

Aplikasi Booking Management IP Address Berbasis Web dengan Integrasi REST API dan Role-Based Access Control (RBAC)

Web-Based IP Address Booking Management Application with REST API Integration and RoleBased Access Control (RBAC)

Agis Buhori
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

agisbuhori@student.telkomuniversity.ac.id

Prajna Deshanta Ibnugraha
Fakultas Ilmu Terapan
Universitas Telkom
Bandung Indonesia

prajna@telkomuniversity.ac.id

Devie Ryana Suchendra
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

Deviersuchendra@telkomuniversity.ac.id

Pengelolaan IP Address menjadi tantangan krusial bagi perusahaan dengan infrastruktur jaringan berskala besar, terutama ketika proses pencatatan dan distribusinya masih bergantung pada metode manual seperti spreadsheet. Kondisi ini rentan memicu konflik alamat IP, duplikasi data, serta ketidakmampuan sistem dalam menyajikan informasi penggunaan IP secara real-time kepada administrator jaringan. Penelitian ini merancang dan membangun Aplikasi Booking Management IP Address berbasis web yang mengintegrasikan REST API serta menerapkan mekanisme Role-Based Access Control (RBAC) untuk mengatur hak akses pengguna secara terstruktur. Teknologi yang digunakan meliputi React JS pada sisi front-end, Node.js dengan Express.js pada sisi back-end, dan MySQL sebagai basis data. Sistem juga dihubungkan dengan platform monitoring jaringan LibreNMS melalui REST API guna mendukung validasi status IP secara otomatis. Hasil pengujian kinerja menggunakan Google PageSpeed Insights mencatat skor performa 93, best practices 100, dan SEO 92. Seluruh fitur utama aplikasi berhasil diverifikasi melalui pengujian fungsionalitas, membuktikan bahwa sistem ini layak diimplementasikan dalam lingkungan operasional Perusahaan.

Kata Kunci—IP Address Management; Aplikasi Berbasis Web; Booking Management; REST API; Role-Based Access Control (RBAC); LibreNMS; React JS; Node.js

I. PENDAHULUAN

Dalam industri telekomunikasi modern, pengelolaan alamat IP (Internet Protocol) merupakan komponen penting yang menentukan efisiensi, keamanan, dan stabilitas operasional jaringan. Bagi perusahaan penyedia infrastruktur berskala besar seperti PT Mora Telematika Indonesia Tbk (Moratelindo), ribuan alamat IP digunakan untuk kebutuhan layanan internal maupun eksternal sehingga proses pengelolaannya harus dilakukan secara terpusat, sistematis, dan memiliki tingkat akurasi tinggi.

Namun, pada kondisi saat ini sebagian proses pencatatan dan pendistribusian IP Address masih dilakukan secara manual melalui spreadsheet dan komunikasi informal antarstaf teknis. Pendekatan manual ini rentan terhadap kesalahan pencatatan, duplikasi data, serta tidak mampu menyediakan informasi penggunaan IP secara real-time sehingga menimbulkan berbagai kendala operasional.

Permasalahan yang muncul dari metode manual tersebut mencakup sering terjadinya konflik IP (IP conflict) yang berpotensi menyebabkan gangguan layanan, hilangnya histori peminjaman IP, serta kesulitan dalam menelusuri penggunaan IP pada perangkat atau pengguna tertentu. Selain itu, tidak adanya pembatasan hak akses pada proses pengelolaan IP menimbulkan risiko keamanan karena semua pengguna memiliki akses yang sama terhadap data sensitif jaringan. Sistem manual juga tidak terintegrasi dengan platform monitoring jaringan seperti LibreNMS, sehingga status perangkat atau IP Address tidak dapat divalidasi secara otomatis.

Penelitian ini mengusulkan pembangunan Aplikasi Booking Management IP Address yang dilengkapi dengan Role-Based Access Control (RBAC) untuk mengatur hak akses pengguna, integrasi LibreNMS melalui REST API untuk memvalidasi status IP secara real-time, serta penerapan HTTPS guna memastikan keamanan komunikasi data. Dengan adanya sistem ini, pengelolaan IP Address di PT Moratelindo diharapkan menjadi lebih efektif, efisien, aman, dan mampu menyediakan histori pemakaian yang akurat untuk kebutuhan pemantauan serta audit teknis perusahaan.

II. TINJAUAN PUSTAKA

Pengelolaan alamat IP (Internet Protocol) merupakan salah satu komponen penting dalam manajemen jaringan modern. Setiap perangkat jaringan memerlukan alamat IP yang unik agar dapat saling berkomunikasi melalui protokol internet. Pada perusahaan berskala besar, khususnya penyedia layanan telekomunikasi dan Internet Service Provider (ISP), pengelolaan IP Address menjadi semakin kompleks seiring meningkatnya jumlah perangkat layanan dan kebutuhan jaringan. Tanpa sistem pengelolaan yang baik, administrator jaringan sering menghadapi konflik IP, kesalahan alokasi, duplikasi pencatatan, serta pemborosan ruang alamat jaringan.

Dalam praktiknya, masih banyak organisasi yang menggunakan metode manual seperti spreadsheet dalam mencatat alokasi IP Address. Metode ini sangat rentan menimbulkan human error, kesalahan pencatatan, duplikasi data, serta konflik IP yang menyebabkan terputusnya layanan jaringan [1]. Permasalahan tersebut mendorong organisasi untuk beralih ke sistem otomatisasi IPAM berbasis web yang mampu menyediakan

pencatatan terpusat, monitoring status IP, serta integrasi dengan perangkat jaringan. Salah satu perangkat lunak IPAM yang banyak digunakan adalah phpIPAM, aplikasi open-source yang dilengkapi dