Error! Bookmark not defined.
LEMBAR PENGESAHAN	Error! Bookmark not defined.
HALAMAN NOTA KONSULTAN.....	iv
HALAMAN NOTA PEMBIMBING.....	vi
NOTA PEMBIMBING	Error! Bookmark not defined.
HALAMAN MOTO	vii
HALAMAN PERSEMBAHAN	viii
KATA PENGANTAR	Error! Bookmark not defined.
ABSTRAK.....	x
ABSTRACT.....	xi
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah.....	2
C. Batasan Masalah	3
D. Tujuan Penelitian	3
E. Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	5
A. Penelitian Terkait	5
B. Landasan Teori.....	13
1. Jaringan <i>Fiber to the Home</i> (FTTH)	13
2. Arsitektur FTTH	13
3. <i>Monitoring</i> Jaringan	13
4. Metode <i>Rendering</i> Pada Aplikasi Web	13
5. Keunggulan CSR dalam <i>Monitoring real-time</i>	15
6. Sistem <i>Real-Time</i>	15
7. Visualisasi Peta Digital	16
8. Arsitektur Sistem <i>Monitoring</i> Jaringan.....	16

BAB III METODOLOGI.....	17
A. Waktu dan Tempat Penelitian.....	17
1. Waktu Penelitian	17
2. Tempat Penelitian.....	17
B. Prosedur Penelitian	17
1. Identifikasi Masalah	18
2. Studi Literatur	18
3. Analisis Kebutuhan Sistem	19
4. Perancangan Sistem	20
5. Implementasi Sistem	34
6. Pengujian Sistem.....	37
7. Analisis Hasil	44
8. Penarikan Kesimpulan	44
C. Jadwal Penelitian	45
BAB IV HASIL DAN PEMBAHASAN.....	46
A. Hasil	46
1. Perancangan Aplikasi Monitoring Jaringan FTTH	46
2. Hasil Pengujian	57
B. Pembahasan.....	63
1. Sistem Monitoring Jaringan FTTH Berbasis <i>Client Side Rendering</i>	63
2. Analisis Keunggulan Sistem	67
3. Analisis Keterbatasan Sistem	68
BAB V KESIMPULAN.....	69
A. Kesimpulan	69
1. Implementasi Aplikasi Monitoring FTTH Berbasis CSR.....	69
2. Hasil Pengujian Fungsional Dan Performa Aplikasi	69
B. Saran	70
DAFTAR PUSTAKA	71
LAMPIRAN.....	74

DAFTAR GAMBAR

Gambar 2. 1 Mind Map Penelitian	10
Gambar 3. 1 Tahapan Penelitian	17
Gambar 3. 2 <i>Use Case Diagram</i> Perancangan Sistem	21
Gambar 3. 3 <i>Activity Diagram Login</i>	22
Gambar 3. 4 <i>Activity Diagram Dashboard Monitoring</i>	23
Gambar 3. 5 <i>Activity Diagram</i> Manajemen Perangkat (Routers)	23
Gambar 3. 6 <i>Activity Diagram</i> Monitoring PPPoE	24
Gambar 3. 7 <i>Activity Diagram</i> Peta Digital	24
Gambar 3. 8 <i>Activity Diagram</i> Logout	25
Gambar 3. 9 <i>Activity Diagram</i> Kelola Lokasi Pelanggan	25
Gambar 3. 10 <i>Class Diagram</i>	27
Gambar 3. 11 <i>Sequence Diagram</i> Monitoring PPPoE	30
Gambar 3. 12 <i>Sequence Diagram</i> Login	29
Gambar 3. 13 <i>Sequence Diagram</i> Manajemen Router MikroTik	31
Gambar 3. 14 <i>Sequence Diagram</i> Monitoring Peta Digital	32
Gambar 3. 15 <i>Sequence Diagram</i> Logout	33
Gambar 3. 16 Login Page	35
Gambar 3. 17 Dashboard dan monitoring	35
Gambar 3. 18 Manajemen perangkat (Routers)	36
Gambar 3. 19 Monitoring PPPoE dan trafik bandwidth	36
Gambar 3. 20 Logout Page	37
Gambar 3. 21 Gantt Chart Penelitian	45
Gambar 4. 1 Halaman Login	48
Gambar 4. 2 Dashboard Monitoring	49
Gambar 4. 3 Monitoring PPPoE	50
Gambar 4. 4 Monitoring FTTH	51
Gambar 4. 5 Manajemen Router	52
Gambar 4. 6 Dashboard Monitoring dan Peta Digital	53
Gambar 4. 7 Logout	55
Gambar 4. 8 Deployment Aplikasi	56
Gambar 4. 9 Hasil Page Dashboard Google Lighthouse	62
Gambar 4. 10 Hasil Page PPPoE Google Lighthouse	62

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	5
Tabel 2. 2 Perbandingan Metode SSR dan CSR.....	15
Tabel 3. 1 Kebutuhan Fungsional Sistem	20
Tabel 3. 2 Kebutuhan Non-Fungsional Sistem	20
Tabel 3. 3 Pengujian Login	38
Tabel 3. 4 Pengujian Dashboard	38
Tabel 3. 5 Pengujian Monitoring PPPoE	39
Tabel 3. 6 Pengujian Peta Digital.....	39
Tabel 3. 7 Pengujian Manajemen Router.....	40
Tabel 3. 8 Pengujian Manajemen FTTH.....	40
Tabel 3. 9 Pengujian Logout	41
Tabel 3. 10 Pengujian Performa.....	42
Tabel 3. 11 Standar Pengujian Google Lighthouse.....	43
Tabel 4. 1 Fitur Aplikasi	47
Tabel 4. 2 Hasil Pengujian Login.....	57
Tabel 4. 3 Hasil Pengujian Dashboard.....	57
Tabel 4. 4 Hasil Pengujian Monitoring PPPoE.....	58
Tabel 4. 5 Hasil Pengujian Peta Digital	58
Tabel 4. 6 Pengujian Manajemen Router	59
Tabel 4. 7 Hasil Pengujian Manajemen FTTH	59
Tabel 4. 8 Hasil Pengujian Logout.....	60
Tabel 4. 9 Hasil Pengujian Performa	60
Tabel 4. 10 Komponen dan Teknologi Sistem.....	64
Tabel 4. 11 Analisis Performance Google Lighthouse Dashboard	66
Tabel 4. 12 Analisis Performance Google Lighthouse PPPoE	66

BAB I PENDAHULUAN

A. Latar Belakang

Perkembangan internet di Indonesia menunjukkan pertumbuhan yang sangat pesat. Berdasarkan survei yang dilakukan oleh Asosiasi Penyelenggara Internet Indonesia (APJII), tingkat penetrasi internet skala nasional mencapai 80,66% di tahun 2025 atau sekitar 229 juta jiwa dari total warga negara Indonesia. Angka tersebut terus meningkat dari tahun ke tahun yang mencerminkan kebutuhan masyarakat terhadap internet yang berkualitas[1].

Jaringan *Fiber to the Home* (FTTH) merupakan salah satu teknologi jaringan *broadband* yang menggunakan serat optik sebagai media transmisi untuk menghubungkan pusat penyedia layanan dengan pelanggan akhir. Teknologi ini berkembang pesat karena menawarkan berbagai keunggulan, seperti kecepatan transmisi data yang tinggi, kapasitas *bandwidth* yang besar, dan ketahanan terhadap interferensi elektromagnetik. Dalam era digital ini, kebutuhan akan jaringan internet yang cepat dan stabil semakin meningkat, terutama di kawasan perkotaan yang memiliki tingkat kepadatan penduduk yang tinggi. Oleh karena itu, evaluasi dan pengembangan kinerja jaringan FTTH menjadi hal yang sangat penting untuk menjamin kualitas layanan yang diberikan kepada pengguna[2].

Client Side Rendering (CSR) merupakan teknologi rendering halaman website yang dilakukan di sisi client menggunakan *browser* pengguna. Pada metode ini, *browser* bertugas memproses dan menampilkan antarmuka aplikasi setelah menerima data dari server. Pendekatan CSR banyak digunakan pada aplikasi *web* modern karena mampu memberikan pengalaman pengguna yang lebih interaktif, dinamis, dan responsif[3], [4].

Metode CSR sangat sesuai diterapkan pada aplikasi *monitoring* jaringan berbasis *real-time* karena memungkinkan pembaruan data dilakukan secara langsung tanpa perlu memuat ulang halaman website secara keseluruhan. Dengan pendekatan ini, proses *monitoring user* PPPoE, *traffic bandwidth*, serta visualisasi peta digital dapat ditampilkan secara cepat dan interaktif. Selain itu, CSR juga mendukung integrasi teknologi *real-time* seperti *WebSocket* yang memungkinkan pertukaran data berlangsung secara kontinu antara server dan *browser* pengguna[3].

Implementasi *Client Side Rendering* (CSR) pada sistem *monitoring* jaringan FTTH berbasis peta digital bertujuan meningkatkan interaktivitas aplikasi, efisiensi pembaruan data, dan pengalaman pengguna dalam proses *monitoring* jaringan. Dengan metode CSR, administrator jaringan dapat memantau perubahan kondisi jaringan secara langsung melalui antarmuka *web* yang responsif dan dinamis[5].

Penerapan *Network Monitoring* Menggunakan The Dude MikroTik dan Notifikasi Pesan dengan Aplikasi Telegram, WhatsApp, dan Email bertujuan

Penerapan sistem *monitoring* jaringan menggunakan The Dude memungkinkan pengiriman notifikasi melalui pesan seperti Telegram, WhatsApp, dan Email, sehingga dapat meningkatkan kualitas layanan kepada pelanggan Ceo Point RT/RW NET. Metode Penelitian ini mengimplementasikan *monitoring* jaringan komputer pada perangkat Cloud Hosted Router yang terhubung dengan Ceo Point RT/RW NET melalui metode tunneling. Hasil dari penelitian menunjukkan bahwa sistem *monitoring* di Ceo Point RT/RW NET mampu melakukan pemantauan jaringan komputer dengan memanfaatkan fitur Maps pada aplikasi *The Dude Client*. Penelitian yang berjudul “Implementasi Jaringan PPPOE Dan Hotspot Server RT/RW NET Berbasis MikroTik Dengan Fitur Mikhmon di ADINET SAMARINDA SEBERANG” Bertujuan untuk menciptakan jaringan yang aman dan stabil dengan biaya yang terjangkau, sehingga seluruh lapisan masyarakat, termasuk mereka dengan kondisi ekonomi rendah, dapat mengakses internet tanpa harus mengeluarkan biaya yang tinggi[6].

Penelitian *monitoring* jaringan sebelumnya cenderung berfokus pada evaluasi performa teknis jaringan FTTH dan implementasi *monitoring* menggunakan tools tertentu yaitu The Dude MikroTik dengan fitur notifikasi. Penelitian yang mengintegrasikan *monitoring* jaringan FTTH secara *real-time* belum banyak dilakukan, visualisasi berbasis peta digital, dan arsitektur *Client Side Rendering* untuk performa *web*[7]. Penelitian ini secara *eksplisit* mengkaji dan mengimplementasikan metode CSR sebagai solusi untuk meningkatkan *initial load performance*, serta menjaga data *monitoring* yang diperbarui secara *real-time*. Oleh karena itu, penelitian ini bukan sekedar mengembangkan sistem *monitoring*, tetapi menawarkan pendekatan arsitektur yang lebih efisien dan adaptif terhadap infrastruktur jaringan FTTH yang tersebar secara geografis, sehingga memberikan kontribusi yang praktis dan konseptual sebagai aplikasi *monitoring* jaringan FTTH modern. Penelitian ini menawarkan pendekatan yang berbeda dengan mengintegrasikan *Client Side Rendering* pada sistem *monitoring* jaringan berbasis peta digital untuk mendapatkan performa yang baik dan efisiensi sistem

B. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana merancang aplikasi *monitoring* jaringan FTTH berbasis peta digital dengan metode *Client Side Rendering* menggunakan MikroTik sebagai *server* sumber data?
2. Bagaimana hasil pengujian fungsional dan performa aplikasi *monitoring* jaringan FTTH berbasis peta digital yang dibangun menggunakan metode *Client Side Rendering* (CSR)?

C. Batasan Masalah

Agar penelitian ini lebih terfokus dan tidak meluas, maka penelitian ini dibatasi pada :

1. Penelitian ini menggunakan perangkat MikroTik RouterOS 7 sebagai server utama dan sumber data *monitoring* jaringan.
2. *Backend* aplikasi dikembangkan menggunakan *Node.js* dengan komunikasi data *real-time* menggunakan *WebSocket*.
3. *Frontend* aplikasi dikembangkan menggunakan *React.js* dengan implementasi metode *Client Side Rendering* (CSR).
4. Antarmuka aplikasi menggunakan *React UI* dan *framework Bootstrap 5* untuk mendukung tampilan yang responsif dan interaktif.
5. Sistem menggunakan *Redis* sebagai media penyimpanan sementara (*cache*) dan pengelolaan komunikasi *real-time*.
6. Visualisasi peta digital hanya menampilkan lokasi *user* PPPoE MikroTik yang ditambahkan secara manual.
7. Sistem *monitoring* hanya berbasis *web* dan tidak mencakup aplikasi *mobile*.

D. Tujuan Penelitian

Berdasarkan uraian penelitian di atas, maka penelitian ini bertujuan untuk :

1. Merancang aplikasi *monitoring* jaringan FTTH berbasis peta digital menggunakan metode *Client Side Rendering* (CSR) dengan integrasi API MikroTik sebagai sumber data *monitoring* jaringan.
2. Melakukan pengujian fungsional dan performa pada aplikasi *monitoring* jaringan FTTH berbasis web untuk mengetahui kemampuan sistem dalam menampilkan data *monitoring* secara *real-time*, responsif, dan interaktif.

E. Manfaat Penelitian

Penelitian yang akan dilakukan diharapkan memberikan manfaat sebagai berikut :

1. Bagi administrator jaringan
 - a. Mempermudah proses pemantauan *user* PPPoE dan perangkat pelanggan secara *real-time*.
 - b. Mempercepat proses identifikasi dan penanganan gangguan.
 - c. Mengurangi beban *monitoring* jaringan secara manual.
2. Bagi teknisi jaringan
 - a. Mempermudah pelacakan lokasi jalur distribusi dan *user* pelanggan.
 - b. Mendukung *troubleshooting* berbasis peta digital.
3. Bagi penelitian

- a. Menjadi model sistem *monitoring* jaringan FTTH yang modern.
- b. Membuka peluang pengembangan tingkat lanjut seperti notifikasi otomatis, analitik *traffic* internet dan pemeliharaan infrastruktur jaringan FTTH.

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Untuk menerapkan sistem ini, penulis mengacu pada berbagai penelitian sebelumnya yang relevan. Penelitian terkait adalah kumpulan sumber penelitian sebelumnya yang mendukung dan menjadi acuan penulis, serta menghindari duplikasi hasil. Selain itu, penelitian tersebut menginspirasi penemuan baru atau pembaruan dengan metode ilmiah. Berikut adalah beberapa penelitian yang dilakukan dalam 5 tahun terakhir yang digunakan sebagai referensi :

Tabel 2. 1 Penelitian Terkait

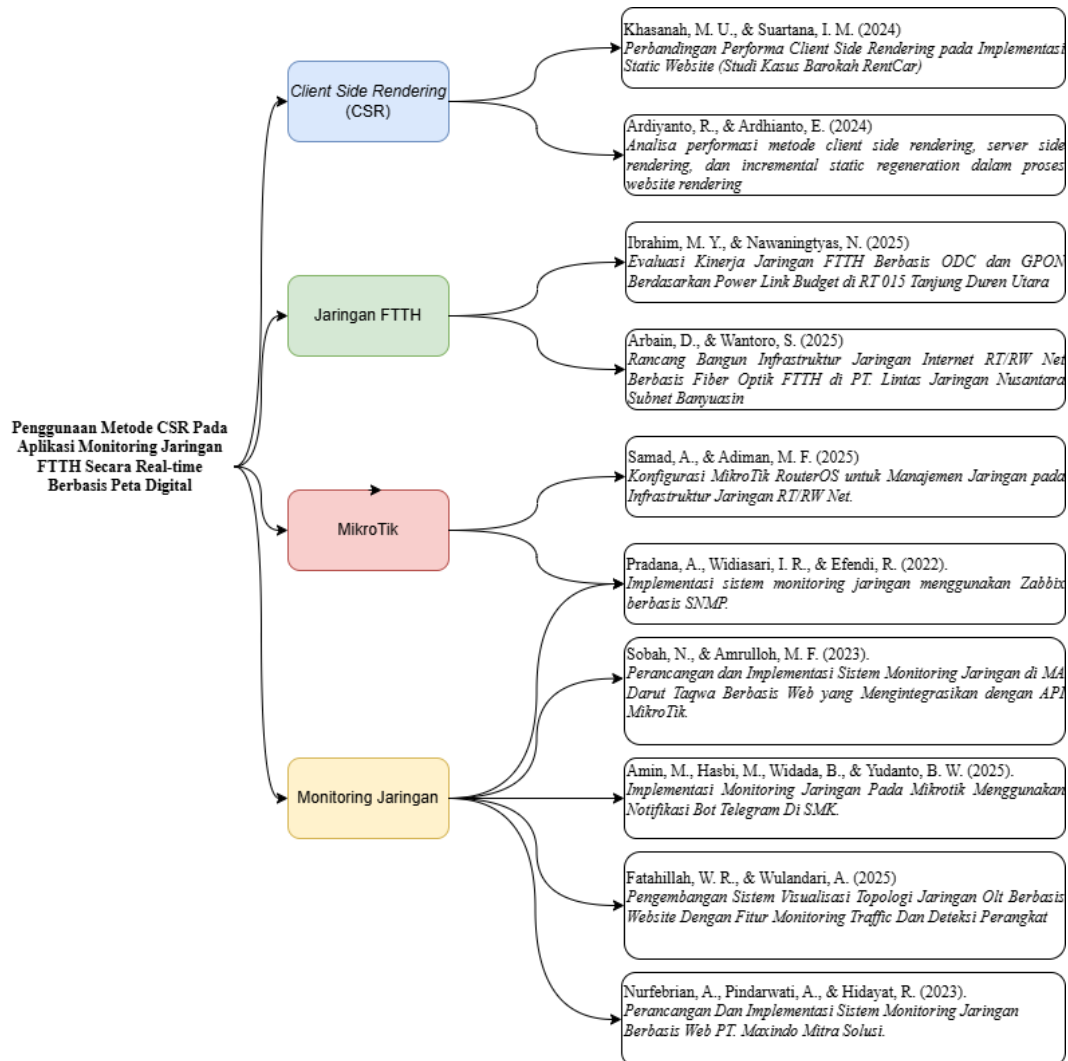
No	Nama Peneliti & Tahun	Keterangan Penelitian
1	Khasanah, M. U., & Suartana, I. M. (2024)	Judul : <i>Perbandingan Performa Client Side Rendering pada Implementasi Static Website (Studi Kasus Barokah RentCar)</i> Metode : Eksperimen. Hasil Penelitian : - Penelitian ini membandingkan performa metode <i>Client Side Rendering</i> (CSR) dan <i>Incremental Static Regeneration</i> (ISR) pada website statis. - Hasil penelitian menunjukkan bahwa ISR memiliki performa <i>loading</i> lebih baik dibandingkan CSR. - Penelitian ini menjadi dasar dalam mengevaluasi performa <i>rendering</i> website pada penelitian yang akan dilakukan.
2	Ardiyanto, R., & Ardianto, E. (2024)	Judul : <i>Analisa performasi metode client side rendering, server side rendering, dan incremental static regeneration dalam proses website rendering</i> Metode : Eksperimen. Hasil Penelitian : - Penelitian ini membandingkan metode CSR, SSR, dan ISR menggunakan <i>framework Next.js</i>. - Hasil penelitian menunjukkan bahwa metode <i>rendering</i> mempengaruhi

		performa website, terutama pada waktu <i>loading</i> dan ukuran halaman.
3	Ibrahim, M. Y., & Nawaningtyas, N. (2025)	Judul : <i>Evaluasi Kinerja Jaringan FTTH Berbasis ODC dan GPON Berdasarkan Power Link Budget di RT 015 Tanjung Duren Utara.</i> Metode : Kuantitatif, Observasi lapangan Hasil Penelitian : - Penelitian ini mengevaluasi kinerja jaringan FTTH berbasis ODC dan GPON/EPON sebagai jalur distribusi layanan ke pelanggan. - Komponen kabel, splitter, konektor yang berkualitas dapat meningkatkan performa distribusi power redaman. - Topologi FTTH yang menggunakan teori dari penelitian ini akan meningkatkan kualitas jaringan internet.
4	Arbain, D., & Wantoro, S. (2025)	Judul : <i>Rancang Bangun Infrastruktur Jaringan Internet RT/RW Net Berbasis Fiber Optik FTTH di PT. Lintas Jaringan Nusantara Subnet Banyuasin</i> Metode : Penelitian terapan (applied research) Hasil Penelitian : - Penelitian ini berfokus pada rencana pembangunan jaringan internet FTTH pada mitra PT. Lintas Jaringan Nusantara. - Topologi jaringan, konfigurasi, infrastruktur yang akan digunakan menjadi titik pembahasan dalam penelitian ini. - Implementasi MikroTik OS sebagai <i>server</i> jaringan dan konfigurasi PPPoE sebagai manajemen <i>user</i> pelanggan.
5	Samad, A., & Adiman, M. F. (2025)	Judul : <i>Konfigurasi MikroTik RouterOS untuk Manajemen Jaringan pada Infrastruktur Jaringan RT/RW Net.</i> Metode : Eksperimen Hasil Penelitian :

		- Penelitian ini berfokus pada MikroTik OS sebagai <i>server</i> manajemen jaringan. - Kesimpulan yang dapat diambil dari penelitian ini antara lain adalah : 1. Integrasi PPPoE multi-profil, <i>load balancing</i>, <i>firewall</i> terbukti efektif. 2. Meningkatkan efisiensi jaringan, konektivitas stabil, pemerataan <i>bandwidth</i>. 3. Sangat efisien untuk diterapkan pada ISP kelas kecil.
6	Pradana, A., Wideasari, I. R., & Efendi, R. (2022).	Judul : <i>Implementasi sistem monitoring jaringan menggunakan Zabbix berbasis SNMP.</i> Software : <i>Zabbix server v5.0</i> Hasil Penelitian : - Penelitian ini menjelaskan instalasi zabbix pada ubuntu <i>server</i> sehingga dapat membaca data dari MikroTik dengan menggunakan protokol <i>Simple Network Management Protocol</i> (SNMP) pada layer aplikasi (OSI layer 7) berbasis TCP/IP. - <i>Monitoring</i> berbasis <i>real-time</i> dapat berjalan maksimal, beberapa fitur yang dapat digunakan adalah : 1. Monitor <i>traffic</i> jaringan masuk dan keluar. 2. Mendeteksi problem jaringan secara <i>real-time</i>. 3. Menampilkan <i>log error</i> dan <i>trigger</i> secara detail. 4. Memberikan informasi performa <i>server</i> dan perangkat jaringan. -
7	Sobah, N., & Amrulloh, M. F. (2023).	Judul : <i>Perancangan dan Implementasi Sistem Monitoring Jaringan di MA Darut Taqwa Berbasis Web yang Mengintegrasikan dengan API MikroTik.</i> Metode : Observasi Hasil Penelitian :

		- Penelitian ini berhasil mengembangkan sistem <i>monitoring</i> jaringan berbasis web menggunakan API MikroTik yang mampu menampilkan <i>monitoring</i> hotspot dan <i>user</i> PPPoE secara <i>real-time</i>.
8	Amin, M., Hasbi, M., Widada, B., & Yudanto, B. W. (2025).	Judul : <i>Implementasi Monitoring Jaringan Pada Mikrotik Menggunakan Notifikasi Bot Telegram Di SMK.</i> Metode : Analisa, Desain, Simulasi Prototyping, Implementasi. Software : Telegram Hasil Penelitian : - Hasil dari implementasi tersebut antara lain : 1. Sistem berhasil mendeteksi perubahan status perangkat secara otomatis (<i>online/offline</i>) 2. Notifikasi dikirim ke telegram secara <i>real-time</i> 3. Administrator jaringan tidak perlu melakukan pengecekan secara manual.
9	Fatahillah, W. R., & Wulandari, A. (2025)	Judul : <i>Pengembangan Sistem Visualisasi Topologi Jaringan Olt Berbasis Website Dengan Fitur Monitoring Traffic Dan Deteksi Perangkat.</i> Metode : Eksperimental Software : <i>Node.js, Express.js, React.js, SNMP (simple network management protocol), Google Lighthouse (chrome devtools), google slide</i> Hasil Penelitian : - Penelitian ini mengembangkan sistem visualisasi topologi jaringan berbasis website dengan fitur monitoring <i>traffic</i> dan deteksi perangkat secara <i>real-time</i>. - Hasil penelitian menunjukkan bahwa visualisasi jaringan dapat membantu efisiensi <i>monitoring</i> jaringan FTTH.
10	Nurfebrian, A., Pindarwati, A., & Hidayat, R. (2023).	Judul : <i>Perancangan Dan Implementasi Sistem Monitoring Jaringan Berbasis Web PT. Maxindo Mitra Solusi.</i>

		Metode : Studi kasus PT. Maxindo Mitra Solusi, Pengumpulan data, Wawancara Software : - <i>Framework Frontend : React.js</i> - Database : MySQL - OS : Windows / Linux - <i>Web Browser : Google Chrome</i> - Code Editor : Visual Studio Code Hasil Penelitian : - Penelitian ini berhasil merancang dan mengimplementasikan sistem <i>monitoring</i> jaringan berbasis <i>web</i> yang mampu memberikan informasi jaringan secara <i>real-time</i>, efisien dan mudah diakses. - Sistem ini memungkinkan administrator dan customer service untuk : 1. Mendeteksi gangguan secara cepat 2. Mengakses informasi status perangkat 3. Memantau <i>traffic</i> 4. Melihat <i>log</i> aktivitas tanpa membuka sistem terpisah.
--	--	--



Gambar 2. 1 *Mind Map* Penelitian

Penyusunan penelitian ini didasarkan pada analisis alur berpikir yang tertuang dalam *Mind Map* serta kajian terhadap berbagai penelitian terdahulu yang disajikan pada Tabel Penelitian Terkait. Berdasarkan kedua instrumen tersebut, penulis mengidentifikasi beberapa kesenjangan penelitian (*research gap*) yang menjadi dasar dilakukannya penelitian ini, sekaligus merumuskan kebaruan (*novelty*) yang ditawarkan.

1. Gap Penelitian

Berdasarkan hasil telaah terhadap penelitian-penelitian terdahulu, ditemukan beberapa kesenjangan penelitian yang mendasari pentingnya penelitian ini dilakukan, yaitu sebagai berikut :

a. Keterbatasan Integrasi Monitoring Jaringan Berbasis Peta Digital

Penelitian terdahulu mengenai monitoring jaringan umumnya berfokus pada pemantauan kondisi perangkat, penggunaan bandwidth, maupun status pengguna menggunakan aplikasi seperti *Zabbix*, *The Dude*, dan API

MikroTik. Meskipun sistem tersebut mampu menampilkan informasi jaringan secara *real-time*, penyajian data masih didominasi dalam bentuk tabel, grafik, atau daftar perangkat sehingga belum memberikan visualisasi geografis yang terintegrasi dengan lokasi pelanggan. Kondisi tersebut menyebabkan proses identifikasi lokasi gangguan dan distribusi jaringan masih kurang efektif, khususnya pada jaringan FTTH yang memiliki cakupan wilayah cukup luas.

- b. Pemanfaatan *Client Side Rendering* pada Sistem Monitoring Jaringan Masih Terbatas

Penelitian mengenai *Client Side Rendering* (CSR) sebagian besar membahas perbandingan performa metode rendering pada aplikasi web modern. Implementasi CSR sebagai pendekatan utama dalam sistem monitoring jaringan FTTH yang memerlukan pembaruan data secara *real-time* masih relatif sedikit ditemukan. Akibatnya, belum banyak penelitian yang mengevaluasi bagaimana CSR dapat menunjukkan performa aplikasi yang baik dan mendukung responsivitas antarmuka, efisiensi pembaruan data, dan pengalaman pengguna pada aplikasi monitoring jaringan.

- c. Masih terbatas ditemukan Integrasi Monitoring *Real-Time* dalam Satu Platform

Beberapa penelitian telah berhasil mengembangkan sistem monitoring berbasis *web* maupun aplikasi visualisasi jaringan. Namun, penelitian tersebut umumnya hanya berfokus pada satu aspek monitoring, seperti status pengguna PPPoE, *traffic bandwidth*, atau visualisasi topologi jaringan. Belum banyak penelitian yang mengintegrasikan seluruh informasi tersebut ke dalam satu aplikasi monitoring berbasis peta digital yang mampu menyajikan data secara *real-time* melalui pendekatan *Client Side Rendering*.

2. Novelty

Kebaruan yang ditawarkan dalam penelitian ini, yang membedakannya dengan penelitian-penelitian sebelumnya, terletak pada beberapa aspek berikut :

- a. Integrasi Monitoring Jaringan FTTH Berbasis Peta Digital

Penelitian ini mengembangkan aplikasi monitoring jaringan FTTH yang mengintegrasikan informasi pengguna PPPoE, penggunaan *bandwidth*, status perangkat, dan lokasi pelanggan ke dalam satu dashboard berbasis peta digital. Pendekatan ini memberikan kemudahan bagi administrator maupun teknisi dalam memantau kondisi jaringan secara lebih informatif dan mempercepat proses identifikasi gangguan berdasarkan lokasi geografis pelanggan.

- b. Penerapan *Client Side Rendering* untuk Monitoring Jaringan *Real-Time*

Berbeda dengan penelitian sebelumnya yang lebih banyak membahas CSR sebagai metode *rendering* pada aplikasi *web* umum, penelitian ini menerapkan *Client Side Rendering* sebagai arsitektur utama dalam sistem monitoring jaringan FTTH. Pendekatan tersebut memungkinkan proses

pembaruan data dilakukan secara dinamis melalui *WebSocket* tanpa memuat ulang halaman, sehingga menghasilkan antarmuka yang lebih responsif dan interaktif untuk kebutuhan monitoring *real-time*.

c. Evaluasi Sistem Melalui Pengujian Fungsional dan Performa

Penelitian ini tidak hanya berfokus pada pembangunan aplikasi monitoring jaringan, tetapi juga melakukan evaluasi terhadap kualitas sistem melalui dua pendekatan pengujian. Pengujian fungsional dilakukan menggunakan metode *Black Box Testing* untuk memastikan seluruh fitur berjalan sesuai kebutuhan pengguna, sedangkan pengujian performa dilakukan menggunakan *Google Lighthouse* dan *Chrome DevTools* untuk mengukur kinerja aplikasi *web* berbasis *Client Side Rendering*. Pendekatan ini diharapkan dapat memberikan gambaran yang lebih komprehensif mengenai efektivitas sistem yang dikembangkan.

B. Landasan Teori

1. Jaringan *Fiber to the Home* (FTTH)

Jaringan *Fiber to the Home* (FTTH) merupakan teknologi jaringan *broadband* yang menggunakan serat optik sebagai media transmisi utama hingga ke pelanggan akhir. Teknologi ini termasuk ke dalam jaringan FTTx yang menawarkan kapasitas *bandwidth* tinggi, latensi rendah, serta stabilitas koneksi yang lebih baik dibandingkan jaringan berbasis tembaga dan wireless[8].

Teknologi ini menawarkan keunggulan dibandingkan jaringan berbasis tembaga, dimana media tembaga umumnya memiliki keterbatasan kecepatan hingga sekitar 4 Mbps, sedangkan FTTH mampu menyediakan kecepatan hingga 100 Mbps atau lebih[9]. Sejalan dengan itu, perkembangan teknologi telekomunikasi saat ini berlangsung sangat pesat, terutama pada teknologi akses data berkecepatan tinggi yang beralih ke penggunaan media serat optik sebagai solusi utama dalam meningkatkan kualitas layanan komunikasi[10].

2. Arsitektur FTTH

Dalam arsitektur FTTH umumnya menggunakan teknologi GPON (*Gigabit Passive Optical Network*) atau EPON (*Ethernet Passive Optical Network*) yang terdiri dari komponen OLT (*Optical Line Terminal*). Struktur ini memungkinkan distribusi jaringan secara luas dengan efisiensi biaya dan performa yang optimal[11].

3. *Monitoring* Jaringan

Monitoring jaringan adalah proses pengawasan terhadap kondisi jaringan untuk memastikan kinerja, ketersediaan dan keamanan sistem tetap optimal. Sistem *monitoring* modern mampu menampilkan data secara *real-time* sehingga mempermudah proses *troubleshooting*[12].

Monitoring jaringan merupakan kegiatan yang dilakukan untuk mengelola suatu sistem jaringan di lokasi atau area tertentu dengan topologi tertentu. Sistem *monitoring* dipergunakan untuk mempermudah tim teknisi dalam melakukan pemantauan secara rutin kondisi jaringan di lapangan. Sistem *monitoring* biasanya memanfaatkan protokol seperti SNMP, ICMP atau API untuk mengumpulkan data dari perangkat jaringan[13], [14].

4. Metode *Rendering* Pada Aplikasi Web

Metode *rendering* merupakan mekanisme yang digunakan aplikasi web untuk menghasilkan dan menampilkan antarmuka pengguna (user interface) pada browser. Secara umum, terdapat dua pendekatan utama yang banyak digunakan dalam pengembangan aplikasi web modern, yaitu *Server Side Rendering* (SSR) dan *Client Side Rendering* (CSR). Perbedaan utama kedua metode tersebut terletak pada lokasi proses *rendering* dilakukan, yaitu di sisi server atau di sisi browser pengguna. Pemilihan metode *rendering* akan

memengaruhi performa aplikasi, waktu muat halaman, interaktivitas, pemanfaatan sumber daya server, serta pengalaman pengguna secara keseluruhan. Oleh karena itu, pemahaman terhadap karakteristik masing-masing metode diperlukan agar pendekatan yang dipilih sesuai dengan kebutuhan sistem yang dikembangkan[15].

Server Side Rendering (SSR) merupakan metode rendering halaman web yang dilakukan di sisi server sebelum halaman dikirimkan kepada browser pengguna. Ketika pengguna mengakses suatu halaman, server akan memproses permintaan, mengambil data yang diperlukan, kemudian menghasilkan dokumen HTML secara lengkap sebelum dikirimkan ke browser. Dengan pendekatan ini, pengguna dapat langsung melihat isi halaman tanpa harus menunggu proses eksekusi JavaScript secara penuh. SSR banyak diterapkan pada aplikasi yang berorientasi pada penyajian konten karena mampu meningkatkan kecepatan pemuatan awal (*initial page load*) dan mempermudah proses pengindeksan oleh mesin pencari (*Search Engine Optimization/SEO*). Meskipun demikian, proses rendering di sisi server menyebabkan beban komputasi server menjadi lebih besar, terutama ketika jumlah pengguna yang mengakses aplikasi meningkat secara bersamaan[16].

Client Side Rendering (CSR) merupakan metode rendering halaman website yang dilakukan di sisi *client* menggunakan *browser* pengguna. Dalam metode ini, *server* hanya mengirimkan data dasar dan file JavaScript, sedangkan proses rendering tampilan dilakukan oleh *browser*[4].

CSR memungkinkan aplikasi *web* menjadi lebih interaktif karena proses pembaruan data dapat dilakukan tanpa melakukan *reload* halaman secara keseluruhan. Pendekatan ini banyak digunakan pada aplikasi berbasis *real-time* seperti *Dashboard Monitoring* dan aplikasi berbasis peta digital [17]. Untuk memberikan gambaran yang lebih jelas mengenai karakteristik kedua metode rendering tersebut, diperlukan perbandingan terhadap beberapa aspek penting, seperti lokasi proses rendering, performa pemuatan awal, interaktivitas aplikasi, optimasi mesin pencari, beban server, serta kesesuaiannya terhadap jenis aplikasi yang dikembangkan. Perbandingan tersebut disajikan pada tabel berikut.

Tabel 2. 2 Perbandingan Metode SSR dan CSR

NO	Aspek	SSR	CSR
1	Lokasi Rendering	Server	Browser pengguna
2	Initial Loading	Lebih cepat	Sedikit lebih lambat
3	Interaktivitas	Terbatas	Sangat tinggi
4	SEO	Sangat baik	Kurang optimal tanpa optimasi tambahan
5	Beban Server	Lebih tinggi	Lebih rendah
6	Beban Browser	Rendah	Lebih tinggi
7	Pembaruan Data	Memerlukan request ulang	Tanpa reload halaman
8	Cocok Digunakan	Website berita, profil perusahaan, e-commerce	Dashboard, monitoring, aplikasi real-time, SPA

5. Keunggulan CSR dalam *Monitoring real-time*

Pada website *monitoring* jaringan real-time, *Client Side Rendering* (CSR) unggul karena halaman *dirender* langsung di browser menggunakan *JavaScript*, sehingga pembaruan tampilan dapat dilakukan secara dinamis tanpa memuat ulang halaman secara penuh[18]. Pendekatan ini cocok untuk *Dashboard Monitoring* karena aplikasi *live dashboard* membutuhkan respons cepat terhadap perubahan data dan interaksi pengguna yang sering, sementara *React* dikenal efektif untuk membangun antarmuka yang responsif dan interaktif.

Selain itu, CSR dapat membantu mengurangi beban render di server karena visualisasi dan pembaruan antarmuka diproses di sisi klien, sehingga server lebih fokus menyediakan data melalui API[19]. Dengan demikian, CSR mendukung pengalaman pemantauan jaringan yang lebih cepat, lebih interaktif, dan lebih nyaman digunakan pada kondisi data yang terus berubah.

6. Sistem *Real-Time*

Sistem *real-time* adalah sistem yang mampu memproses dan menampilkan data secara langsung atau dengan delay yang sangat kecil. Dalam konteks *monitoring* jaringan, *real-time* berarti perubahan kondisi jaringan dapat langsung terlihat oleh administrator[20].

Monitoring jaringan secara *real-time* merupakan kebutuhan penting dalam menjaga kinerja dan keandalan infrastruktur Teknologi Informasi (TI) pada perusahaan modern. Ketika terjadi gangguan pada sistem jaringan atau *server*, keterlambatan dalam mendeteksi dan menindaklanjuti masalah tersebut dapat berdampak pada penurunan produktivitas, bahkan gangguan layanan bisnis[21].

7. Visualisasi Peta Digital

Visualisasi peta digital merupakan teknik untuk menampilkan data geografis dalam bentuk peta interaktif[22]. Dalam sistem *monitoring* jaringan FTTH, peta digital digunakan untuk menampilkan lokasi pelanggan dan perangkat jaringan.

Peta *online* yang didesain pada sebuah aplikasi disajikan sebagai peta digital yang menampilkan titik koordinat. Peta bisa disajikan dalam berbagai cara yang berbeda, mulai dari peta konvensional yang tercetak hingga peta digital yang tampil di layar komputer[23].

8. Arsitektur Sistem *Monitoring* Jaringan

Arsitektur *monitoring* jaringan FTTH (*Fiber to the Home*) umumnya dibangun menggunakan teknologi GPON (*Gigabit Passive Optical Network*) yang terdiri dari perangkat OLT (*Optical Line Terminal*), ODC (*Optical Distribution Cabinet*), ODP (*Optical Distribution Point*), serta ONT/ONU pada sisi pelanggan. OLT berfungsi sebagai pusat pengendali jaringan dan penghubung utama antara *server* dengan pelanggan melalui media serat optik[24]. Pada sistem *monitoring*, data kondisi jaringan seperti status koneksi, trafik, redaman optik, dan kondisi perangkat pelanggan dikumpulkan dari perangkat OLT kemudian dikirim ke *server monitoring* untuk diproses dan ditampilkan secara *real-time* dalam bentuk dashboard atau visualisasi topologi jaringan[25]. Arsitektur *monitoring* ini memudahkan administrator dalam melakukan pengawasan jaringan secara terpusat, mempercepat deteksi gangguan, serta meningkatkan efisiensi pemeliharaan jaringan FTTH karena informasi jaringan dapat dipantau secara langsung melalui aplikasi berbasis *web*.

Sistem *monitoring* jaringan modern umumnya menggunakan arsitektur terintegrasi yang menggabungkan proses pengumpulan data, pemrosesan, serta visualisasi dalam satu kesatuan sistem. Data yang diperoleh dari perangkat jaringan dikumpulkan melalui berbagai metode, kemudian diproses untuk dianalisis, dan selanjutnya disajikan dalam bentuk visualisasi seperti grafik atau *dashboard* guna memudahkan pemantauan oleh administrator[26].

BAB III METODOLOGI

A. Waktu dan Tempat Penelitian

1. Waktu Penelitian

Penelitian ini dilaksanakan pada bulan Maret hingga Juli 2026.

2. Tempat Penelitian

Penelitian ini dilakukan di PT. Wifian Solution cabang Nusawungu yang beralamat di Jl. Jayusman Nusawungu, Kecamatan Nusawungu, Cilacap.

B. Prosedur Penelitian



Gambar 3. 1 Tahapan Penelitian

Penjelasan mengenai beberapa tahapan diatas adalah sebagai berikut:

1. Identifikasi Masalah

Tahap awal penelitian diawali dengan proses identifikasi masalah yang terjadi pada sistem *monitoring* jaringan FTTH. Permasalahan utama yang ditemukan adalah belum optimalnya sistem *monitoring* dalam menampilkan kondisi jaringan secara *real-time*, serta belum adanya visualisasi berbasis peta digital yang memudahkan pemantauan lokasi pelanggan.

Selain itu, sistem *monitoring* konvensional masih memiliki keterbatasan dalam proses pembaruan data secara dinamis karena sebagian besar masih menggunakan metode *reload* halaman secara keseluruhan sehingga mengurangi efisiensi *monitoring* jaringan.

2. Studi Literatur

Studi literatur dilakukan sebagai langkah awal untuk memperoleh landasan teoritis dan konseptual yang mendukung penelitian mengenai aplikasi *monitoring* jaringan FTTH berbasis peta digital dengan pendekatan *Client Side Rendering* (CSR). Kegiatan ini dilakukan melalui pengkajian berbagai sumber ilmiah, seperti jurnal nasional dan internasional, prosiding konferensi, buku referensi, serta dokumentasi teknologi yang relevan dengan topik penelitian. Studi literatur bertujuan untuk memahami perkembangan penelitian terdahulu, mengidentifikasi kesenjangan penelitian (*research gap*), serta menentukan pendekatan yang tepat dalam perancangan sistem yang akan dikembangkan.

Kajian literatur difokuskan pada beberapa aspek utama, yaitu teknologi jaringan *Fiber to the Home* (FTTH), sistem *monitoring* jaringan berbasis *web*, implementasi API MikroTik sebagai sumber data *monitoring*, serta teknologi komunikasi *real-time* menggunakan *WebSocket*. Selain itu, penelitian ini juga mempelajari konsep *Client Side Rendering* (CSR) pada aplikasi *web* modern, khususnya yang dikembangkan menggunakan *React.js*, untuk mengetahui karakteristik, kelebihan, dan keterbatasannya dalam mendukung pembaruan data secara dinamis pada sistem *monitoring* jaringan.

Studi literatur juga dilakukan terhadap penelitian-penelitian yang membahas visualisasi data jaringan menggunakan peta digital. Kajian ini bertujuan untuk memahami metode penyajian informasi geografis yang efektif sehingga kondisi jaringan dan lokasi pelanggan dapat ditampilkan secara lebih informatif dan mudah dipahami oleh administrator jaringan. Berbagai teknologi pemetaan digital seperti *Leaflet.js* dan *OpenStreetMap* dipelajari sebagai alternatif yang dapat diintegrasikan ke dalam sistem *monitoring* yang akan dibangun.

Selain aspek teknis, kajian literatur digunakan untuk menentukan metode pengujian yang sesuai dengan tujuan penelitian. Pengujian fungsional dipelajari melalui metode *Black Box Testing* untuk memastikan setiap fitur sistem berjalan sesuai kebutuhan pengguna. Sementara itu, pengujian performa dipelajari melalui penggunaan *Google Lighthouse* dan *Chrome DevTools* untuk mengukur kinerja aplikasi *web*, khususnya pada aspek waktu muat halaman (*page load time*), kecepatan *rendering*, responsivitas antarmuka, dan efektivitas implementasi CSR dalam mendukung *monitoring* jaringan secara *real-time*.

3. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengetahui kebutuhan yang diperlukan dalam pengembangan aplikasi *monitoring* jaringan FTTH berbasis peta digital. Berdasarkan hasil observasi, proses *monitoring* jaringan masih dilakukan melalui beberapa aplikasi yang terpisah sehingga kurang efektif dalam menampilkan informasi jaringan secara menyeluruh dan *real-time*.

Sistem yang dikembangkan harus mampu menampilkan informasi status pengguna PPPoE, penggunaan *bandwidth*, serta lokasi pelanggan pada peta digital dalam satu platform berbasis *web*. Selain itu, sistem juga menyediakan fitur autentikasi pengguna, pengelolaan data perangkat, dan manajemen pengguna untuk mendukung proses *monitoring* jaringan.

Dari sisi non-fungsional, sistem dirancang berbasis *web* menggunakan *React.js* dengan metode *Client Side Rendering* (CSR) serta memanfaatkan *WebSocket* untuk mendukung pembaruan data secara *real-time*. *Backend* menggunakan *Node.js* dan *Redis* untuk mendapatkan performa yang baik serta efisiensi pengelolaan data *monitoring*.

Hasil analisis kebutuhan ini menjadi dasar dalam perancangan arsitektur sistem, basis data, antarmuka pengguna, dan fitur-fitur yang akan dikembangkan pada penelitian ini.

Tabel 3. 1 Kebutuhan Fungsional Sistem

NO	Kebutuhan Fungsional	Keterangan
1	<i>Login Sistem</i>	Pengguna melakukan autentikasi
2	<i>Dashboard Monitoring</i>	Menampilkan ringkasan kondisi jaringan
3	<i>Monitoring PPPoE</i>	Menampilkan status user PPPoE
4	<i>Monitoring Bandwidth</i>	Menampilkan traffic upload dan download
5	<i>Monitoring FTTH</i>	Menampilkan Perangkat FTTH
6	<i>Monitoring Peta Digital</i>	Menampilkan lokasi pelanggan
7	Manajemen Router MikroTik	Mengelola data perangkat jaringan

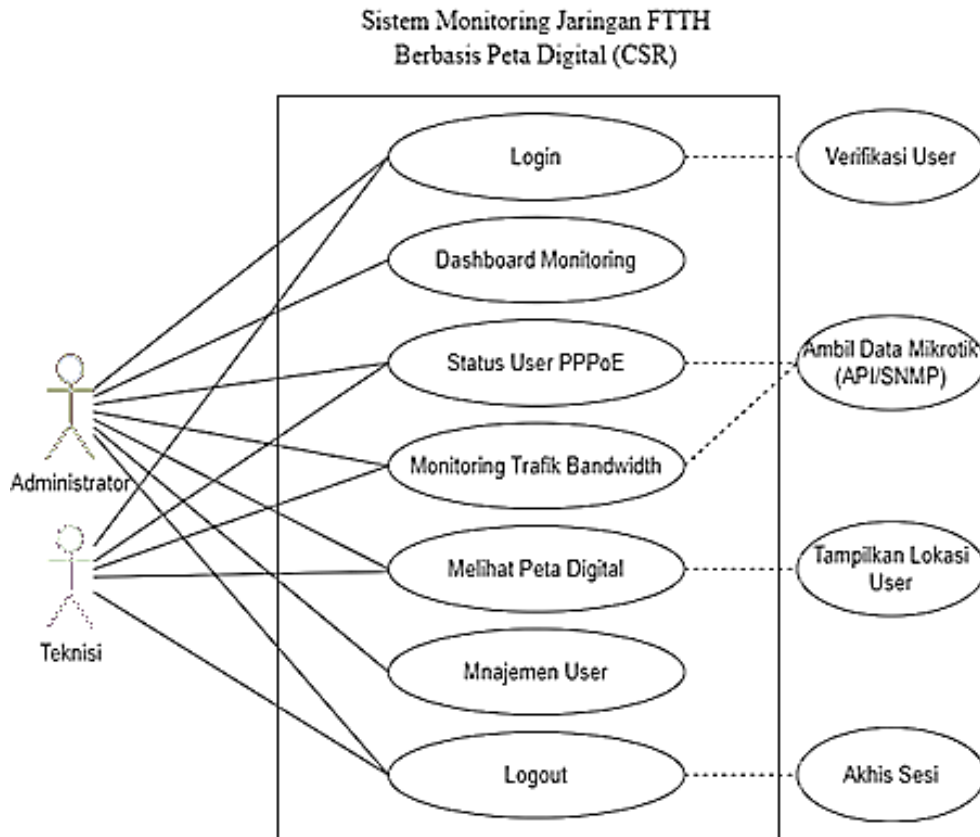
Tabel 3. 2 Kebutuhan Non-Fungsional Sistem

NO	Kebutuhan Non-Fungsional	Keterangan
1	Berbasis <i>Web</i>	Dapat diakses melalui <i>browser</i>
2	<i>Real-Time</i>	Menggunakan <i>WebSocket</i>
3	Responsif	Menggunakan <i>React.js</i> dan <i>Bootstrap</i>
4	Keamanan	Autentikasi pengguna
5	Performa	Mendukung CSR dan <i>cache Redis</i>

4. Perancangan Sistem

Tahap perencanaan sistem dilakukan dengan mengacu pada kebutuhan sistem serta interaksi antar aktor dan sistem yang tergambar pada *Use Case Diagram*. Sistem *monitoring FTTH* Dirancang berbasis *web* dengan pendekatan *Client Side Rendering* (CSR) dan berfokus pada kemudahan *monitoring* secara *real-time* melalui peta digital.

a. *Use Case Diagram*



Gambar 3. 2 *Use Case Diagram* Perancangan Sistem

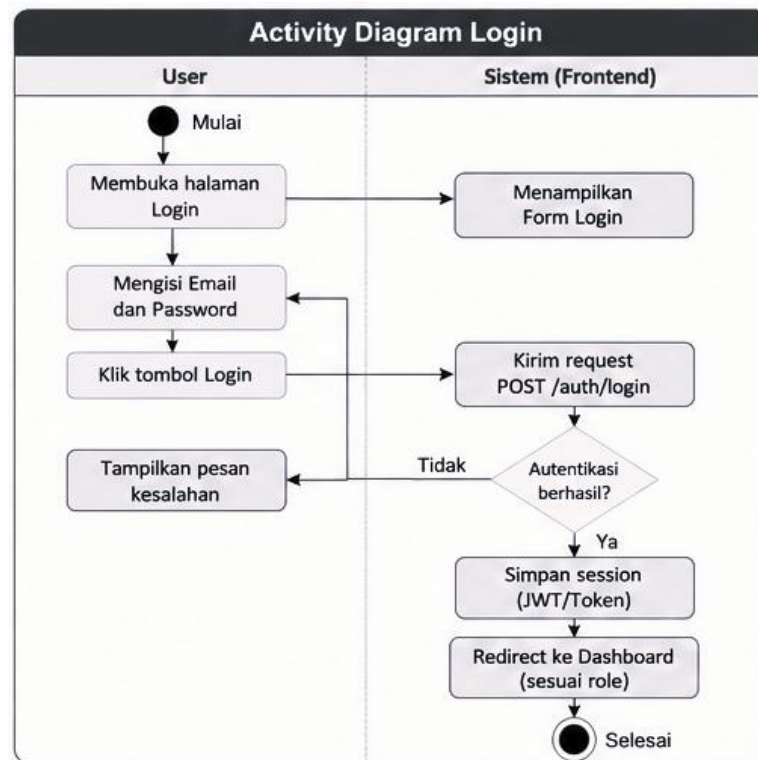
Use Case Diagram digunakan untuk menggambarkan interaksi antara pengguna dan sistem. Berdasarkan hasil analisis kebutuhan, sistem memiliki dua aktor utama yaitu Administrator dan Teknisi. Administrator memiliki hak akses penuh terhadap seluruh fitur sistem, meliputi pengelolaan pengguna, pengelolaan perangkat jaringan, *monitoring* pengguna PPPoE, *monitoring bandwidth*, *monitoring* peta digital, serta pengelolaan *Dashboard Monitoring*. Sementara itu, teknisi menggunakan antarmuka aplikasi yang sama dengan administrator, namun memiliki hak akses yang lebih terbatas. Teknisi hanya dapat mengakses fitur-fitur *monitoring* untuk melihat kondisi jaringan, status pelanggan, penggunaan *bandwidth*, dan lokasi pelanggan pada peta digital tanpa memiliki kewenangan untuk menambah, mengubah, maupun menghapus data yang terdapat pada sistem. Dengan demikian, perbedaan kedua aktor terletak pada hak akses yang diberikan, bukan pada perbedaan antarmuka maupun mekanisme kerja sistem.

Use Case Diagram menunjukkan bahwa seluruh aktivitas pengguna dilakukan di dalam batas sistem (*system boundary*) yang merepresentasikan aplikasi *monitoring* jaringan FTTH berbasis peta digital.

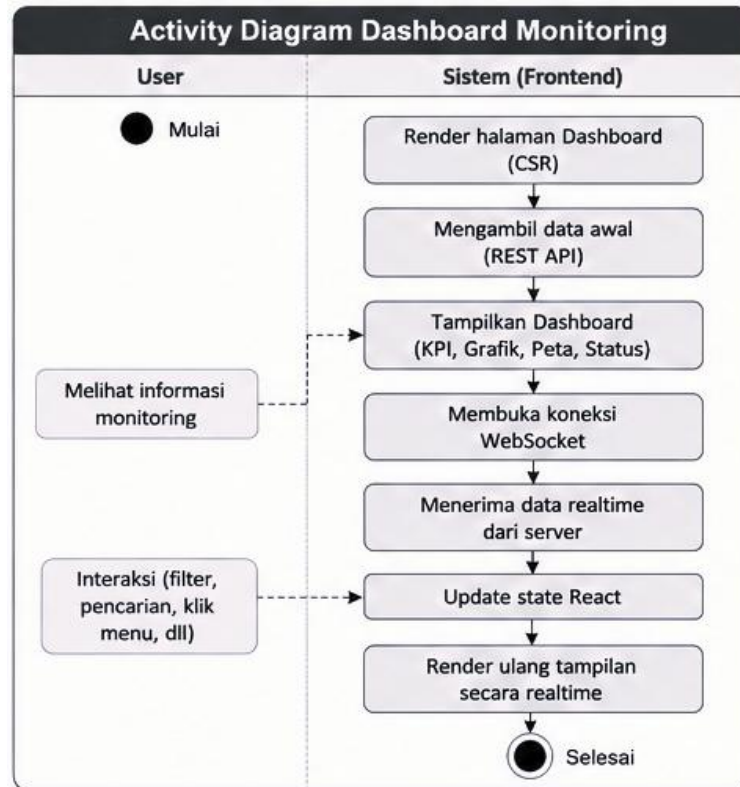
Use Case Diagram memberikan gambaran mengenai batasan sistem serta hak akses masing-masing aktor sehingga kebutuhan fungsional dapat diidentifikasi sejak tahap perancangan. Diagram ini menjadi dasar dalam pengembangan fitur aplikasi agar seluruh fungsi yang dibutuhkan pengguna dapat diimplementasikan secara sistematis.

b. *Activity Diagram*

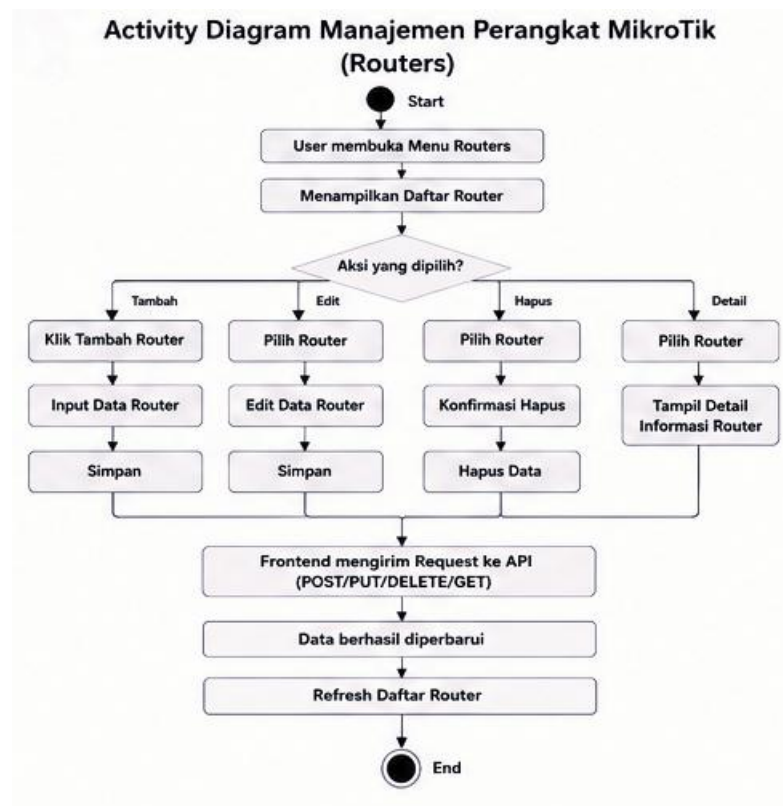
Merupakan proses yang menjelaskan rangkaian aktivitas dalam sistem untuk mempermudah pemahaman. Berikut ini adalah diagram aktivitas dari aplikasi *Client Side Rendering* yang akan dikembangkan. *Activity Diagram* membantu memvalidasi alur kerja setiap proses bisnis dalam aplikasi, mulai dari proses login hingga monitoring jaringan. Dengan adanya diagram ini, urutan aktivitas dan kemungkinan percabangan proses dapat dianalisis sebelum sistem diimplementasikan sehingga dapat meminimalkan kesalahan logika pada tahap pengembangan.



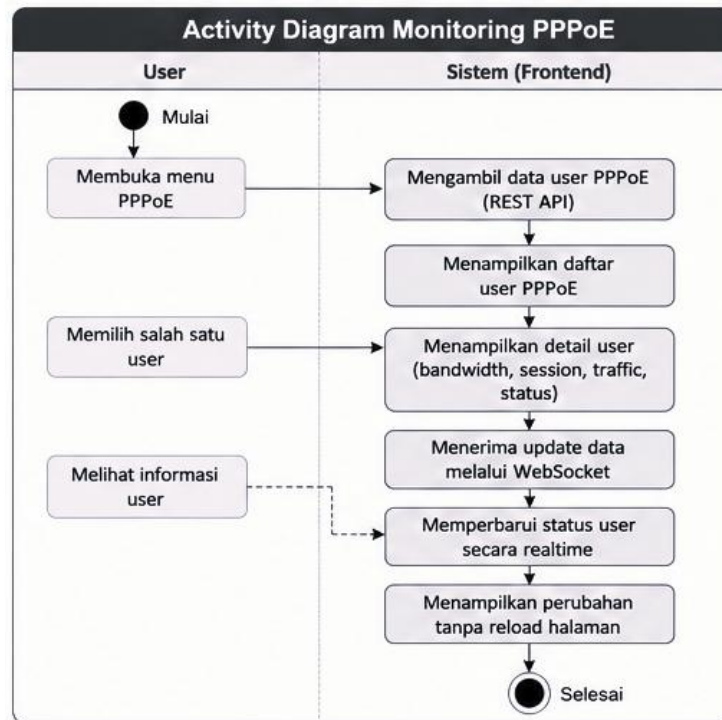
Gambar 3. 3 *Activity Diagram Login*



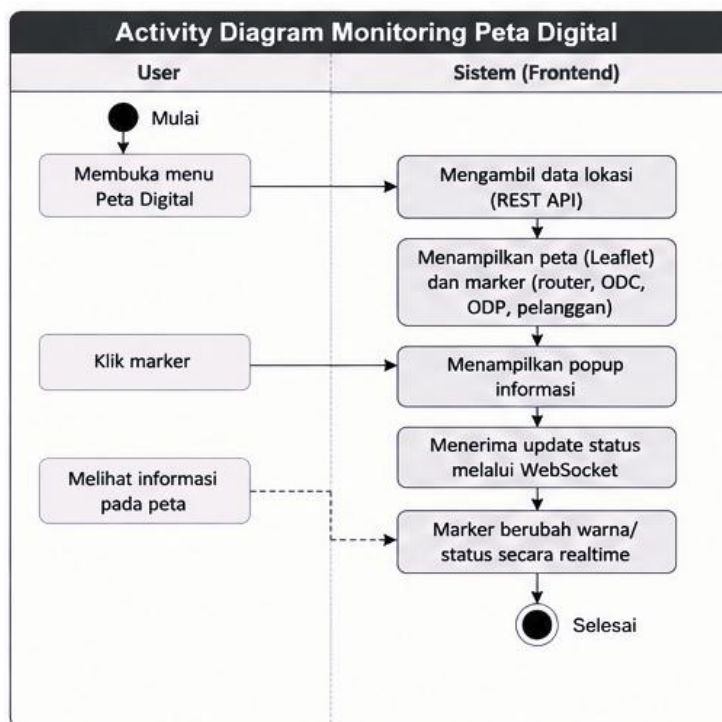
Gambar 3. 4 Activity Diagram Dashboard Monitoring



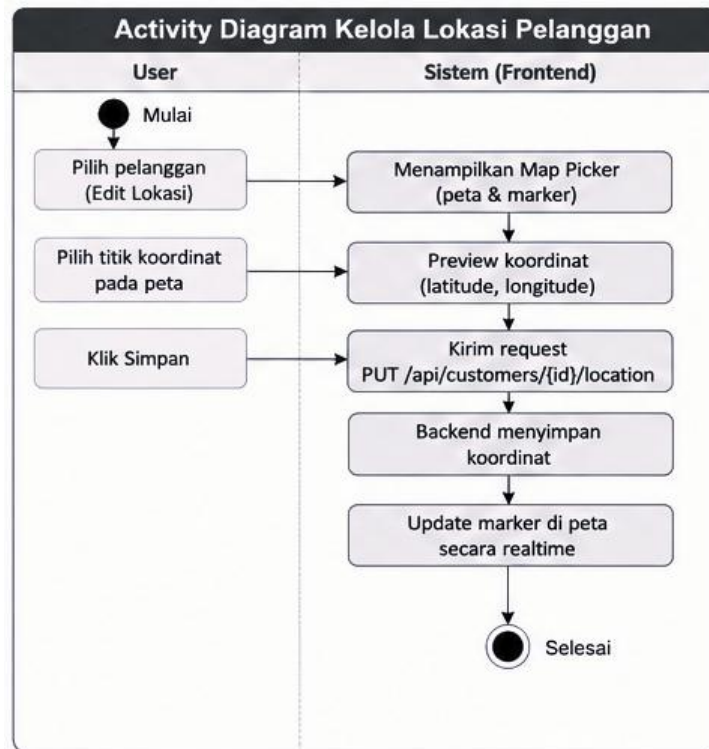
Gambar 3. 5 Activity Diagram Manajemen Router MikroTik



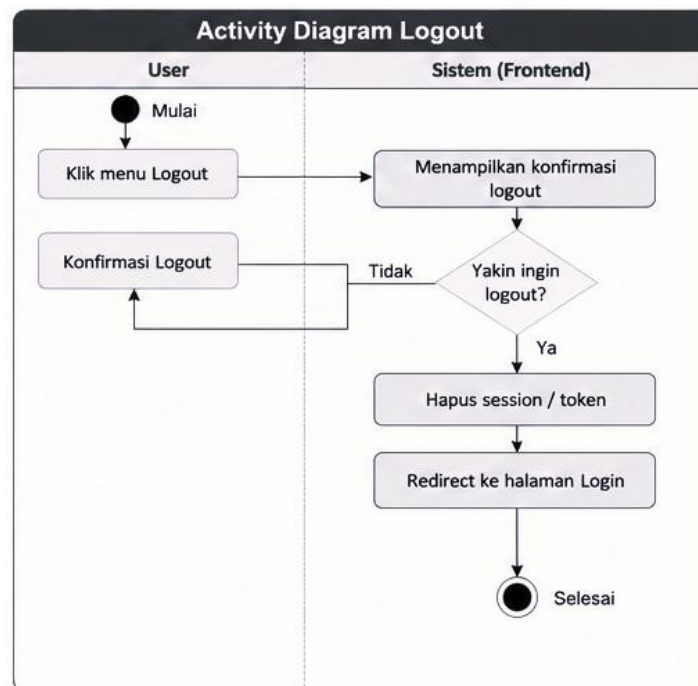
Gambar 3. 6 Activity Diagram Monitoring PPPoE



Gambar 3. 7 Activity Diagram Peta Digital



Gambar 3. 9 *Activity Diagram* Kelola Lokasi Pelanggan



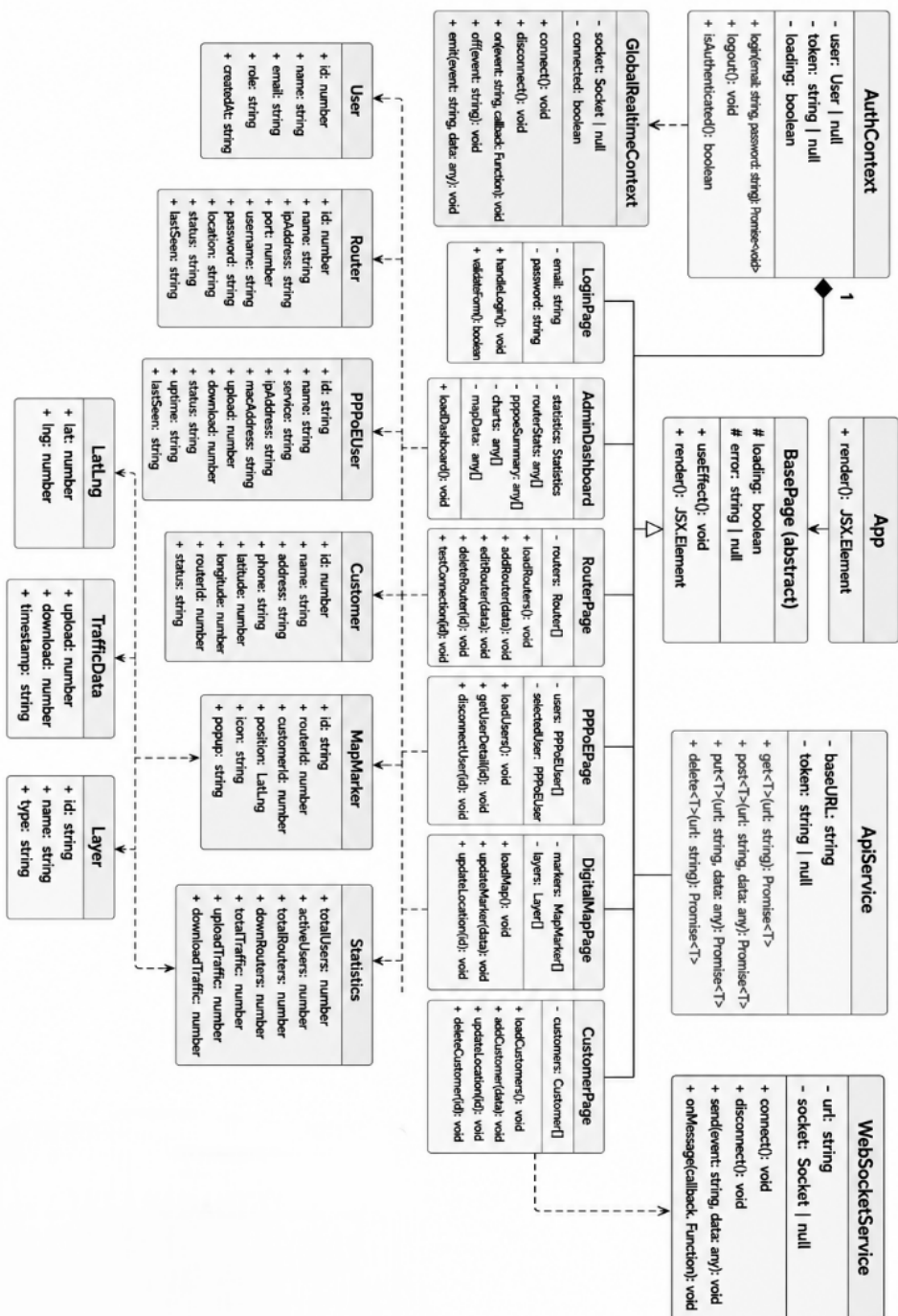
Gambar 3. 8 *Activity Diagram* Logout

c. *Class Diagram*

Class Diagram digunakan untuk menggambarkan struktur statis aplikasi *monitoring* jaringan FTTH yang dibangun menggunakan *React.js* dengan pendekatan *Client Side Rendering* (CSR). Diagram ini menunjukkan hubungan antar komponen utama aplikasi, meliputi halaman (*page*), *context*, *service*, serta model data yang digunakan dalam proses *monitoring* jaringan.

Berdasarkan implementasi sistem, aplikasi terdiri atas beberapa kelas utama, yaitu *App*, *AuthContext*, *GlobalRealtimeContext*, *ApiService*, *LoginPage*, *AdminDashboard*, *RouterPage*, *PPPoEPage*, *DigitalMapPage*, dan *CustomerPage*. Seluruh halaman memanfaatkan *ApiService* untuk melakukan komunikasi dengan *backend* menggunakan *REST API*, sedangkan pembaruan data secara *real-time* dikelola oleh *GlobalRealtimeContext* melalui protokol *WebSocket*.

Selain itu, sistem menggunakan beberapa kelas model seperti *User*, *Router*, *PPPoEUser*, *Customer*, *MapMarker*, dan *Statistics* yang digunakan sebagai representasi data hasil *monitoring* jaringan. Hubungan antar kelas menunjukkan bahwa aplikasi menerapkan pemisahan tanggung jawab (*separation of concern*) sehingga memudahkan proses pengembangan maupun pemeliharaan sistem.

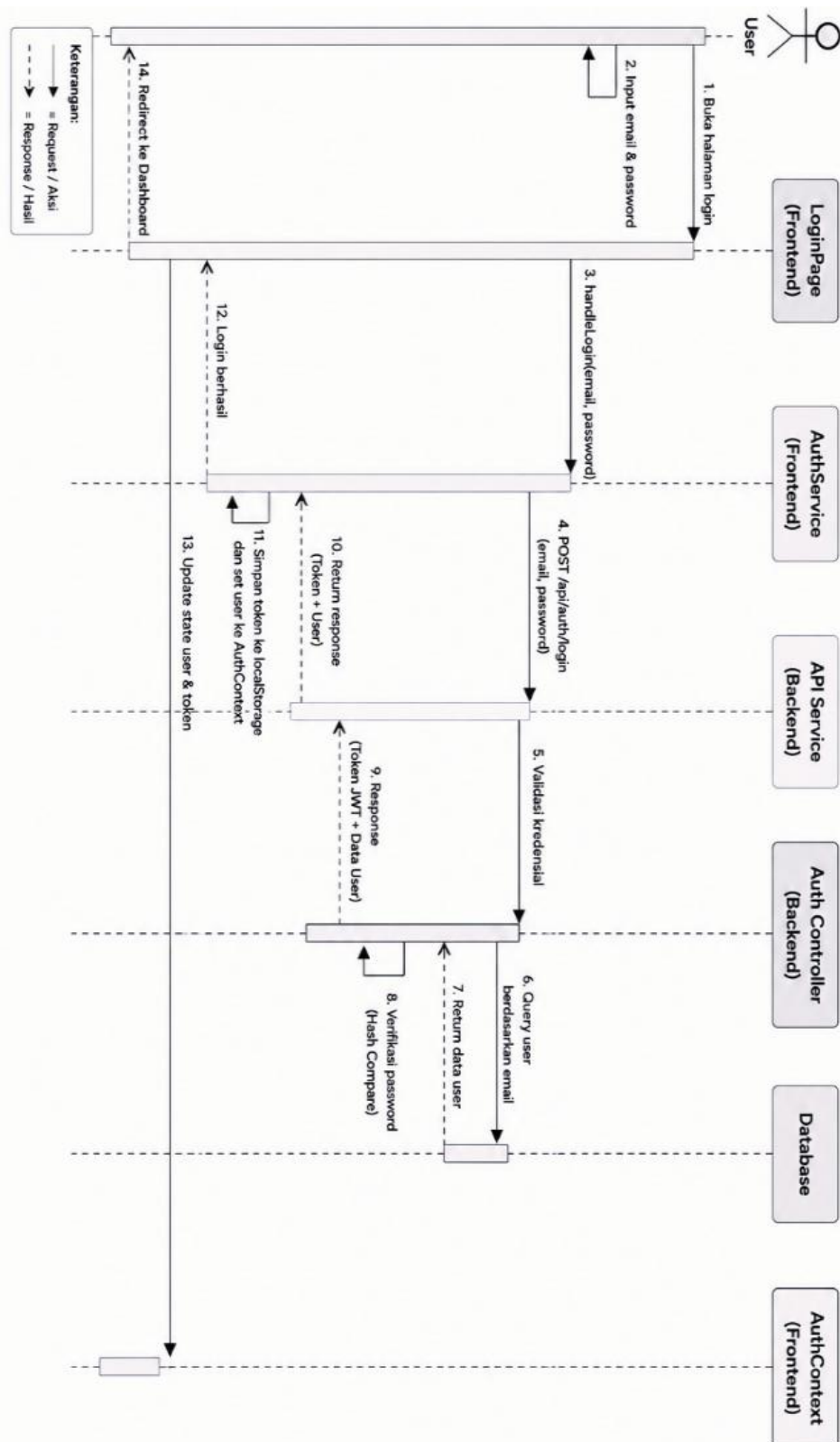


Gambar 3. 10 Class Diagram

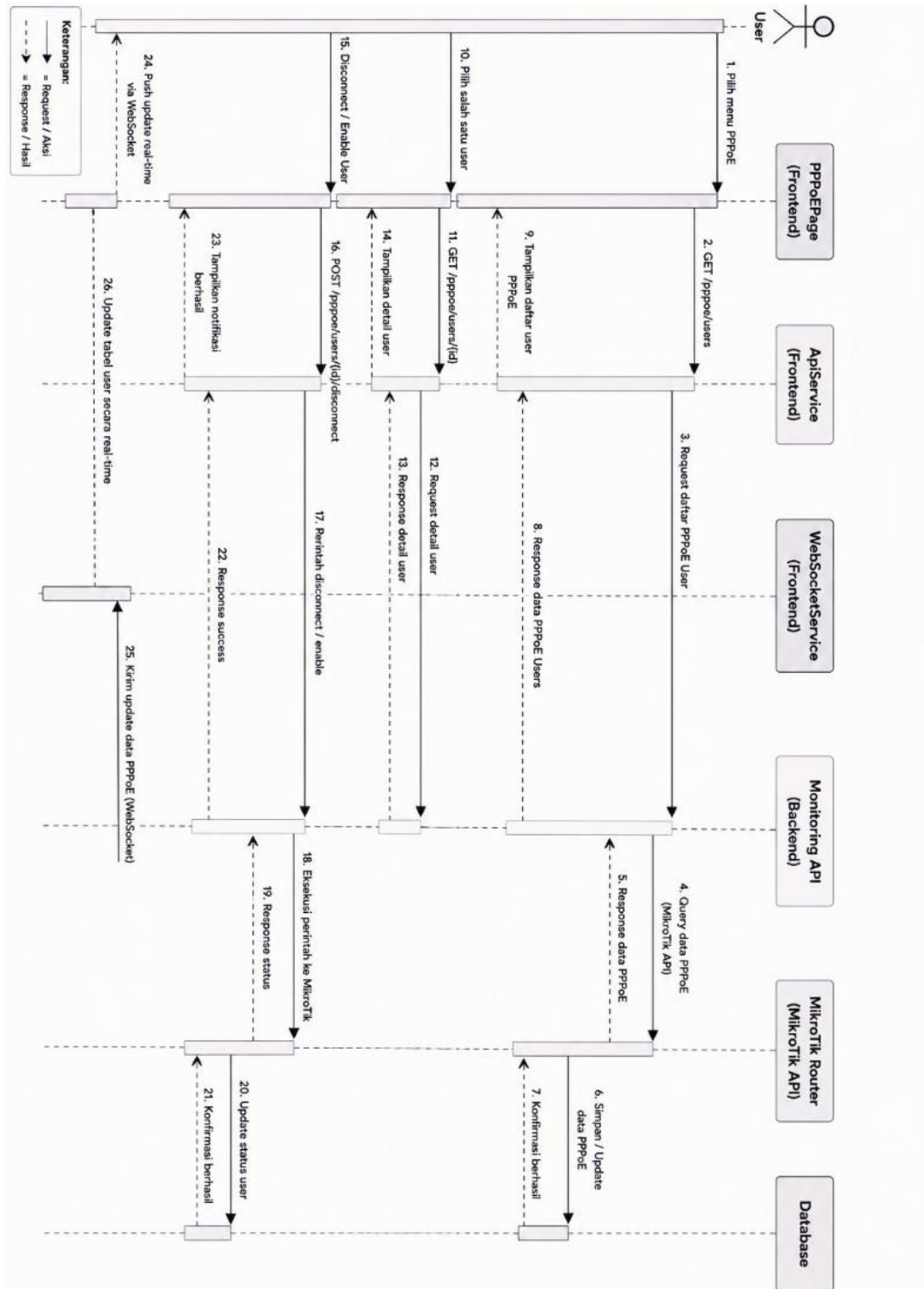
d. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan urutan komunikasi antar objek selama suatu proses berlangsung. Diagram ini menunjukkan interaksi antara pengguna, *frontend*, *backend*, *database*, *API MikroTik*, serta layanan *WebSocket* dalam menjalankan setiap fitur aplikasi *monitoring* jaringan FTTH. Diagram ini berfungsi untuk memastikan bahwa alur komunikasi data antar komponen sistem berjalan sesuai dengan kebutuhan fungsional yang telah dirancang. Selain itu, *Sequence Diagram* membantu memvalidasi proses pertukaran data secara real-time antara *frontend*, *backend*, *WebSocket*, *database*, dan *MikroTik*.

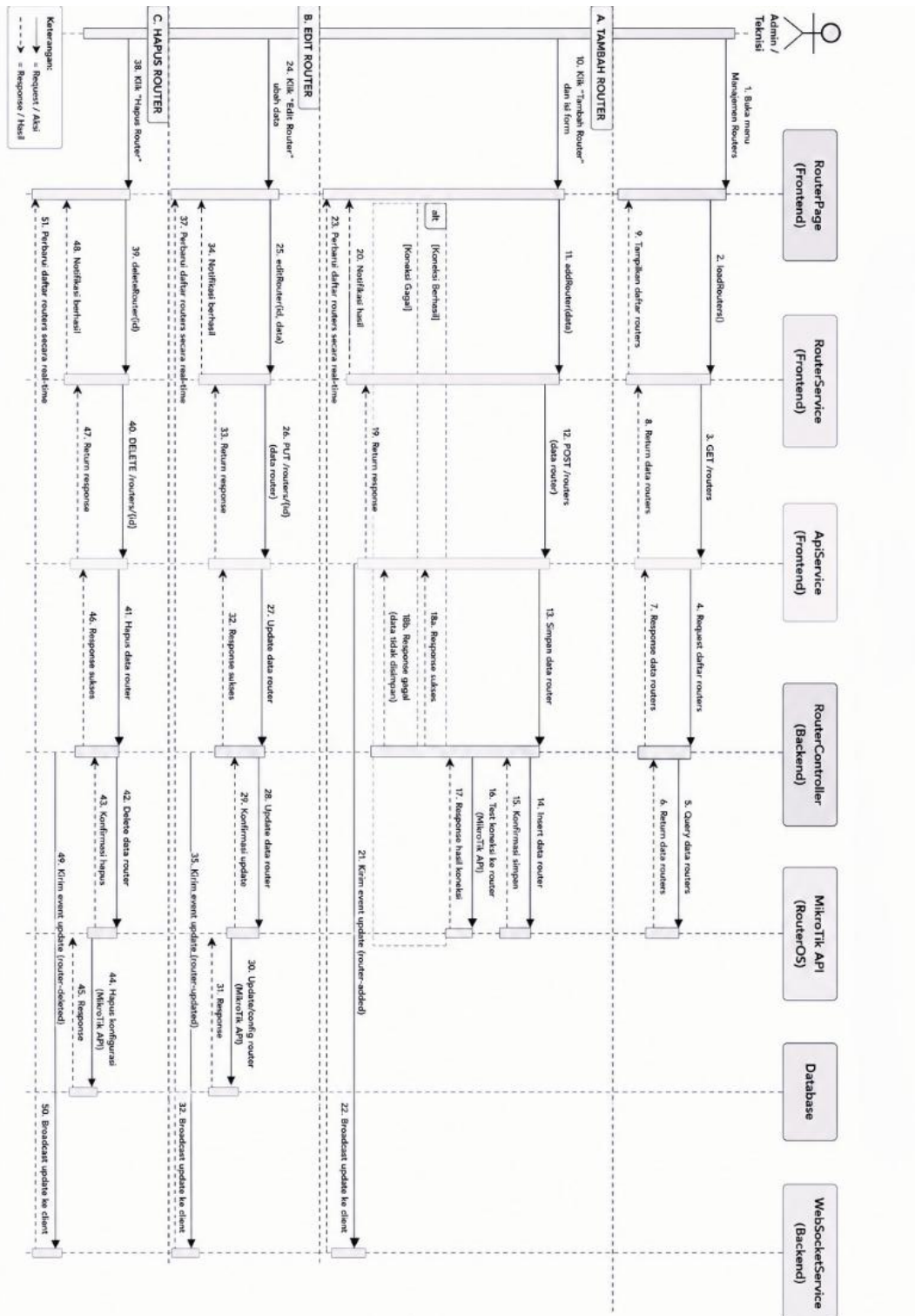
Pada penelitian ini digunakan lima *sequence diagram* utama, yaitu *Sequence Diagram Login*, *Sequence Diagram Monitoring PPPoE*, *Sequence Diagram Monitoring Peta Digital*, *Sequence Diagram Manajemen Router MikroTik*, dan *Sequence Diagram Logout*.



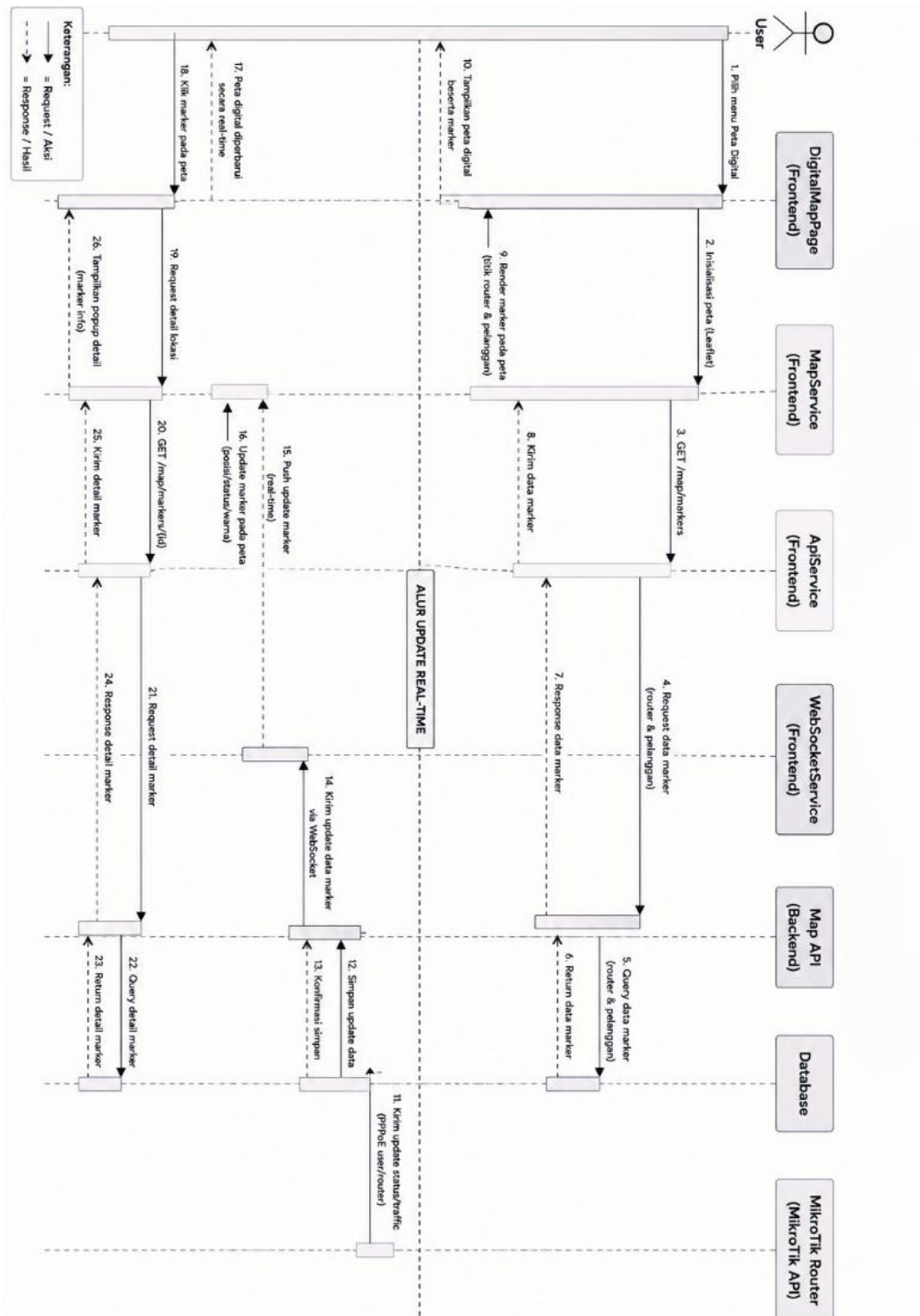
Gambar 3. 11 *Sequence Diagram Login*



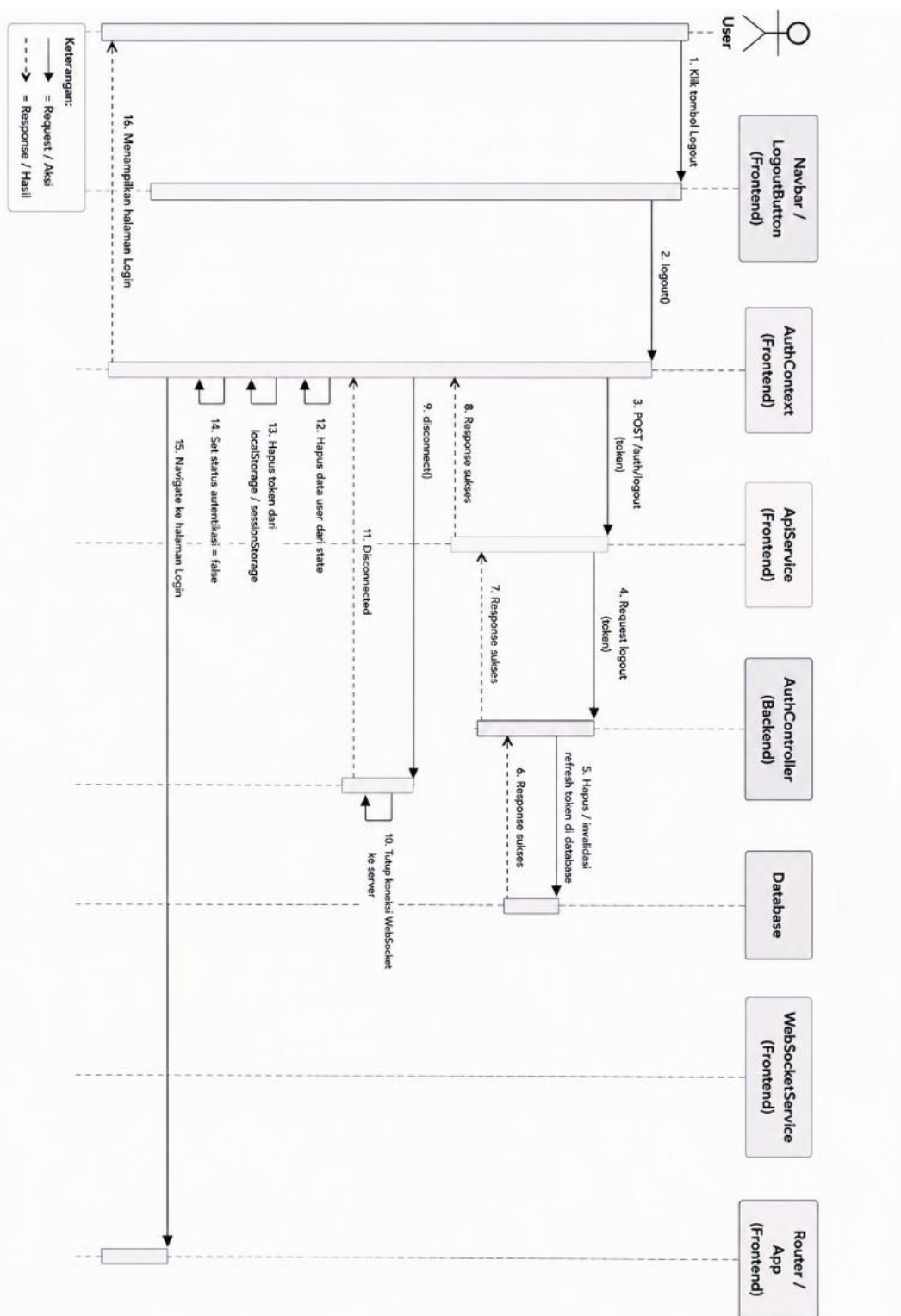
Gambar 3. 12 *Sequence Diagram Monitoring PPPoE*



Gambar 3. 13 *Sequence Diagram* Manajemen Router MikroTik



Gambar 3. 14 Sequence Diagram Monitoring Peta Digital



Gambar 3. 15 *Sequence Diagram Logout*

e. Perancangan Arsitektur Sistem

Arsitektur sistem dirancang menggunakan pendekatan *client-server* yang terdiri dari beberapa lapisan (*layer*). Pada lapisan sumber data (*data source layer*), perangkat MikroTik berfungsi sebagai penyedia informasi jaringan seperti data pengguna PPPoE dan penggunaan *bandwidth*. Data tersebut diakses oleh *backend Node.js* melalui API MikroTik.

Backend berfungsi sebagai pusat pengolahan data, autentikasi pengguna, dan pengelolaan komunikasi *real-time* menggunakan *WebSocket*. Untuk mendapatkan performa yang baik pada sistem, *Redis* digunakan sebagai media *cache* dan penyimpanan sementara data *monitoring*. Selanjutnya data dikirimkan ke *frontend React.js* yang menerapkan metode *Client Side Rendering* (CSR) untuk menampilkan informasi monitoring secara dinamis. Hasil monitoring kemudian dapat diakses oleh administrator dan teknisi melalui *Dashboard Monitoring* dan peta digital berbasis *Leaflet*.

Dengan rancangan tersebut, sistem diharapkan mampu menyediakan informasi jaringan FTTH secara *real-time*, responsif, dan mudah digunakan dalam mendukung proses *monitoring* maupun *troubleshooting* jaringan.

5. Implementasi Sistem

Implementasi sistem merupakan tahap penerapan hasil perancangan ke dalam bentuk aplikasi *monitoring* jaringan FTTH berbasis *web*. Pada tahap ini, seluruh rancangan yang telah dibuat sebelumnya, meliputi arsitektur sistem, basis data, antarmuka pengguna, serta mekanisme komunikasi data *real-time*, diimplementasikan menjadi sebuah sistem yang dapat digunakan untuk melakukan monitoring jaringan.

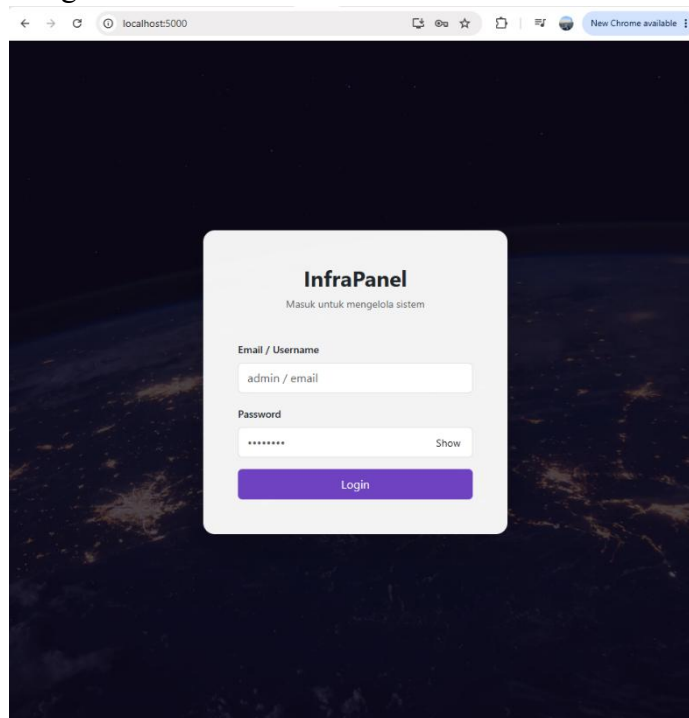
Sistem yang dikembangkan mampu menampilkan informasi pengguna PPPoE, penggunaan *bandwidth*, status perangkat jaringan, serta lokasi pelanggan pada peta digital secara *real-time*. Data *monitoring* diperoleh dari perangkat MikroTik dan diolah oleh sistem sebelum ditampilkan kepada pengguna melalui antarmuka berbasis *web*.

Tahap implementasi juga mencakup integrasi seluruh komponen sistem sehingga proses pertukaran data antara perangkat MikroTik, server aplikasi, basis data, dan antarmuka pengguna dapat berjalan dengan baik. Hasil dari tahap ini adalah aplikasi monitoring jaringan FTTH yang siap digunakan sebagai objek pengujian pada tahap berikutnya.

Implementasi sistem dilakukan berdasarkan rancangan yang telah disusun pada *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram* sehingga seluruh fungsi yang dirancang dapat direalisasikan sesuai

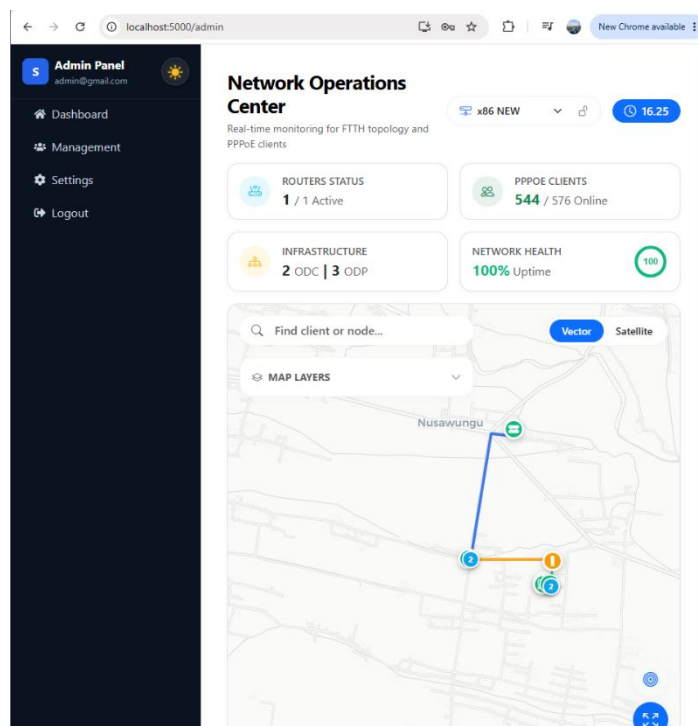
kebutuhan pengguna. Berikut adalah representasi dari implementasi sistem aplikasi monitoring FTTH berbasis peta digital :

a. Login Page



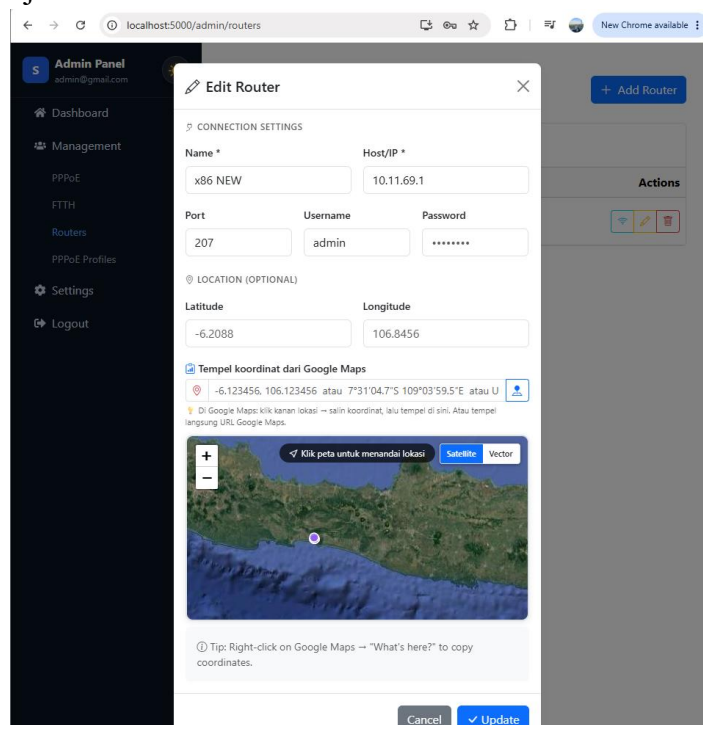
Gambar 3. 16 *Login Page*

b. Dashboard dan monitoring berbasis peta digital



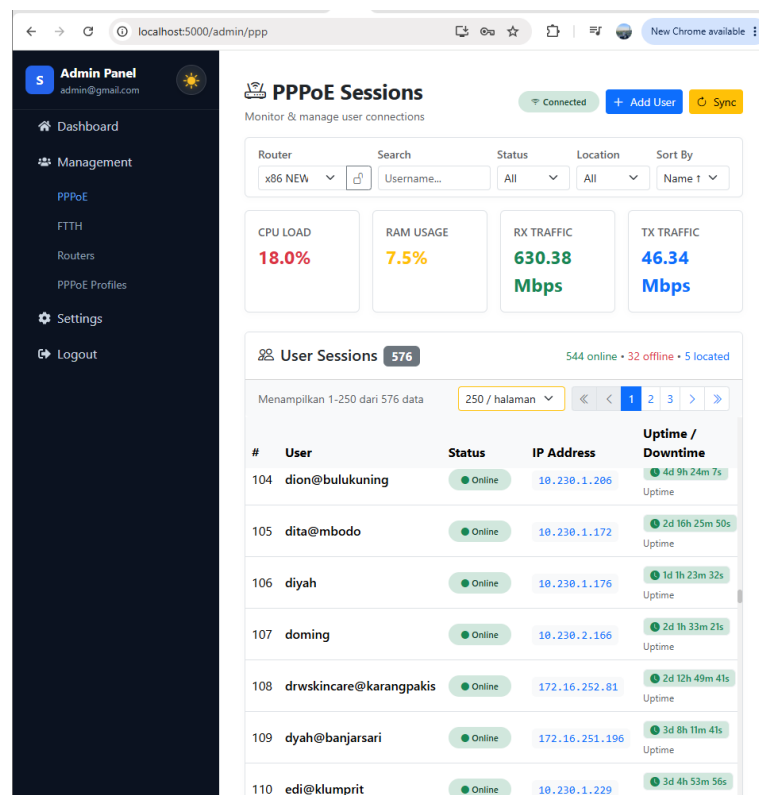
Gambar 3. 17 *Dashboard dan monitoring*

c. Manajemen Router MikroTik



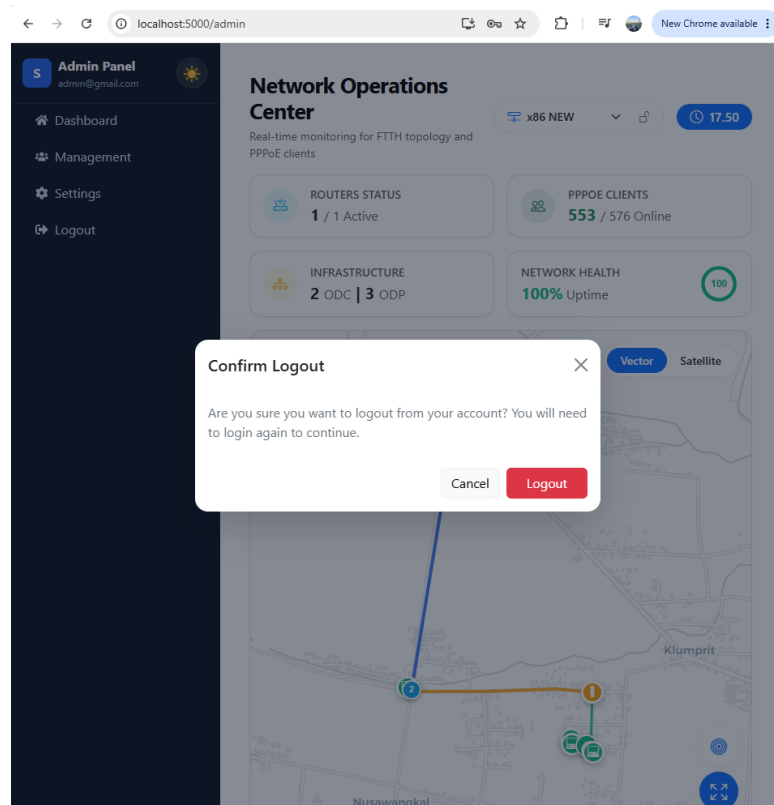
Gambar 3. 18 Manajemen Router MikroTik

d. Monitoring PPPoE dan trafik *bandwidth*



Gambar 3. 19 Monitoring PPPoE dan trafik *bandwidth*

e. Logout page



Gambar 3. 20 Logout Page

Setelah proses pengembangan selesai, aplikasi dideploy menggunakan Docker agar seluruh layanan *frontend*, *backend*, dan cache dapat berjalan pada lingkungan yang konsisten. Selanjutnya aplikasi dipublikasikan melalui Cloudflare menggunakan domain pribadi sehingga dapat diakses secara daring oleh pengguna.

6. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa aplikasi yang dikembangkan dapat berfungsi sesuai kebutuhan serta memiliki performa yang baik saat digunakan. Pengujian dalam penelitian ini terdiri dari pengujian fungsional dan pengujian performa aplikasi *web*.

a. Pengujian Fungsional

Pengujian fungsional dilakukan menggunakan metode *Black Box Testing*. Menurut teori *Black Box Testing*, pengujian dilakukan dengan memeriksa fungsi sistem berdasarkan masukan (*input*) dan keluaran (*output*) tanpa memperhatikan struktur kode program yang digunakan. Metode ini bertujuan untuk memastikan bahwa setiap fitur aplikasi berjalan sesuai dengan kebutuhan yang telah ditentukan.

Pengujian dilakukan pada fitur-fitur utama sistem, meliputi *login*, *Dashboard Monitoring*, *monitoring* pengguna PPPoE, *monitoring bandwidth*, visualisasi peta digital, manajemen pengguna, dan Manajemen Router MikroTik jaringan. Hasil pengujian digunakan untuk mengetahui apakah seluruh fungsi sistem dapat berjalan sesuai dengan kebutuhan pengguna.

1) *Black Box Testing Login*

Tabel 3. 3 Pengujian *Login*

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Validasi form kosong	Kosongkan <i>username</i> dan <i>password</i> kemudian klik <i>Login</i>	Muncul pesan "Username dan Password wajib diisi"	Berhasil { } Tidak Berhasil { }
2	Username salah	Masukkan username salah dan password benar	Muncul pesan "User tidak ditemukan"	Berhasil { } Tidak Berhasil { }
3	Password salah	Masukkan username benar dan password salah	Muncul pesan "Password salah"	Berhasil { } Tidak Berhasil { }
4	Login berhasil	Masukkan username dan password yang valid	Sistem menampilkan Dashboard Monitoring	Berhasil { } Tidak Berhasil { }

2) *Black Box Testing Dashboard Monitoring*

Tabel 3. 4 Pengujian *Dashboard*

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Membuka Dashboard	Membuka Dashboard	Dashboard tampil	Berhasil { } Tidak Berhasil { }
2	Menampilkan statistik	Dashboard selesai dimuat	Jumlah Router, User PPPoE, Traffic tampil	Berhasil { } Tidak Berhasil { }
3	Update real-time	Backend mengirim data baru	Data dashboard berubah otomatis	Berhasil { } Tidak Berhasil { }

3) *Black Box Testing Monitoring PPPoE*

Tabel 3. 5 Pengujian Monitoring PPPoE

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Membuka menu PPPoE dan trafik bandwidth user	Klik menu PPPoE	Daftar user PPPoE tampil	Berhasil { } Tidak Berhasil { }
2	Atur lokasi pelanggan	Klik menu add location	Tempel koordinat dari Google Maps, lokasi tersimpan di database	Berhasil { } Tidak Berhasil { }
3	Perubahan status real-time	Status user berubah di MikroTik	Status pada aplikasi berubah otomatis	Berhasil { } Tidak Berhasil { }
4	Refresh data	Klik tombol Sync	Data terbaru ditampilkan	Berhasil { } Tidak Berhasil { }

4) *Black Box Testing Peta Digital*

Tabel 3. 6 Pengujian Peta Digital

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Membuka dashboard	Klik menu Peta	Peta digital berhasil ditampilkan	Berhasil { } Tidak Berhasil { }
2	Menampilkan marker perangkat dan pelanggan	Peta selesai dimuat	Marker berhasil ditampilkan	Berhasil { } Tidak Berhasil { }
3	Klik marker	Klik salah satu marker	Popup informasi pelanggan tampil	Berhasil { } Tidak Berhasil { }
4	Update marker real-time	Lokasi/status berubah	Marker diperbarui otomatis	Berhasil { } Tidak Berhasil { }

5) *Black Box Testing* Manajemen Router MikroTik

Tabel 3. 7 Pengujian Manajemen Router

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Menampilkan daftar router	Klik menu Router	Daftar router tampil	Berhasil { } Tidak Berhasil { }
2	Tambah router	Klik Add Router, Isi data router valid lalu klik Simpan, Create	Router berhasil ditambahkan	Berhasil { } Tidak Berhasil { }
3	Test koneksi router	Klik Test Connection	Klik Test Connection	Berhasil { } Tidak Berhasil { }
4	Edit router	Ubah data router	Perubahan berhasil disimpan	Berhasil { } Tidak Berhasil { }
5	Hapus router	Klik Hapus kemudian konfirmasi	Router berhasil dihapus	Berhasil { } Tidak Berhasil { }

6) *Black Box Testing* Manajemen FTTH

Tabel 3. 8 Pengujian Manajemen FTTH

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Menampilkan perangkat FTTH	Klik menu FTTH	Daftar perangkat (OLT, ODC, ODP)	Berhasil { } Tidak Berhasil { }
2	Tambah perangkat FTTH	Klik tambah OLT, ODC, ODP	Perangkat FTTH berhasil ditambahkan	Berhasil { } Tidak Berhasil { }
3	Tambah User PPPoE pada ODP	Klik port ODP kosong, tambahkan User PPPoE	User PPPoE berhasil ditambahkan	Berhasil { } Tidak Berhasil { }
4	Hapus perangkat FTTH	Klik Hapus kemudian konfirmasi	Perangkat berhasil dihapus	Berhasil { } Tidak Berhasil { }

7) *Black Box Testing Logout*

Tabel 3. 9 Pengujian Logout

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Logout berhasil	Klik tombol Logout	Kembali ke halaman Login	Berhasil { } Tidak Berhasil { }
2	Session berakhir	Login kembali tanpa token	Harus melakukan login ulang	Berhasil { } Tidak Berhasil { }

b. Pengujian Performa

Pengujian performa dilakukan untuk mengevaluasi kualitas kinerja aplikasi monitoring jaringan FTTH yang dikembangkan menggunakan metode *Client Side Rendering* (CSR). Pengujian ini bertujuan untuk mengetahui kemampuan aplikasi dalam memberikan waktu muat (*loading time*) yang cepat, responsivitas yang baik, stabilitas tampilan, serta penerapan praktik pengembangan web sesuai dengan standar yang direkomendasikan. Pada penelitian ini, pengujian performa menggunakan *Google Lighthouse* yang terintegrasi pada *Google Chrome DevTools*. *Lighthouse* merupakan alat pengujian yang dikembangkan oleh *Google* untuk menganalisis kualitas aplikasi *web* berdasarkan berbagai metrik performa dan kualitas halaman.

Parameter yang digunakan pada pengujian ini meliputi *Performance Score*, *Accessibility*, *Best Practices*, *SEO*, *First Contentful Paint* (FCP), *Largest Contentful Paint* (LCP), *Speed Index* (SI), *Total Blocking Time* (TBT), dan *Cumulative Layout Shift* (CLS). Masing-masing parameter memiliki fungsi dan tujuan pengukuran yang berbeda untuk mengevaluasi performa aplikasi secara menyeluruh. Penjelasan setiap parameter yang digunakan dalam penelitian ini disajikan pada Tabel.

Tabel 3. 10 Pengujian Performa

NO	Parameter	Tools	Fungsi
1	Performance Score	Google Lighthouse	Menunjukkan nilai keseluruhan performa halaman berdasarkan beberapa metrik Lighthouse.
2	Accessibility	Google Lighthouse	Mengukur tingkat aksesibilitas aplikasi bagi seluruh pengguna, termasuk pengguna berkebutuhan khusus.
3	Best Practices	Google Lighthouse	Menilai apakah aplikasi telah menerapkan praktik pengembangan web yang direkomendasikan Google.
4	SEO	Google Lighthouse	Mengukur tingkat optimasi struktur halaman untuk mesin pencari.
5	First Contentful Paint (FCP)	Google Lighthouse	Waktu yang dibutuhkan hingga konten pertama berhasil ditampilkan kepada pengguna.
6	Largest Contentful Paint (LCP)	Google Lighthouse	Waktu yang dibutuhkan hingga elemen utama halaman selesai dimuat.
7	Speed Index (SI)	Google Lighthouse	Mengukur kecepatan tampilan visual halaman selama proses pemuatan.
8	Total Blocking Time (TBT)	Google Lighthouse	Total waktu ketika thread utama browser terblokir sehingga pengguna belum dapat berinteraksi secara optimal.
9	Cumulative Layout Shift (CLS)	Google Lighthouse	Mengukur kestabilan tata letak halaman selama proses pemuatan. Nilai semakin kecil menunjukkan tampilan semakin stabil.

Berdasarkan parameter-parameter tersebut, hasil pengujian akan dianalisis dengan membandingkan nilai yang diperoleh aplikasi terhadap standar kualitas yang direkomendasikan oleh *Google Lighthouse*. Semakin tinggi nilai *Performance Score*, *Accessibility*, *Best Practices*, dan SEO, serta semakin kecil nilai FCP, LCP, *Speed Index*, TBT, dan CLS, maka semakin baik kualitas performa aplikasi yang dikembangkan. Hasil pengujian ini selanjutnya akan dibahas pada BAB IV untuk mengetahui sejauh mana penerapan metode *Client Side Rendering* (CSR) mampu memiliki performa baik pada aplikasi monitoring jaringan FTTH secara *real-time*.

Untuk memberikan interpretasi yang objektif terhadap hasil pengujian, penelitian ini menggunakan standar penilaian yang mengacu pada rekomendasi Google Lighthouse dan Core Web Vitals sebagaimana disajikan pada Tabel.

Tabel 3. 11 Standar Pengujian Google Lighthouse

NO	Parameter	Standar Baik	Kategori
1	Performance Score	≥ 90	Baik
2	Accessibility	≥ 90	Baik
3	Best Practices	≥ 90	Baik
4	SEO	≥ 90	Baik
5	First Contentful Paint (FCP)	$\leq 1,8$ s	Baik
6	Largest Contentful Paint (LCP)	$\leq 2,5$ s	Baik
7	Speed Index (SI)	$\leq 3,4$ s	Baik
8	Total Blocking Time (TBT)	≤ 200 ms	Baik
9	Cumulative Layout Shift (CLS)	$\leq 0,1$	Baik

Standar pada Tabel digunakan sebagai acuan dalam mengevaluasi hasil pengujian performa aplikasi yang dikembangkan. Nilai yang diperoleh dari setiap parameter akan dibandingkan dengan standar tersebut untuk menentukan apakah performa aplikasi telah memenuhi kategori yang direkomendasikan. Dengan menggunakan acuan yang terukur, hasil pengujian tidak hanya menunjukkan nilai numerik, tetapi juga memberikan interpretasi mengenai tingkat kualitas performa aplikasi secara objektif sehingga dapat menjadi dasar dalam menganalisis efektivitas implementasi metode *Client Side Rendering* (CSR) pada sistem monitoring jaringan FTTH.

7. Analisis Hasil

Analisis hasil merupakan tahap yang dilakukan setelah proses pengujian sistem selesai dilaksanakan. Pada tahap ini, data yang diperoleh dari pengujian fungsional dan pengujian performa akan dianalisis untuk mengetahui tingkat keberhasilan sistem yang telah dikembangkan.

Hasil pengujian fungsional menggunakan metode *Black Box Testing* akan dianalisis untuk memastikan bahwa seluruh fitur aplikasi dapat berjalan sesuai dengan kebutuhan yang telah ditetapkan. Analisis dilakukan dengan membandingkan hasil pengujian terhadap fungsi yang dirancang pada tahap analisis kebutuhan dan perancangan sistem.

Selain itu, hasil pengujian performa menggunakan *Google Lighthouse* dan *Chrome DevTools* akan dianalisis untuk mengevaluasi kinerja aplikasi berbasis *Client Side Rendering* (CSR). Analisis dilakukan berdasarkan parameter performa seperti *Performance Score*, *First Contentful Paint* (FCP), *Largest Contentful Paint* (LCP), *Speed Index*, serta waktu pembaruan data secara *real-time*. Hasil analisis tersebut akan digunakan untuk menilai efektivitas penerapan CSR dalam mendukung aplikasi monitoring jaringan FTTH berbasis peta digital.

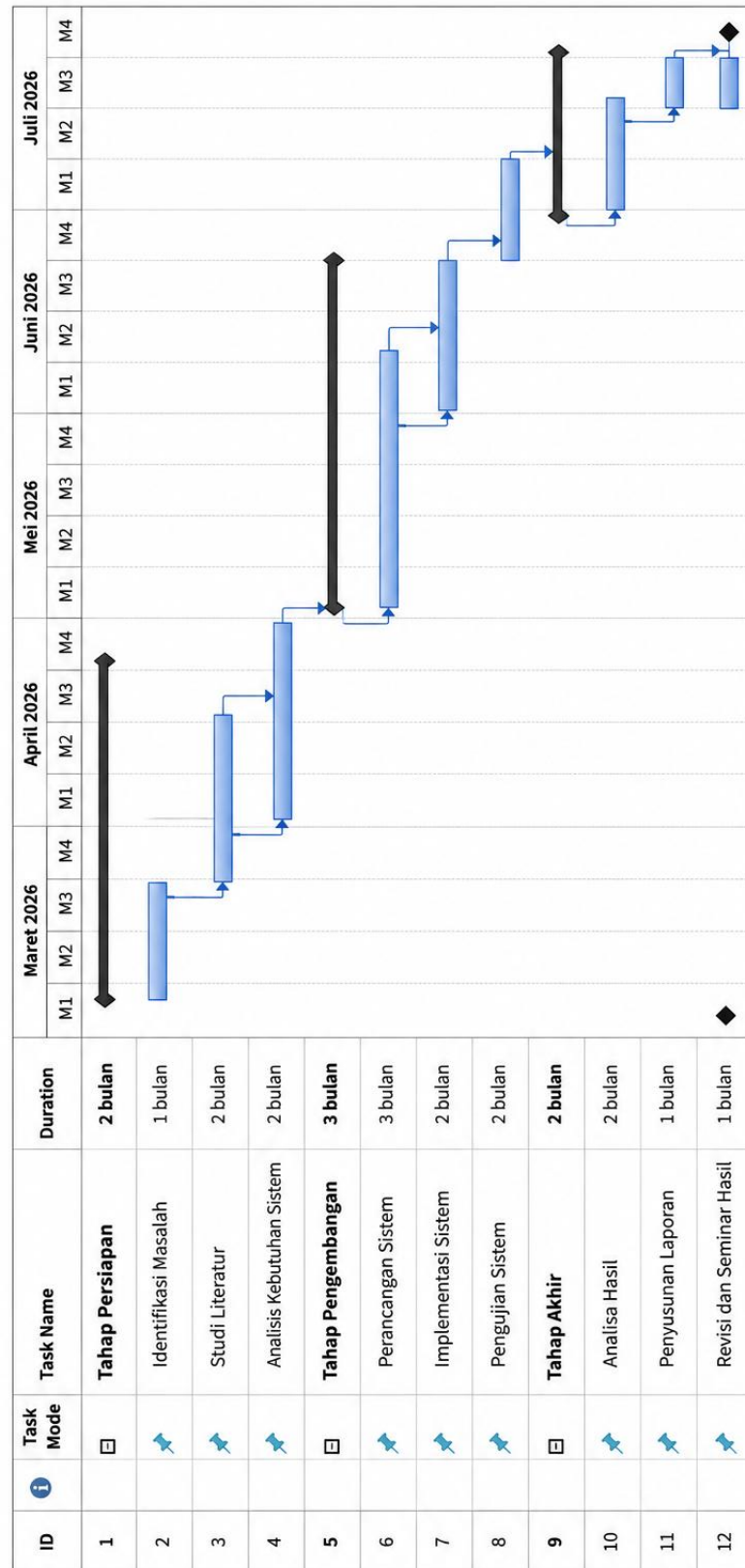
Melalui tahap analisis hasil, diharapkan dapat diketahui sejauh mana sistem yang dikembangkan mampu memenuhi kebutuhan monitoring jaringan secara *real-time* serta memberikan performa yang baik dalam proses penyajian informasi kepada pengguna.

8. Penarikan Kesimpulan

Kesimpulan merupakan tahap akhir penelitian yang dilakukan berdasarkan hasil implementasi dan pengujian sistem. Pada tahap ini akan dirumuskan kesimpulan mengenai keberhasilan perancangan dan pengembangan aplikasi *monitoring* jaringan FTTH berbasis peta digital dengan menggunakan metode *Client Side Rendering* (CSR).

Kesimpulan disusun berdasarkan hasil analisis terhadap pengujian fungsional dan pengujian performa yang telah dilakukan. Selain itu, pada tahap ini juga akan dijelaskan apakah tujuan penelitian telah tercapai sesuai dengan rumusan masalah yang ditetapkan. Hasil kesimpulan diharapkan dapat memberikan gambaran mengenai manfaat serta kontribusi sistem yang dikembangkan terhadap proses *monitoring* jaringan FTTH secara *real-time*.

C. Jadwal Penelitian



Gambar 3. 21 *Gantt Chart* Penelitian

BAB IV HASIL DAN PEMBAHASAN

A. Hasil

Penelitian ini menghasilkan aplikasi monitoring jaringan *Fiber to the Home* (FTTH) berbasis web yang dikembangkan menggunakan *framework React.js* dengan metode *Client Side Rendering* (CSR). Aplikasi menerapkan arsitektur *client-server*, di mana *frontend* dibangun menggunakan *React.js* sedangkan *backend* menggunakan *Node.js* sebagai penyedia layanan REST API dan *WebSocket*.

Frontend memanfaatkan *Protected Route* untuk mengatur hak akses pengguna, *Context API* untuk mengelola data global, serta *WebSocket* untuk menerima pembaruan data monitoring secara *real-time*. Selain itu, sistem menggunakan *Leaflet* sebagai media visualisasi topologi jaringan FTTH pada peta digital.

1. Perancangan Aplikasi Monitoring Jaringan FTTH

Penelitian ini menghasilkan sebuah aplikasi monitoring jaringan FTTH berbasis *web* yang dikembangkan menggunakan arsitektur *client-server*. Pada sisi *backend*, aplikasi dibangun menggunakan *Node.js* yang berfungsi sebagai pusat pengolahan data, penyedia layanan REST API, pengelola proses autentikasi pengguna, komunikasi dengan perangkat MikroTik melalui MikroTik API, serta pengiriman data secara *real-time* menggunakan *WebSocket*. Pada sisi *frontend*, aplikasi dikembangkan menggunakan *React.js* dengan pendekatan *Client Side Rendering* (CSR). *React.js* berfungsi untuk membangun antarmuka pengguna yang interaktif dan dinamis sehingga proses perpindahan halaman serta pembaruan data dapat dilakukan tanpa memuat ulang halaman (*reload*). Pendekatan ini memberikan pengalaman penggunaan yang lebih cepat dan responsif dibandingkan aplikasi web konvensional.

Untuk memperoleh performa sistem yang baik, aplikasi memanfaatkan *Redis* sebagai media penyimpanan sementara (*cache*). Penggunaan *Redis* memungkinkan data monitoring yang sering diakses dapat disimpan sementara sehingga proses pengambilan data menjadi lebih cepat dan beban kerja server dapat dikurangi. Komunikasi data secara *real-time* diimplementasikan menggunakan *WebSocket*. Teknologi ini memungkinkan server mengirimkan perubahan data secara langsung ke browser tanpa menunggu permintaan baru dari pengguna. Dengan demikian, perubahan status router, pengguna PPPoE, maupun perangkat FTTH dapat ditampilkan secara otomatis tanpa perlu melakukan penyegaran (*refresh*) halaman.

Seluruh informasi lokasi perangkat jaringan divisualisasikan menggunakan *Leaflet.js*, yaitu pustaka JavaScript untuk membangun peta digital interaktif. *Leaflet* digunakan untuk menampilkan posisi Router MikroTik, OLT, ODC, ODP, dan pelanggan pada peta sehingga administrator dapat memantau kondisi

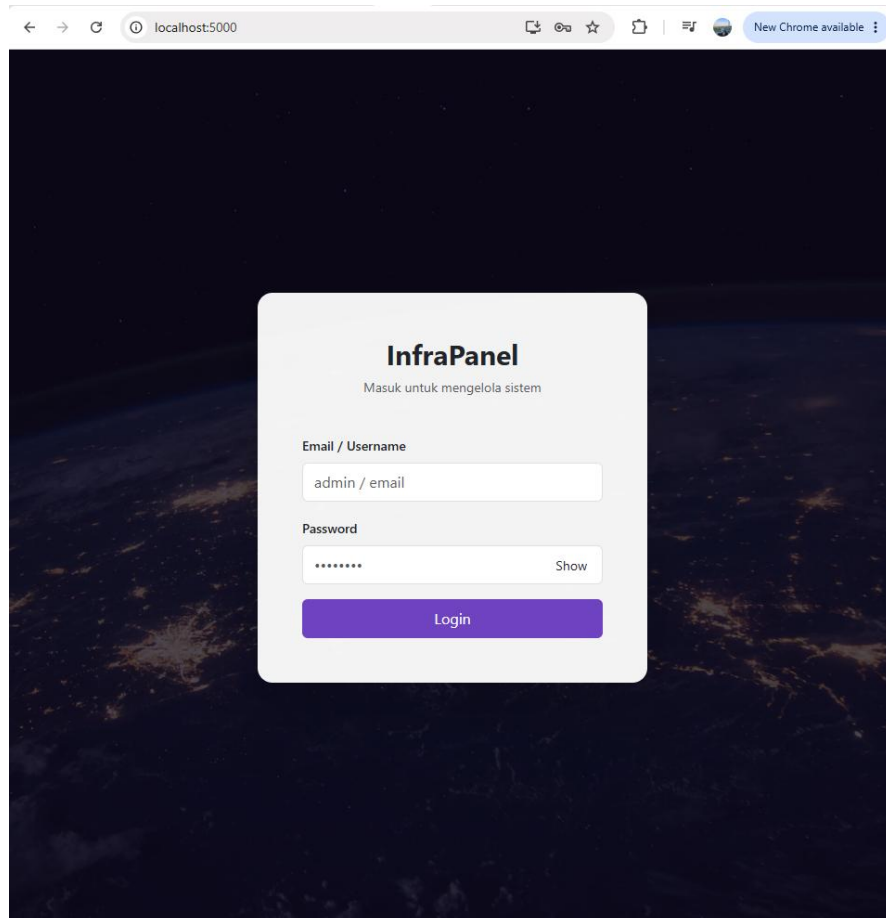
jaringan berdasarkan lokasi geografis. Data monitoring jaringan diperoleh melalui MikroTik API, yang berfungsi sebagai penghubung antara aplikasi dengan perangkat MikroTik. Melalui API tersebut, sistem dapat mengambil informasi seperti status pengguna PPPoE, penggunaan *bandwidth*, kondisi router, serta data jaringan lainnya untuk kemudian diproses dan ditampilkan pada aplikasi.

Melalui integrasi React.js, Node.js, Redis, WebSocket, Leaflet.js, dan MikroTik API, aplikasi mampu menyediakan informasi jaringan FTTH secara real-time, responsif, dan mudah dipahami. Arsitektur ini mendukung proses monitoring yang lebih efektif karena seluruh informasi jaringan dapat diakses melalui satu platform berbasis web tanpa harus menggunakan beberapa aplikasi yang berbeda..

Tabel 4. 1 Fitur Aplikasi

NO	Fitur	Deskripsi
1	Login	Melakukan autentikasi administrator sebelum mengakses sistem monitoring.
2	Dashboard Monitoring	Menampilkan ringkasan kondisi jaringan FTTH secara real-time.
3	Monitoring PPPoE	Menampilkan daftar pelanggan PPPoE beserta status koneksi, bandwidth, IP Address, dan uptime.
4	Monitoring FTTH	Menampilkan struktur jaringan OLT, ODC, ODP serta hubungan antar perangkat.
5	Manajemen Router	Mengelola perangkat MikroTik beserta konfigurasi IP, username, password, port API, dan koordinat lokasi.
6	Peta Digital	Menampilkan lokasi perangkat jaringan menggunakan Leaflet beserta visualisasi topologi jaringan.
7	Monitoring Real-Time	Melakukan pembaruan data menggunakan WebSocket tanpa refresh halaman
8	Logout	Mengakhiri sesi pengguna dengan menghapus token autentikasi dan memutus koneksi WebSocket.

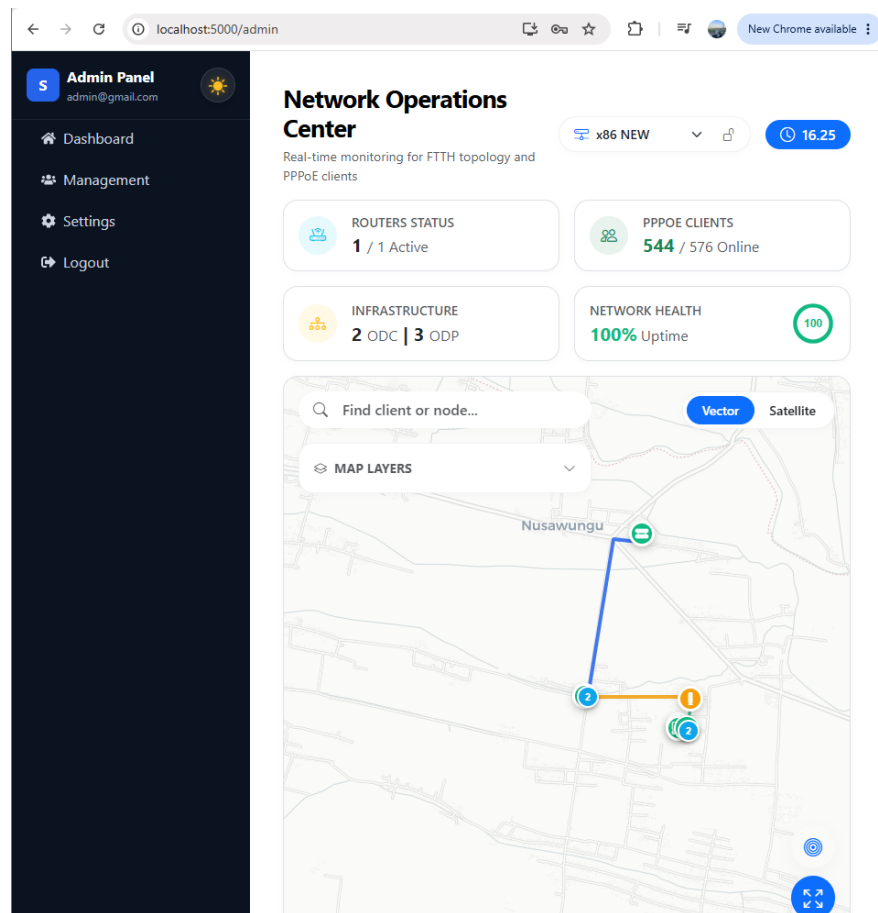
Sebagai representasi visual dari hasil pengembangan aplikasi, berikut merupakan tampilan-tampilan antarmuka dari sistem yang telah selesai diimplementasikan. Seluruh implementasi dan pengujian pada penelitian ini dilakukan menggunakan akun Administrator karena akun tersebut telah mencakup seluruh hak akses sistem. Adapun akun Teknisi menggunakan antarmuka yang sama, namun hanya memiliki hak akses untuk melihat informasi monitoring tanpa dapat melakukan perubahan data, sehingga tidak memerlukan implementasi antarmuka yang berbeda.



Gambar 4. 1 Halaman *Login*

Halaman *Login* merupakan pintu masuk aplikasi yang digunakan untuk melakukan autentikasi pengguna sebelum mengakses sistem monitoring. Berdasarkan implementasi pada file *Login.jsx*, halaman ini terdiri atas dua buah *field* yaitu *Email/Username* dan *Password* serta tombol *Login*. Sistem melakukan validasi terhadap data yang dimasukkan pengguna sebelum mengirimkan permintaan autentikasi menuju backend menggunakan REST API.

Apabila autentikasi berhasil, backend mengembalikan *access* token yang kemudian disimpan pada *browser*. Selanjutnya pengguna diarahkan menuju *Dashboard* Administrator melalui mekanisme *Protected Route* sehingga hanya pengguna yang telah terautentikasi yang dapat mengakses aplikasi.



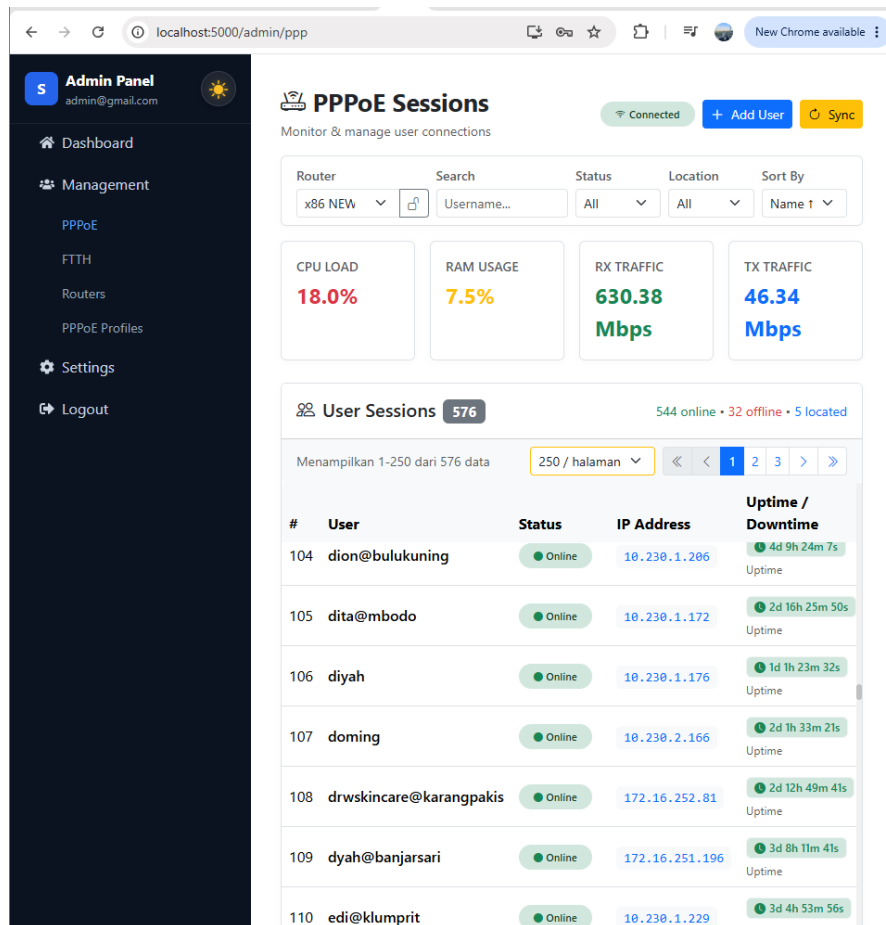
Gambar 4. 2 Dashboard Monitoring

Dashboard Monitoring merupakan halaman utama aplikasi yang dibangun menggunakan komponen *Dashboard Kpi Cards*, *Dashboard Map Components*, *Dashboard Bandwidth Chart*, *Dashboard System Logs*, dan *Live Session Duration*. *Dashboard* menampilkan beberapa indikator utama jaringan, yaitu :

- 1) Router Status
- 2) PPPoE Client
- 3) Infrastruktur FTTH
- 4) Network Health
- 5) Peta Digital

Pada saat halaman dibuka, *hook useDashboard* melakukan pengambilan data dari *backend* melalui REST API. Setelah data awal berhasil dimuat, aplikasi membuka koneksi *WebSocket* menggunakan *GlobalRealtimeContext* sehingga perubahan data monitoring dapat diperbarui secara otomatis tanpa

memerlukan refresh halaman. Implementasi tersebut menunjukkan penerapan metode *Client Side Rendering*, di mana proses pembaruan data dilakukan secara dinamis pada sisi *client*.



Gambar 4. 3 Monitoring PPPoE

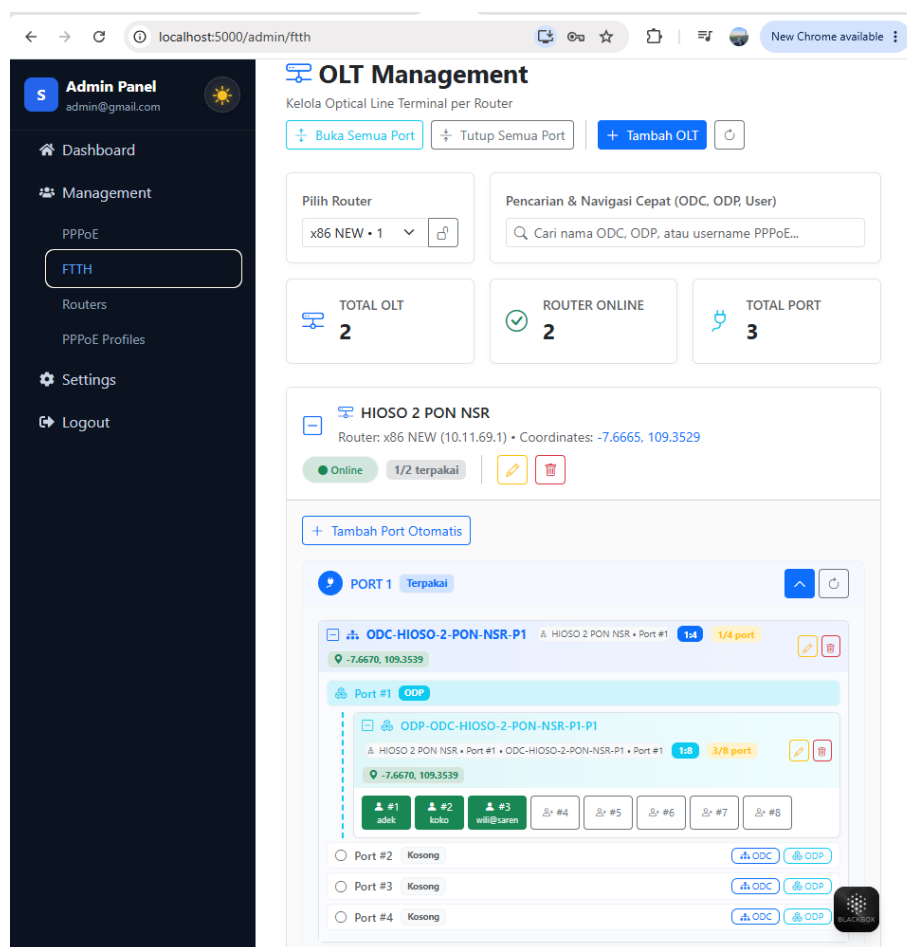
Gambar 4. 3 menunjukkan halaman Monitoring PPPoE yang digunakan untuk memantau seluruh pengguna PPPoE yang terhubung pada jaringan FTTH secara *real-time*. Halaman ini merupakan salah satu fitur utama aplikasi karena berfungsi sebagai pusat monitoring aktivitas pelanggan yang terhubung ke perangkat MikroTik.

Pada bagian atas halaman ditampilkan informasi kondisi perangkat *router* yang meliputi *CPU Load*, *RAM Usage*, *RX Traffic*, dan *TX Traffic*. Informasi tersebut diperoleh secara langsung dari perangkat MikroTik melalui *backend* menggunakan MikroTik API, kemudian dikirimkan ke *frontend* menggunakan WebSocket sehingga nilainya selalu diperbarui tanpa melakukan *refresh* halaman.

Selain informasi perangkat, sistem juga menyediakan beberapa fasilitas pencarian dan penyaringan data seperti pemilihan *router*, pencarian berdasarkan username pelanggan, penyaringan status koneksi, lokasi pelanggan, serta pengurutan data. Fasilitas tersebut bertujuan untuk

memper memudahkan administrator dalam mencari pelanggan tertentu ketika melakukan proses monitoring maupun *troubleshooting* jaringan. Pada bagian tabel, aplikasi menampilkan daftar seluruh pelanggan PPPoE yang berisi informasi *username*, status koneksi, alamat IP, serta lama koneksi (*uptime*). Status pelanggan ditampilkan menggunakan indikator warna sehingga administrator dapat dengan mudah membedakan pelanggan yang sedang *Online* maupun *Offline*.

Implementasi halaman ini memanfaatkan metode *Client Side Rendering* (CSR) sehingga perubahan status pelanggan dapat langsung ditampilkan tanpa melakukan pemuatan ulang halaman. Dengan demikian proses monitoring menjadi lebih cepat, responsif, dan efisien.



Gambar 4. 4 Monitoring FTTH

Gambar 4. 4 menunjukkan halaman Monitoring FTTH yang digunakan untuk mengelola serta memvisualisasikan struktur infrastruktur jaringan *Fiber to the Home* (FTTH). Halaman ini menampilkan hubungan antar perangkat mulai dari *Optical Line Terminal* (OLT), *Optical Distribution Cabinet* (ODC), *Optical Distribution Point* (ODP) hingga pelanggan yang terhubung pada masing-masing port.

Administrator dapat memilih router yang akan dimonitor melalui daftar router yang tersedia. Setelah router dipilih, sistem akan menampilkan seluruh perangkat OLT beserta informasi jumlah *port*, status perangkat, jumlah *port* yang telah digunakan, serta koordinat lokasi perangkat. Setiap perangkat OLT memiliki beberapa port yang dapat dikembangkan menjadi perangkat ODC dan ODP. Hubungan antar perangkat divisualisasikan secara bertingkat sehingga mempermudah administrator dalam memahami jalur distribusi jaringan *optik* dari pusat hingga pelanggan.

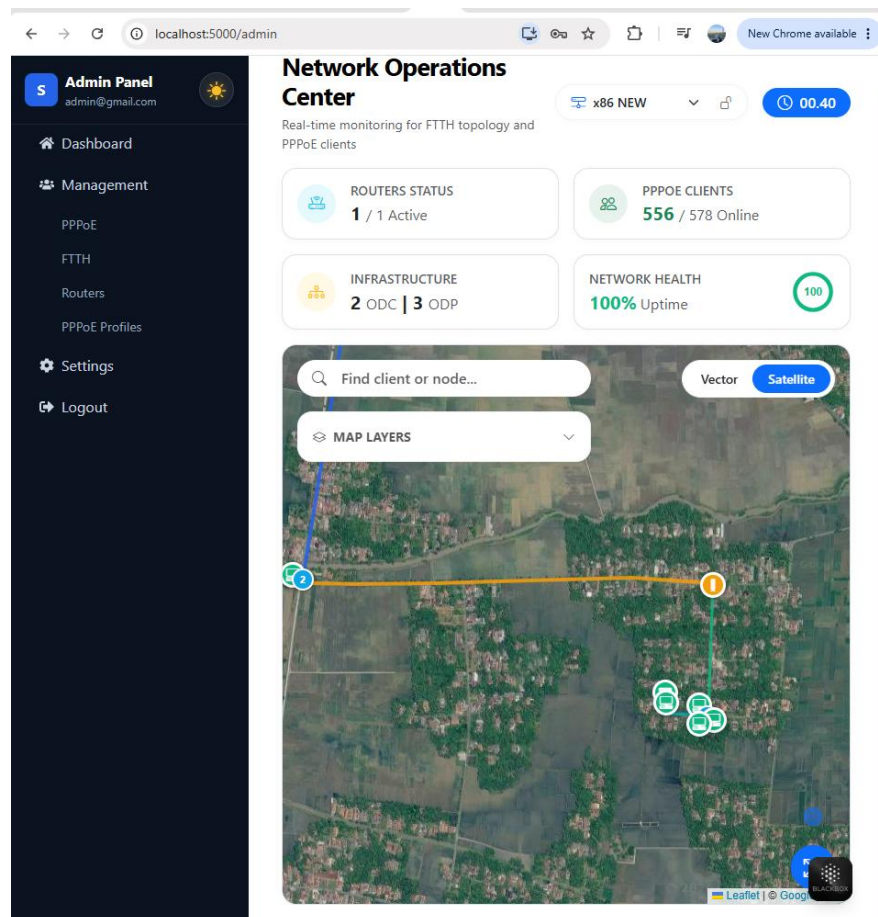
Selain menampilkan informasi monitoring, halaman ini juga menyediakan fasilitas untuk menambah, mengubah, maupun menghapus data perangkat OLT, ODC, ODP, serta port jaringan. Seluruh perubahan yang dilakukan administrator akan langsung diperbarui pada tampilan menggunakan komunikasi *WebSocket*, sehingga seluruh pengguna aplikasi memperoleh informasi terbaru secara *real-time*. Implementasi halaman Monitoring FTTH memberikan kemudahan dalam melakukan dokumentasi jaringan sekaligus membantu proses identifikasi gangguan apabila terjadi kerusakan pada salah satu titik distribusi jaringan.

Gambar 4. 5 Manajemen Router

Gambar 4. 5 menunjukkan halaman Manajemen Router MikroTik yang digunakan untuk mengelola perangkat router yang menjadi sumber utama data

monitoring jaringan. Pada halaman ini administrator dapat menambahkan perangkat router baru maupun mengubah konfigurasi router yang telah tersedia. Informasi yang dikelola meliputi nama router, alamat IP (Host/IP), *port* API, *username*, *password*, serta koordinat *geografis* berupa *latitude* dan *longitude*.

Sistem juga menyediakan fasilitas pemilihan koordinat menggunakan peta digital sehingga administrator dapat menentukan lokasi perangkat secara lebih akurat. Koordinat tersebut selanjutnya digunakan sebagai posisi marker router pada halaman peta digital. Seluruh data konfigurasi router disimpan pada basis data melalui REST API. Setelah proses penyimpanan berhasil, backend akan mengirimkan pembaruan kepada *frontend* menggunakan *WebSocket* sehingga data router pada halaman dashboard maupun peta digital langsung diperbarui tanpa memerlukan proses *refresh* halaman. Dengan adanya fitur ini, administrator dapat mengelola perangkat MikroTik secara lebih mudah sekaligus mendukung proses monitoring jaringan FTTH secara terintegrasi.



Gambar 4. 6 *Dashboard* Monitoring dan Peta Digital

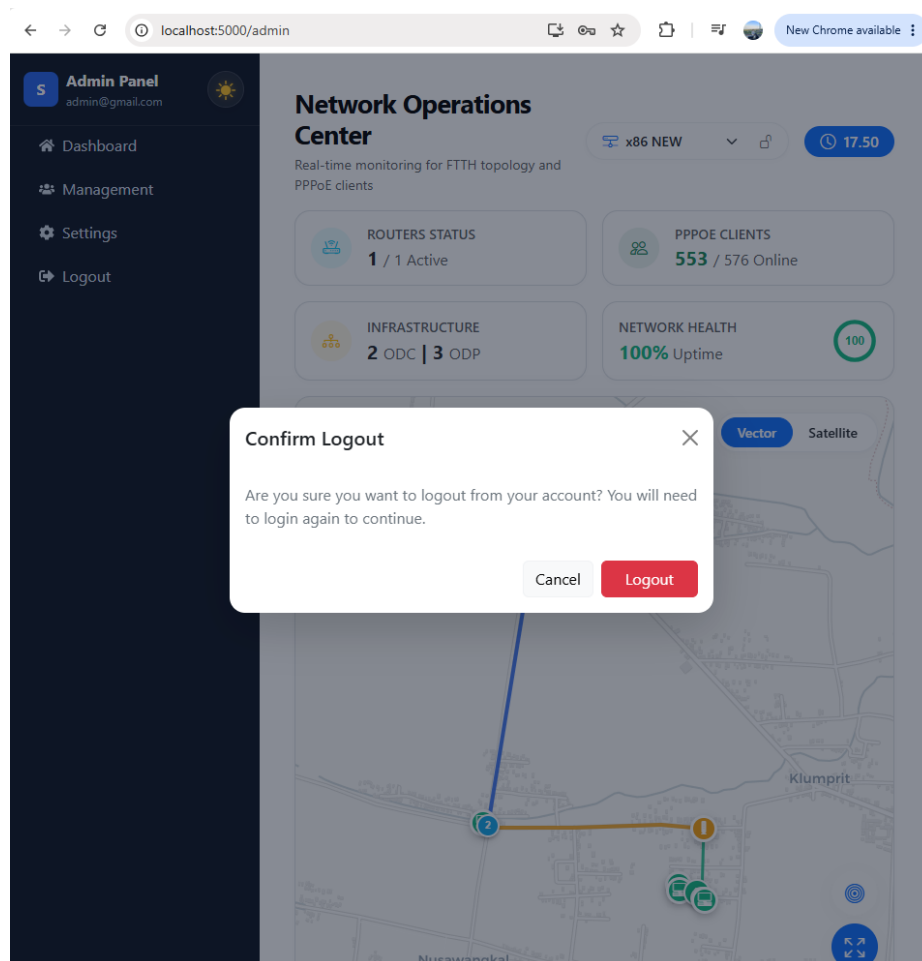
Gambar 4. 6 menunjukkan halaman *Dashboard* Monitoring yang merupakan halaman utama aplikasi setelah administrator berhasil melakukan proses autentikasi. *Dashboard* menampilkan ringkasan kondisi jaringan secara *real-time* yang meliputi status router, jumlah pelanggan PPPoE yang aktif,

jumlah perangkat ODC dan ODP, serta *Network Health*. Informasi tersebut disajikan dalam bentuk kartu informasi (*information card*) sehingga administrator dapat mengetahui kondisi jaringan secara cepat tanpa harus membuka setiap menu monitoring. Pada bagian bawah dashboard terdapat peta digital yang dibangun menggunakan *Leaflet.js*. Peta ini menampilkan lokasi perangkat jaringan beserta hubungan antar perangkat menggunakan garis koneksi (*polyline*). *Marker* yang ditampilkan terdiri atas Router MikroTik, OLT, ODC, ODP, serta pelanggan FTTH yang telah didaftarkan ke dalam sistem.

Selain menampilkan lokasi perangkat, aplikasi juga menerapkan indikator warna pada *marker* sebagai representasi kondisi pelanggan secara real-time. *Marker* berwarna hijau menunjukkan bahwa pelanggan dalam kondisi *online* atau koneksi PPPoE masih aktif sehingga layanan internet berjalan dengan normal. Sebaliknya, apabila pelanggan mengalami gangguan atau koneksi PPPoE terputus (*offline*), maka sistem akan secara otomatis mengubah *marker* menjadi berwarna merah. Perubahan warna tersebut dilakukan berdasarkan data status yang diterima dari *backend* melalui komunikasi *WebSocket*, sehingga administrator dapat segera mengetahui lokasi pelanggan yang mengalami gangguan tanpa harus melakukan pengecekan secara manual.

Penggunaan indikator warna pada *marker* memberikan kemudahan dalam proses monitoring karena administrator dapat langsung mengidentifikasi pelanggan yang bermasalah hanya dengan melihat tampilan peta digital. Selain itu, fitur ini juga membantu mempercepat proses *troubleshooting*, karena lokasi pelanggan yang mengalami gangguan dapat diketahui secara visual dan akurat. Selain fitur visualisasi status, aplikasi juga menyediakan fasilitas pencarian perangkat serta pengaturan layer peta, sehingga administrator dapat memfokuskan tampilan pada jenis perangkat tertentu sesuai kebutuhan. Seluruh perubahan status perangkat maupun pelanggan akan diperbarui secara otomatis melalui *WebSocket*, sehingga informasi yang ditampilkan pada *dashboard* dan peta digital selalu mencerminkan kondisi jaringan terkini.

Implementasi *dashboard* berbasis *Client Side Rendering* (CSR) memungkinkan seluruh komponen informasi, termasuk perubahan warna *marker*, diperbarui secara dinamis tanpa memerlukan proses *reload* halaman. Dengan demikian, aplikasi mampu memberikan pengalaman monitoring yang lebih cepat, responsif, dan interaktif dalam mendukung pengelolaan jaringan FTTH secara *real-time*.



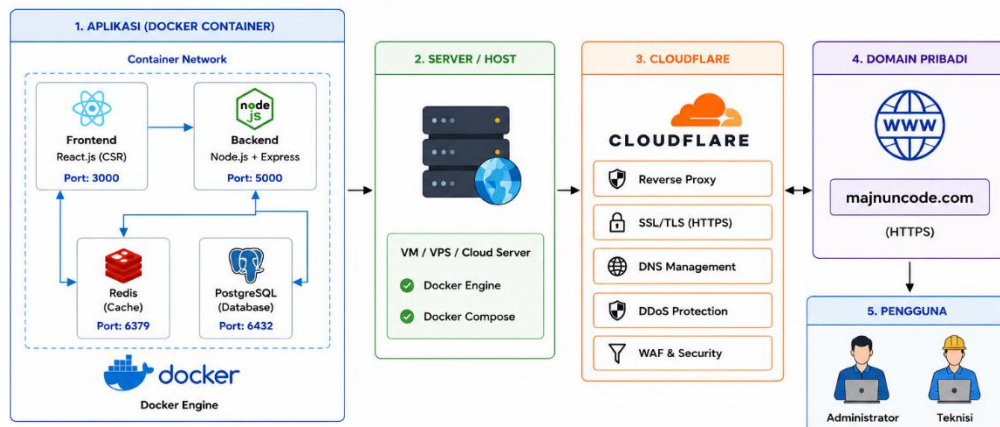
Gambar 4. 7 Logout

Gambar 4. 7 menunjukkan proses *Logout* pada aplikasi monitoring jaringan FTTH. Ketika administrator memilih menu *Logout*, sistem akan menampilkan kotak dialog konfirmasi untuk memastikan bahwa pengguna benar-benar ingin mengakhiri sesi penggunaan aplikasi. Apabila pengguna memilih tombol *Logout*, sistem akan menghapus token autentikasi yang tersimpan pada *browser*, memutus koneksi *WebSocket*, serta membersihkan data sesi pengguna. Setelah seluruh proses selesai, pengguna akan diarahkan kembali menuju halaman *Login* sehingga diperlukan proses autentikasi ulang untuk mengakses sistem.

Mekanisme tersebut bertujuan untuk menjaga keamanan aplikasi dengan memastikan bahwa setiap sesi pengguna berakhir secara benar dan tidak dapat digunakan kembali tanpa proses *login*.

Setelah proses pengembangan selesai, aplikasi monitoring jaringan FTTH dideploy menggunakan Docker agar seluruh layanan dapat berjalan dalam lingkungan yang konsisten dan terisolasi. Aplikasi terdiri atas *Frontend* React.js (*Client Side Rendering*) pada port 3000, *Backend* Node.js pada port 5000, PostgreSQL sebagai basis data pada port 6432, dan Redis sebagai media *cache* pada port 6379. Seluruh layanan dijalankan pada server menggunakan

Docker Engine dan Docker Compose. Selanjutnya, aplikasi dipublikasikan melalui Cloudflare yang berfungsi sebagai *reverse proxy*, pengelola DNS, serta penyedia layanan keamanan HTTPS, sehingga aplikasi dapat diakses secara daring menggunakan domain pribadi majnuncode.com. Dengan proses deployment tersebut, aplikasi berhasil dijalankan pada lingkungan produksi dan seluruh fitur monitoring dapat diakses oleh administrator maupun teknisi sesuai hak akses yang dimiliki.



Gambar 4. 8 Deployment Aplikasi

2. Hasil Pengujian

Setelah proses implementasi sistem selesai dilakukan, tahap selanjutnya adalah pengujian sistem. Pengujian bertujuan untuk memastikan bahwa aplikasi telah memenuhi kebutuhan pengguna serta memiliki performa yang baik ketika dioperasikan. Pada penelitian ini, pengujian dibagi menjadi dua jenis, yaitu pengujian fungsional menggunakan metode *Black Box Testing* dan pengujian non-fungsional untuk mengevaluasi performa aplikasi.

a. Hasil *Black Box Testing Login*

Tabel 4. 2 Hasil Pengujian *Login*

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Validasi form kosong	Kosongkan <i>username</i> dan <i>password</i> kemudian klik <i>Login</i>	Muncul pesan "Username dan Password wajib diisi"	Berhasil {✓} Tidak Berhasil { }
2	Username salah	Masukkan username salah dan password benar	Muncul pesan "User tidak ditemukan"	Berhasil {✓} Tidak Berhasil { }
3	Password salah	Masukkan username benar dan password salah	Muncul pesan "Password salah"	Berhasil {✓} Tidak Berhasil { }
4	Login berhasil	Masukkan username dan password yang valid	Sistem menampilkan Dashboard Monitoring	Berhasil {✓} Tidak Berhasil { }

b. Hasil *Black Box Testing Dashboard Monitoring*

Tabel 4. 3 Hasil Pengujian *Dashboard*

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Membuka Dashboard	Membuka Dashboard	Dashboard tampil	Berhasil {✓} Tidak Berhasil { }
2	Menampilkan statistik	Dashboard selesai dimuat	Jumlah Router, User PPPoE, Traffic tampil	Berhasil {✓} Tidak Berhasil { }
3	Update real-time	Backend mengirim data baru	Data dashboard berubah otomatis	Berhasil {✓} Tidak Berhasil { }

c. Hasil *Black Box Testing Monitoring PPPoE*

Tabel 4. 4 Hasil Pengujian Monitoring PPPoE

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Membuka menu PPPoE dan trafik bandwidth user	Klik menu PPPoE	Daftar user PPPoE tampil	Berhasil {✓} Tidak Berhasil { }
2	Atur lokasi pelanggan	Klik menu add location	Tempel koordinat dari Google Maps, lokasi tersimpa di database	Berhasil {✓} Tidak Berhasil { }
3	Perubahan status real-time	Status user berubah di MikroTik	Status pada aplikasi berubah otomatis	Berhasil {✓} Tidak Berhasil { }
4	Refresh data	Klik tombol Sync	Data terbaru ditampilkan	Berhasil {✓} Tidak Berhasil { }

d. Hasil *Black Box Testing Peta Digital*

Tabel 4. 5 Hasil Pengujian Peta Digital

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Membuka dashboard	Klik menu Peta	Peta digital berhasil ditampilkan	Berhasil {✓} Tidak Berhasil { }
2	Menampilkan marker perangkat dan pelanggan	Peta selesai dimuat	Marker berhasil ditampilkan	Berhasil {✓} Tidak Berhasil { }
3	Klik marker	Klik salah satu marker	Popup informasi pelanggan tampil	Berhasil {✓} Tidak Berhasil { }
4	Update marker real-time	Lokasi/status berubah	Marker diperbarui otomatis	Berhasil {✓} Tidak Berhasil { }

e. Hasil *Black Box Testing* Manajemen Router MikroTik

Tabel 4. 6 Pengujian Manajemen Router

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Menampilkan daftar router	Klik menu Router	Daftar router tampil	Berhasil {✓} Tidak Berhasil { }
2	Tambah router	Klik Add Router, Isi data router valid lalu klik Simpan, Create	Router berhasil ditambahkan	Berhasil {✓} Tidak Berhasil { }
3	Test koneksi router	Klik Test Connection	Klik Test Connection	Berhasil {✓} Tidak Berhasil { }
4	Edit router	Ubah data router	Perubahan berhasil disimpan	Berhasil {✓} Tidak Berhasil { }
5	Hapus router	Klik Hapus kemudian konfirmasi	Router berhasil dihapus	Berhasil {✓} Tidak Berhasil { }

f. Hasil *Black Box Testing* Manajemen FTTH

Tabel 4. 7 Hasil Pengujian Manajemen FTTH

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Menampilkan perangkat FTTH	Klik menu FTTH	Daftar perangkat (OLT, ODC, ODP)	Berhasil {✓} Tidak Berhasil { }
2	Tambah perangkat FTTH	Klik tambah OLT, ODC, ODP	Perangkat FTTH berhasil ditambahkan	Berhasil {✓} Tidak Berhasil { }
3	Tambah User PPPoE pada ODP	Klik port ODP kosong, tambahkan User PPPoE	User PPPoE berhasil ditambahkan	Berhasil {✓} Tidak Berhasil { }
4	Hapus perangkat FTTH	Klik Hapus kemudian konfirmasi	Perangkat berhasil dihapus	Berhasil {✓} Tidak Berhasil { }

g. Hasil *Black Box Testing Logout*

Tabel 4. 8 Hasil Pengujian *Logout*

NO	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Logout berhasil	Klik tombol Logout	Kembali ke halaman Login	Berhasil {✓} Tidak Berhasil { }
2	Session berakhir	Login kembali tanpa token	Harus melakukan login ulang	Berhasil {✓} Tidak Berhasil { }

h. Hasil Pengujian Performa

Tabel 4. 9 Hasil Pengujian Performa

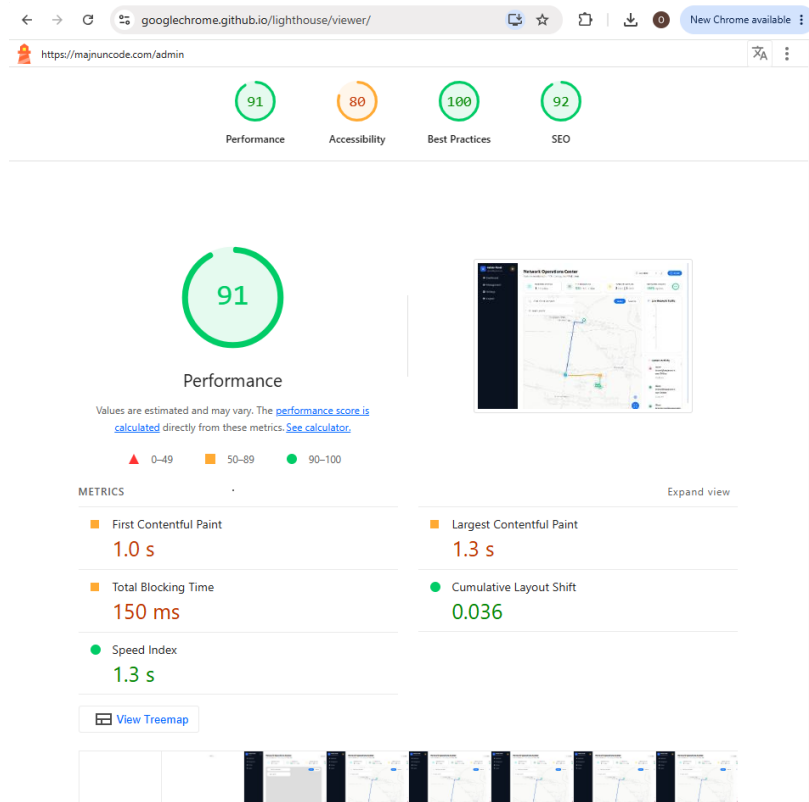
NO	Parameter	Tools	Halaman Dashboard	Halaman PPPoE
1	Performance	Google Lighthouse	91	90
2	Accessibility	Google Lighthouse	80	75
3	Best Practices	Google Lighthouse	100	100
4	SEO	Google Lighthouse	92	91
5	First Contentful Paint (FCP)	Google Lighthouse	1,0 s	1,0 s
6	Largest Contentful Paint (LCP)	Google Lighthouse	1,3 s	1,1 s
7	Speed Index	Google Lighthouse	1,3 s	1,1 s
8	Total Blocking Time (TBT)	Google Lighthouse	150 ms	60 ms
9	Cumulative Layout Shift (CLS)	Google Lighthouse	0,036	0,141

Pengujian performa pada aplikasi monitoring jaringan FTTH dilakukan menggunakan Google Lighthouse yang terintegrasi pada Google Chrome DevTools. Pengujian ini bertujuan untuk mengevaluasi kualitas aplikasi dari aspek Performance, Accessibility, Best Practices, SEO, serta metrik Core Web Vitals. Pengujian dilakukan pada dua halaman utama aplikasi, yaitu Dashboard Monitoring dan Monitoring PPPoE.

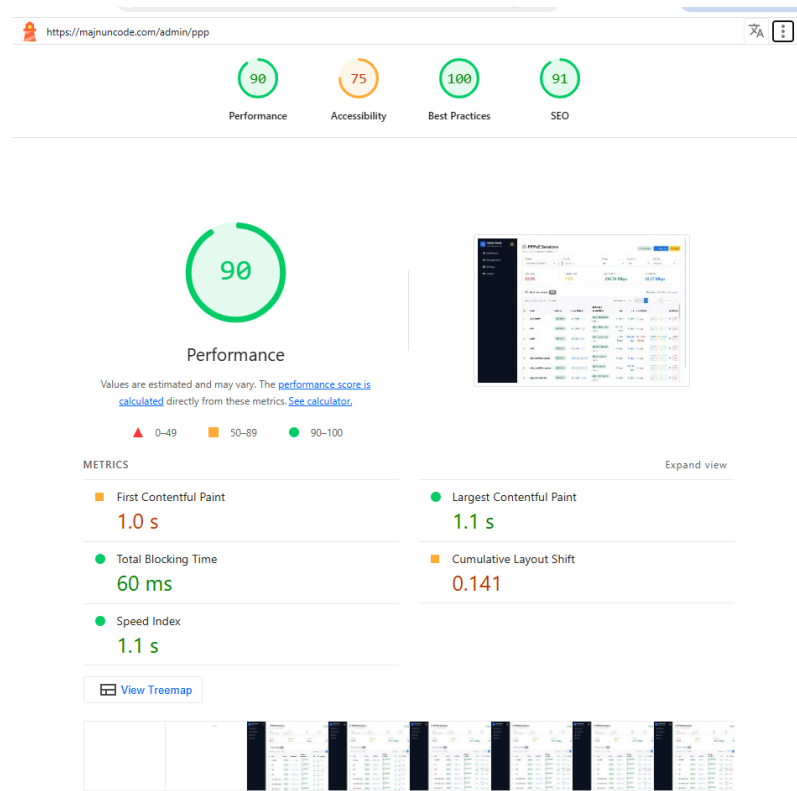
Berdasarkan hasil pengujian pada Dashboard Monitoring, aplikasi memperoleh Performance Score sebesar 91, Accessibility sebesar 80, Best Practices sebesar 100, dan SEO sebesar 92. Nilai tersebut menunjukkan bahwa aplikasi telah memiliki performa yang baik, meskipun masih terdapat beberapa aspek aksesibilitas yang dapat ditingkatkan agar lebih sesuai dengan rekomendasi Google Lighthouse. Pada halaman Monitoring PPPoE, aplikasi memperoleh Performance Score sebesar 90, Accessibility sebesar 75, Best Practices sebesar 100, dan SEO sebesar 91. Nilai tersebut menunjukkan bahwa halaman mampu memberikan performa yang baik dengan penerapan praktik pengembangan web yang sesuai standar, walaupun aspek aksesibilitas masih memerlukan penyempurnaan.

Berdasarkan parameter First Contentful Paint (FCP), kedua halaman memperoleh waktu 1,0 detik, sehingga konten pertama dapat ditampilkan dengan cepat kepada pengguna. Parameter Largest Contentful Paint (LCP) menunjukkan nilai 1,3 detik pada Dashboard dan 1,1 detik pada halaman PPPoE. Kedua nilai tersebut masih berada di bawah batas rekomendasi Google Lighthouse sebesar 2,5 detik, sehingga termasuk kategori baik. Nilai Speed Index masing-masing sebesar 1,3 detik pada Dashboard dan 1,1 detik pada halaman PPPoE menunjukkan bahwa tampilan aplikasi dapat dirender dengan cepat. Pada aspek responsivitas, Total Blocking Time (TBT) sebesar 150 ms pada Dashboard dan 60 ms pada halaman PPPoE masih berada di bawah batas rekomendasi Google Lighthouse yaitu 200 ms, sehingga interaksi pengguna tetap responsif. Parameter Cumulative Layout Shift (CLS) memperoleh nilai 0,036 pada Dashboard dan 0,141 pada halaman PPPoE. Nilai CLS pada Dashboard termasuk baik karena masih berada di bawah batas yang direkomendasikan, sedangkan nilai CLS pada halaman PPPoE sedikit melebihi batas ideal sehingga masih diperlukan optimalisasi tata letak halaman agar perpindahan elemen saat proses pemuatan dapat diminimalkan.

Secara keseluruhan, hasil pengujian menunjukkan bahwa implementasi *Client Side Rendering* (CSR) mampu menghasilkan aplikasi monitoring jaringan FTTH yang memiliki performa baik, waktu muat cepat, responsif terhadap interaksi pengguna, serta mampu menyajikan informasi monitoring secara real-time.



Gambar 4. 9 Hasil Page Dashboard Google Lighthouse



Gambar 4. 10 Hasil Page PPPoE Google Lighthouse

B. Pembahasan

1. Sistem Monitoring Jaringan FTTH Berbasis *Client Side Rendering*

Berdasarkan hasil implementasi, aplikasi berhasil mengintegrasikan monitoring PPPoE, bandwidth, router, dan peta digital ke dalam satu aplikasi. Aplikasi monitoring jaringan FTTH berhasil dikembangkan menggunakan arsitektur client-server dengan memanfaatkan beberapa teknologi utama, yaitu React.js sebagai frontend, Node.js sebagai backend, Redis sebagai media cache, WebSocket sebagai komunikasi data real-time, Leaflet.js sebagai visualisasi peta digital, serta MikroTik API sebagai sumber data monitoring jaringan. Penerapan teknologi tersebut bertujuan untuk menghasilkan aplikasi monitoring yang mampu menampilkan kondisi jaringan FTTH secara cepat, responsif, dan real-time sehingga administrator dapat melakukan pemantauan jaringan secara lebih efektif.

a. Integrasi *React.js*, *Node.js*, *Redis*, *WebSocket*, dan *Leaflet*

Aplikasi monitoring jaringan FTTH dibangun menggunakan lima komponen utama yang saling terintegrasi. *React.js* berfungsi sebagai *frontend* yang menerapkan metode *Client Side Rendering* (CSR). Seluruh proses *rendering* antarmuka dilakukan pada *browser* sehingga perpindahan halaman berlangsung lebih cepat tanpa harus melakukan pemuatan ulang (*reload*) seluruh halaman. *Node.js* berperan sebagai *backend* yang menangani autentikasi pengguna, komunikasi dengan perangkat MikroTik melalui API MikroTik, pengelolaan data FTTH, serta penyedia layanan REST API dan *WebSocket*. *Redis* digunakan sebagai media penyimpanan sementara (*cache*) untuk mempercepat proses distribusi data monitoring kepada *frontend* sehingga mengurangi beban akses langsung ke basis data. *WebSocket* berfungsi sebagai media komunikasi dua arah antara *backend* dan *frontend*. Setiap perubahan status pelanggan PPPoE maupun perangkat jaringan langsung dikirim ke browser administrator tanpa harus menunggu permintaan (*request*) baru dari *client*. Selanjutnya, *Leaflet.js* digunakan untuk menampilkan visualisasi topologi jaringan pada peta digital. Melalui *Leaflet*, lokasi Router MikroTik, OLT, ODC, ODP, serta pelanggan dapat divisualisasikan sehingga administrator dapat mengetahui kondisi jaringan secara geografis. Integrasi kelima teknologi tersebut menghasilkan sistem monitoring yang mampu menampilkan informasi jaringan secara *real-time*, cepat, dan interaktif.

Tabel 4. 10 Komponen dan Teknologi Sistem

NO	Komponen	Tekologi	Peran Utama
1	Frontend	React.js (CSR)	Menampilkan antarmuka, melakukan rendering halaman, menerima pembaruan data secara real-time.
2	Backend	Node.js	Mengelola autentikasi, API, komunikasi MikroTik, dan WebSocket.
3	Cache (Memory)	Redis	Menyimpan data sementara agar proses monitoring lebih cepat.
4	Real-Time	WebSocket	Mengirim perubahan data dari server ke client secara langsung.
5	Visualisasi (Peta Digital)	Leaflet.js	Menampilkan peta digital beserta topologi jaringan FTTH.
6	Data Source	MikroTik API	Menyediakan data router, PPPoE, bandwidth, dan status jaringan.

b. Implementasi Mekanisme *Client Side Rendering* (CSR)

Implementasi metode *Client Side Rendering* (CSR) dimulai ketika administrator berhasil melakukan proses autentikasi. Setelah *login* berhasil, *browser* hanya memuat struktur utama aplikasi satu kali. Selanjutnya seluruh proses perpindahan halaman dilakukan oleh *React Router* tanpa melakukan *reload* halaman. Pada saat administrator membuka menu *Dashboard*, Monitoring PPPoE, Router, maupun FTTH, *frontend* akan mengirim permintaan data kepada backend melalui REST API. Setelah data diterima, *React* melakukan proses rendering hanya pada komponen yang mengalami perubahan.

Pendekatan ini membuat proses navigasi menjadi lebih cepat dibandingkan metode rendering konvensional karena *browser* tidak perlu memuat ulang seluruh halaman setiap kali pengguna berpindah menu. Selain itu, ketika terjadi perubahan status pelanggan maupun perangkat jaringan, *WebSocket* akan mengirimkan data terbaru kepada *frontend* sehingga *React* hanya memperbarui komponen yang berubah tanpa mengganggu komponen lainnya.

c. Implementasi Monitoring *Real-Time*

Komunikasi data secara *real-time* merupakan salah satu bagian utama pada penelitian ini. *Backend* secara terus-menerus melakukan proses sinkronisasi terhadap data yang berasal dari perangkat MikroTik. Ketika terjadi perubahan status pelanggan PPPoE, perubahan *bandwidth*, maupun perubahan kondisi perangkat FTTH, backend akan mengirimkan informasi tersebut kepada seluruh client yang sedang aktif menggunakan *WebSocket*. *Frontend* kemudian menerima data tersebut dan memperbarui tampilan

Dashboard Monitoring, *Monitoring PPPoE*, maupun *Peta Digital* secara otomatis. Dengan mekanisme tersebut administrator dapat mengetahui perubahan kondisi jaringan dalam waktu yang sangat singkat tanpa harus melakukan proses *refresh* halaman.

d. Implementasi Visualisasi Peta Digital

Visualisasi jaringan menggunakan *Leaflet.js* menjadi salah satu pembeda utama penelitian ini dibandingkan sistem monitoring berbasis tabel. Peta digital menampilkan seluruh perangkat jaringan yang terdiri atas Router MikroTik, OLT, ODC, ODP, serta pelanggan FTTH dalam bentuk marker yang saling terhubung menggunakan garis (*polyline*). Selain menampilkan posisi perangkat, aplikasi juga menerapkan indikator warna sebagai representasi kondisi pelanggan. *Marker* hijau menunjukkan pelanggan berada dalam kondisi *online*. *Marker* merah menunjukkan pelanggan berada dalam kondisi *offline* atau koneksi PPPoE terputus. Perubahan warna marker dilakukan secara otomatis berdasarkan status pelanggan yang diperoleh dari backend melalui *WebSocket*. Dengan demikian administrator dapat langsung mengetahui lokasi pelanggan yang mengalami gangguan hanya dengan melihat peta digital. Implementasi visualisasi ini mempercepat proses identifikasi gangguan serta membantu teknisi menentukan lokasi yang memerlukan penanganan.

e. Analisis Hasil Pengujian

Berdasarkan hasil pengujian fungsional menggunakan metode *Black Box Testing*, seluruh fitur aplikasi berhasil berjalan sesuai dengan kebutuhan sistem. Proses *autentikasi*, *Dashboard Monitoring*, monitoring PPPoE, manajemen perangkat FTTH, manajemen router, visualisasi peta digital, serta mekanisme *logout* menghasilkan keluaran yang sesuai dengan skenario pengujian.

Berdasarkan hasil pengujian performa yang telah dipaparkan pada subbab sebelumnya, selanjutnya dilakukan analisis untuk mengetahui tingkat kualitas performa aplikasi monitoring jaringan FTTH yang dikembangkan. Analisis dilakukan dengan membandingkan hasil pengujian terhadap standar kualitas yang ditetapkan oleh *Google Lighthouse*. Perbandingan tersebut bertujuan untuk mengetahui apakah penerapan metode *Client Side Rendering* (CSR) mampu menghasilkan aplikasi yang memiliki performa, responsivitas, dan stabilitas sesuai dengan standar yang direkomendasikan. Standar penilaian *Google Lighthouse* yang digunakan sebagai acuan analisis disajikan pada Tabel.

Tabel 4. 11 Analisis *Performance Google Lighthouse Dashboard*

NO	Parameter	Hasil Aplikasi	Standar Google	Kategori
1	Performance Score	91	≥ 90	Baik
2	Accessibility	80	50-89	Cukup
3	Best Practices	100	≥ 90	Baik
4	SEO	92	≥ 90	Baik
5	First Contentful Paint (FCP)	1,0 s	1,6 – 0,95 s	Cukup
6	Largest Contentful Paint (LCP)	1,3 s	2,41 – 1,22 s	Cukup
7	Speed Index (SI)	1,3 s	$\leq 1,33$ s	Baik
8	Total Blocking Time (TBT)	150 ms	≤ 160 ms	Baik
9	Cumulative Layout Shift (CLS)	0,036	$\leq 0,11$	Baik

Berdasarkan hasil pengujian Google Lighthouse pada halaman Dashboard, aplikasi memperoleh Performance Score sebesar 91 sehingga termasuk kategori baik. Nilai Accessibility sebesar 80 berada pada kategori cukup, yang menunjukkan bahwa aplikasi masih memerlukan beberapa penyempurnaan pada aspek aksesibilitas antarmuka. Sementara itu, Best Practices memperoleh nilai 100 dan SEO sebesar 92, sehingga keduanya termasuk kategori baik. Dari aspek performa pemuatan halaman, First Contentful Paint (FCP) sebesar 1,0 detik menunjukkan bahwa konten awal berhasil ditampilkan dengan cepat. Nilai Largest Contentful Paint (LCP) sebesar 1,3 detik dan Speed Index sebesar 1,3 detik menunjukkan bahwa halaman mampu dirender dalam waktu yang relatif singkat. Selain itu, Total Blocking Time (TBT) sebesar 150 ms masih berada di bawah batas rekomendasi Google sehingga interaksi pengguna tetap responsif. Nilai Cumulative Layout Shift (CLS) sebesar 0,036 menunjukkan bahwa tata letak halaman stabil selama proses pemuatan.

Tabel 4. 12 Analisis *Performance Google Lighthouse PPPoE*

NO	Parameter	Hasil Aplikasi	Standar Google	Kategori
1	Performance Score	90	≥ 90	Baik
2	Accessibility	75	50-89	Cukup
3	Best Practices	100	≥ 90	Baik
4	SEO	91	≥ 90	Baik
5	First Contentful Paint (FCP)	1,0 s	1,6 – 0,95 s	Cukup
6	Largest Contentful Paint (LCP)	1,1 s	$\leq 1,2$ s	Baik
7	Speed Index (SI)	1,1 s	$\leq 1,33$ s	Baik
8	Total Blocking Time (TBT)	60 ms	≤ 160 ms	Baik

9	Cumulative Layout Shift (CLS)	0,141	0,25 - 0,11	Cukup
---	-------------------------------	-------	-------------	-------

Berdasarkan hasil pengujian Google Lighthouse pada halaman Monitoring PPPoE, aplikasi memperoleh Performance Score sebesar 90 sehingga termasuk kategori baik. Nilai Accessibility sebesar 75 berada pada kategori cukup, sedangkan Best Practices sebesar 100 dan SEO sebesar 91 termasuk kategori baik. Pada metrik performa, First Contentful Paint (FCP) sebesar 1,0 detik menunjukkan bahwa halaman mampu menampilkan konten awal dengan cepat. Nilai Largest Contentful Paint (LCP) sebesar 1,1 detik, Speed Index sebesar 1,1 detik, serta Total Blocking Time (TBT) sebesar 60 ms menunjukkan bahwa halaman memiliki performa yang responsif. Sementara itu, nilai Cumulative Layout Shift (CLS) sebesar 0,141 berada pada kategori perlu perbaikan (Needs Improvement). Hal ini menunjukkan masih terdapat sedikit pergeseran tata letak saat halaman dimuat sehingga optimasi layout masih dapat dilakukan pada pengembangan selanjutnya.

Hasil pengujian menunjukkan bahwa penerapan *Client Side Rendering* (CSR) mampu memberikan performa aplikasi yang baik dengan nilai Performance sebesar 91 pada halaman Dashboard dan 90 pada halaman Monitoring PPPoE. Hasil ini mendukung penelitian Khasanah dan Suartana (2024) yang menyatakan bahwa CSR mampu menghasilkan antarmuka yang interaktif dan responsif pada aplikasi web. Selain itu, integrasi WebSocket memungkinkan pembaruan data dilakukan secara real-time tanpa memuat ulang halaman, sehingga sesuai dengan karakteristik sistem monitoring jaringan yang membutuhkan penyajian informasi secara cepat dan dinamis.

2. Analisis Keunggulan Sistem

Berdasarkan hasil implementasi dan pengujian, aplikasi monitoring jaringan FTTH yang dikembangkan memiliki beberapa keunggulan. Integrasi WebSocket memungkinkan perubahan status jaringan ditampilkan secara real-time tanpa memerlukan proses refresh halaman. Selain itu, visualisasi berbasis peta digital mempermudah administrator dalam mengetahui lokasi perangkat dan pelanggan secara geografis. Penerapan *Client Side Rendering* (CSR) juga mampu meningkatkan kecepatan navigasi aplikasi karena proses rendering dilakukan di sisi klien. Seluruh informasi jaringan FTTH, mulai dari Router MikroTik, OLT, ODC, ODP, hingga pelanggan, berhasil diintegrasikan ke dalam satu platform monitoring sehingga proses pengelolaan jaringan menjadi lebih efektif dan efisien.

3. Analisis Keterbatasan Sistem

Meskipun aplikasi berhasil memenuhi tujuan penelitian, masih terdapat beberapa keterbatasan. Lokasi pelanggan dan perangkat masih ditentukan secara manual melalui koordinat yang dimasukkan oleh administrator. Selain itu, sistem hanya mendukung perangkat MikroTik RouterOS sebagai sumber data monitoring dan belum mendukung vendor lain. Aplikasi juga masih tersedia dalam versi web sehingga belum dapat dimanfaatkan melalui perangkat mobile. Di samping itu, parameter monitoring masih berfokus pada status perangkat, pelanggan PPPoE, dan topologi FTTH, sehingga belum mencakup analisis kualitas sinyal optik secara mendalam.

Selain keterbatasan pada implementasi sistem, penelitian ini juga memiliki beberapa keterbatasan dari aspek evaluasi. Pengujian sistem masih berfokus pada pengujian fungsional menggunakan *Black Box Testing* dan pengujian performa aplikasi menggunakan Google Lighthouse, sehingga belum dilakukan uji penerimaan pengguna (*user acceptance testing*) secara langsung yang melibatkan administrator maupun teknisi jaringan sebagai pengguna akhir untuk menilai aspek kemudahan penggunaan, efektivitas, dan kepuasan pengguna. Selain itu, penelitian ini belum melakukan pengukuran delay komunikasi data secara kuantitatif pada mekanisme real-time berbasis WebSocket, seperti pengukuran latensi pembaruan data dari server hingga ditampilkan pada antarmuka aplikasi. Oleh karena itu, efektivitas sistem dalam menyajikan data secara real-time masih dievaluasi berdasarkan indikator performa aplikasi dan keberhasilan pembaruan data, belum berdasarkan nilai latensi yang terukur secara empiris.

BAB V KESIMPULAN

A. Kesimpulan

Berdasarkan hasil penelitian, implementasi sistem, pengujian fungsional, pengujian performa, serta pembahasan yang telah dilakukan mengenai Penggunaan Metode *Client Side Rendering* (CSR) pada Aplikasi Monitoring Jaringan FTTH Secara *Real-Time* Berbasis Peta Digital, maka dapat ditarik kesimpulan sebagai berikut.

1. Implementasi Aplikasi Monitoring FTTH Berbasis CSR

Perancangan aplikasi monitoring jaringan FTTH berbasis peta digital menggunakan metode *Client Side Rendering* (CSR) berhasil diimplementasikan. Aplikasi dikembangkan menggunakan React.js sebagai frontend dengan pendekatan *Client Side Rendering* (CSR), Node.js sebagai backend, WebSocket untuk komunikasi data secara real-time, Redis sebagai media cache, Leaflet.js sebagai visualisasi peta digital, serta MikroTik API sebagai sumber data monitoring jaringan. Sistem mampu mengintegrasikan informasi pengguna PPPoE, perangkat jaringan, penggunaan bandwidth, serta lokasi pelanggan ke dalam satu platform berbasis web sehingga mempermudah administrator dan teknisi dalam melakukan pemantauan jaringan secara real-time.

2. Hasil Pengujian Fungsional Dan Performa Aplikasi

Hasil pengujian menunjukkan bahwa aplikasi yang dikembangkan telah berfungsi dengan baik dan memiliki performa yang baik. Pengujian fungsional menggunakan metode *Black Box Testing* menunjukkan seluruh fitur aplikasi berjalan sesuai dengan kebutuhan pengguna. Sementara itu, pengujian performa menggunakan Google Lighthouse menghasilkan nilai Performance sebesar 91 pada halaman Dashboard dan 90 pada halaman Monitoring PPPoE, Accessibility sebesar 80 dan 75, Best Practices sebesar 100 pada kedua halaman, serta SEO sebesar 92 dan 91. Selain itu, nilai First Contentful Paint (FCP) sebesar 1,0 detik, Largest Contentful Paint (LCP) sebesar 1,3 detik dan 1,1 detik, Speed Index sebesar 1,3 detik dan 1,1 detik, Total Blocking Time (TBT) sebesar 150 ms dan 60 ms, serta Cumulative Layout Shift (CLS) sebesar 0,036 dan 0,141 menunjukkan bahwa implementasi Client Side Rendering (CSR) mampu menghasilkan aplikasi monitoring yang responsif, efisien, dan mendukung penyajian informasi jaringan FTTH secara real-time.

B. Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan sebagai bahan pengembangan pada penelitian selanjutnya, yaitu:

1. Sistem monitoring dapat dikembangkan agar mendukung perangkat jaringan dari berbagai vendor selain MikroTik sehingga dapat diterapkan pada lingkungan jaringan yang lebih beragam.
2. Pengembangan sistem dapat mencakup parameter kualitas jaringan FTTH yang lebih lengkap, seperti pengukuran daya optik (*Rx Power* dan *Tx Power*), nilai redaman (*optical loss*), serta indikator kualitas layanan jaringan lainnya sehingga informasi monitoring menjadi lebih komprehensif.
3. Sistem dapat dilengkapi dengan fitur notifikasi otomatis melalui media seperti Telegram, WhatsApp, atau email ketika terjadi gangguan jaringan, sehingga administrator dapat melakukan penanganan lebih cepat tanpa harus melakukan pemantauan secara terus-menerus.
4. Pengembangan aplikasi dalam bentuk mobile application berbasis Android atau iOS dapat dilakukan agar administrator maupun teknisi dapat melakukan monitoring jaringan secara lebih fleksibel melalui perangkat bergerak.
5. Dari sisi penelitian, penelitian selanjutnya disarankan melakukan uji pengalaman pengguna (User Experience/User Acceptance Testing) yang melibatkan administrator dan teknisi jaringan sebagai pengguna akhir. Pengujian tersebut dapat digunakan untuk mengevaluasi aspek kemudahan penggunaan, efektivitas, efisiensi, dan tingkat kepuasan pengguna terhadap aplikasi yang dikembangkan.
6. Penelitian selanjutnya juga disarankan melakukan pengukuran performa komunikasi real-time secara kuantitatif, seperti mengukur delay (latensi) pembaruan data menggunakan WebSocket dari server hingga informasi ditampilkan pada antarmuka aplikasi. Pengukuran tersebut dapat memberikan bukti empiris mengenai kemampuan sistem dalam menyajikan data secara real-time serta menjadi dasar evaluasi untuk meningkatkan kualitas performa aplikasi.

DAFTAR PUSTAKA

- [1] <https://Survei.Apjii.Or.Id>, “Survei Internet APJII 2025,” <https://Survei.Apjii.Or.Id/Home>.
- [2] M. Y. Ibrahim And N. Nawaningtyas, “Evaluasi Kinerja Jaringan FTTH Berbasis ODC Dan GPON Berdasarkan Power Link Budget Di RT 015 Tanjung Duren Utara,” *Sudo Jurnal Teknik Informatika*, Vol. 4, No. 2, Pp. 112–123, Jun. 2025, Doi: 10.56211/Sudo.V4i2.945.
- [3] M. U. Khasanah And I. M. Suartana, “Perbandingan Performa Client Side Rendering Pada Implementasi Static Website (Studi Kasus Barokah Rentcar),” *Journal Of Informatics And Computer Science (JINACS)*, Vol. 6, No. 02, Pp. 541–550, Jul. 2024, Doi: 10.26740/Jinacs.V6n02.P541-550.
- [4] R. Ardiyanto And E. Ardianto, “Analisa Performasi Metode Client Side Rendering, Server Side Rendering, Dan Incremental Static Regeneration Dalam Proses Website Rendering,” *Computer Science (CO-SCIENCE)*, Vol. 4, No. 1, Pp. 19–27, Jan. 2024, Doi: 10.31294/Coscience.V4i1.2427.
- [5] W. R. F. Author And A. W. Author, “PENGEMBANGAN SISTEM VISUALISASI TOPOLOGI JARINGAN OLT BERBASIS WEBSITE DENGAN FITUR MONITORING TRAFFIC DAN DETEKSI PERANGKAT,” *Jurnal Informatika Dan Teknik Elektro Terapan*, Vol. 13, No. 3S1, Oct. 2025, Doi: 10.23960/Jitet.V13i3s1.7575.
- [6] Vindi Eka Safitri, Iqsyahiro Kresna A, And Cahyo Prihantoro, “Penerapan Network Monitoring Menggunakan The Dude Mikrotik Dan Notifikasi Pesan Dengan Aplikasi Telegram, Whatsapp, Dan Email,” *Decode: Jurnal Pendidikan Teknologi Informasi*, Vol. 4, No. 1, Pp. 94–106, Jan. 2024, Doi: 10.51454/Decode.V4i1.200.
- [7] M. Jaelani, R. N. Whidhiasih, And R. T. Handayanto, “Implementasi Sistem Monitoring Jaringan Mikrotik Dengan The Dude,” *Journal Of Students’ Research In Computer Science*, Vol. 6, No. 1, Pp. 83–98, May 2025, Doi: 10.31599/15w1se86.
- [8] A. Yuhaneef, A. A. Asril, And D. O. Yanti, “Perancangan Dan Analisis Kinerja Jaringan Fiber To The Home (FTTH) Dengan Teknologi Gigabit Passive Optical Network (GPON) Menggunakan Software Optisystem,” *JTEIN: Jurnal Teknik Elektro Indonesia*, Vol. 4, No. 2, Aug. 2023, Doi: 10.24036/Jtein.V4i2.407.
- [9] Z. Abdellaoui, Y. Dieudonne, And A. Aleya, “Design, Implementation And Evaluation Of A Fiber To The Home (FTTH) Access Network Based On A Giga Passive Optical Network GPON,” *Array*, Vol. 10, P. 100058, Jul. 2021, Doi: 10.1016/J.Array.2021.100058.

- [10] R. Ramadhan And E. Sandi, “Analisis Perubahan Jaringan Telekomunikasi Central Toka LIM4 Wilayah Jatinegara Dari Saluran Tembaga Menjadi Fiber Optik,” *JURNAL PENDIDIKAN VOKASIONAL TEKNIK ELEKTRONIKA (Jvote)*, Vol. 4, No. 2, Pp. 34–41, Dec. 2021, Doi: 10.21009/Jvote.V4i2.39453.
- [11] M. R. Alwi, I. Kustiawan, And N. F. Arief Hakim, “MODEL DAN IMPLEMENTASI JARINGAN FIBER TO THE HOME DI DESA ARJASARI BERBASIS ETHERNET PASSIVE OPTICAL NETWORK (EPON),” *TECHNO-SOCIO EKONOMIKA*, Vol. 16, No. 2, P. 174, Oct. 2023, Doi: 10.32897/Techno.2023.16.2.2839.
- [12] A. Izzaturrahmah, M. Tahir, N. Hariri, And H. U. A. Putra, “Implementasi Dan Evaluasi Monitoring Pada Jaringan Lokal Berbasis Mikrotik Menggunakan Visualisasi Graphing Dan Graph,” *Digital Transformation Technology*, Vol. 5, No. 1, Pp. 143–149, May 2025, Doi: 10.47709/Digitech.V5i1.5883.
- [13] . B. And Mi’rajul Rifqi, “IMPLEMENTASI DAN PERBANDINGAN MONITORING JARINGAN BERBASIS SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP) MENGGUNAKAN CACTI DAN MUNIN DI SMK NEGERI 1 PEKANBARU,” *Zonasi: Jurnal Sistem Informasi*, Vol. 1, No. 2, Pp. 58–74, Sep. 2019, Doi: 10.31849/Zn.V1i2.3249.
- [14] M. Sari And D. W. Sari, “Sistem Pemantauan Jaringan Dengan Protokol SNMP Pada Stasiun Gas Talang Duku Menggunakan Zabbix.”
- [15] “\Cite{Rendering Sisi Server (SSR)2025},” Google.
- [16] Rendering Sisi Server (SSR),
“https://developers.google.com/solutions/content-driven/hosting/rendering?utm_source=chatgpt.com&hl=id,” Google.
- [17] W. R. F. Author And A. W. Author, “PENGEMBANGAN SISTEM VISUALISASI TOPOLOGI JARINGAN OLT BERBASIS WEBSITE DENGAN FITUR MONITORING TRAFFIC DAN DETEKSI PERANGKAT,” *Jurnal Informatika Dan Teknik Elektro Terapan*, Vol. 13, No. 3S1, Oct. 2025, Doi: 10.23960/Jitet.V13i3s1.7575.
- [18] @Chromiumdev Pada X, “Rendering Di Web.”
- [19] P. Karampakakis, D. Ioakeimidou, P. Chatzimisios, And K. A. Tsintotas, “A Web-Based Application For Smart City Data Analysis And Visualization,” *Future Internet*, Vol. 17, No. 5, P. 217, May 2025, Doi: 10.3390/Fi17050217.
- [20] A. Anton And H. Supendar, “≪B≫Implementasi Sistem Monitoring Jaringan Real-Time Berbasis Open Source Dengan Integrasi Zabbix Dan

Telegram≪/B≫,” *Jurnal Infortech*, Vol. 7, No. 1, Pp. 56–63, Jun. 2025, Doi: 10.31294/Infortech.V7i1.12365.

- [21] W. Baydowi And H. Al Islami, “SISTEM MONITORING JARINGAN BERBASIS WEB DENGAN FITUR NOTIFIKASI INSIDEN OTOMATIS DAN PENCATATAN LOG HISTORIS PADA PT. MITRA BERSAMA JAYA,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, Vol. 10, No. 1, Pp. 641–645, Jan. 2026, Doi: 10.36040/Jati.V10i1.16840.
- [22] M. I. Fahren And K. Penulis, “Peran Kartografi Dalam Menyajikan Informasi Geografis,” *Jurnal Pendidikan Dan Kewarganegara Indonesia*, Vol. 2, No. 2, Pp. 207–213, Doi: 10.61132/Jupenkei.V2i2.403.
- [23] M. Devega, Y. Yuhelmi, And W. Walhidayat, “PERANCANGAN PETA TIGA DIMENSI (3D) UNIVERSITAS LANCANG KUNING,” *Zonasi: Jurnal Sistem Informasi*, Vol. 2, No. 1, Pp. 48–60, Jan. 2021, Doi: 10.31849/Zn.V2i1.4590.
- [24] M. Enggar Santika, E. Yovita Dwi Utami, B. Murtianta, And U. Kristen Satya Wacana, “Analisis Perencanaan Jaringan Akses Fiber-To-The-Home Berdasarkan Teknologi Gigabit Passive Optical Network (GPON) Di STO Banyumanik Semarang.”
- [25] W. R. Fatahillah And D. A. Wulandari, “Perancangan Visualisasi Topologi Jaringan Akses FTTH Pada Optical Line Terminal (OLT) Dengan Library Vis-Network”.
- [26] Y. J. Sulaeman, “Program Studi D3 Teknik Elektronika Politeknik Gajah Tunggal Implementasi Network Monitoring Dan Notifikasi Sistem Di PT XYZ Menggunakan Zabbix,” 2022. [Online]. Available: <https://Jurnal.Poltek-Gt.Ac.Id/Index.Php/Jiti/>

LAMPIRAN

A. Surat Observasi



PT WIFIAN SOLUTION
Kavling Candra Mustika Blok B No.18
Mustikasari, Mustikajaya, Kota Bekasi 17157
Email: noc@wifian.id

SURAT KETERANGAN
Nomor : 2.1/WSNSW/VII/2026

Yang bertanda tangan di bawah ini:

Nama : Iis Waryanto

Jabatan : Pimpinan Operasional PT WIFIAN SOLUTION Cab. Nusawungu

Alamat : Jl. Jayusman, Nusawungu, Kabupaten Cilacap, Jawa Tengah

Dengan ini menerangkan bahwa:

Nama : Ahmad Mustaufiq

NIM : 22EO10006

Program Studi : Informatika

Fakultas : Matematika dan Ilmu Komputer

Perguruan Tinggi : Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap

Benar telah melaksanakan kegiatan **observasi dan pengumpulan data penelitian** di **PT WIFIAN SOLUTION** dalam rangka penyusunan skripsi dengan judul:

"Penggunaan Metode Client Side Rendering (CSR) pada Aplikasi Monitoring Jaringan FTTH Secara Real-Time Berbasis Peta Digital."

Kegiatan observasi dilaksanakan pada periode **Maret 2026 sampai dengan Juli 2026**. Selama kegiatan tersebut, yang bersangkutan melakukan observasi terhadap proses monitoring jaringan FTTH, pengelolaan perangkat MikroTik, monitoring pengguna PPPoE, pengumpulan kebutuhan sistem, serta mempelajari infrastruktur jaringan yang digunakan sebagai bahan penelitian akademik.

Dengan diterbitkannya surat keterangan ini, kami menyatakan bahwa yang bersangkutan benar telah melaksanakan observasi di lingkungan PT WIFIAN SOLUTION. Surat keterangan ini dibuat untuk digunakan sebagai salah satu persyaratan penyusunan dan penyelesaian skripsi pada Program Studi Informatika, Fakultas Matematika dan Ilmu Komputer, Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap.

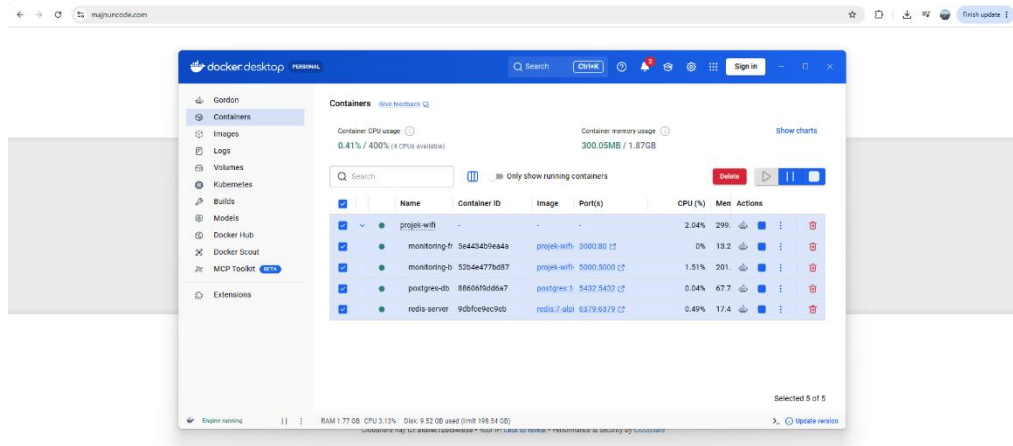
Demikian surat keterangan ini dibuat dengan sebenarnya agar dapat dipergunakan sebagaimana mestinya.

Cilacap, 25 Juli 2026

Hormat kami,


Iis Waryanto
WIFIAN SOLUTION NSW

B. Lampiran *Deployment* Aplikasi



C. Lampiran Code Program

<https://github.com/wifioplon-stack/ahmadmustaufiqwifiansolution>

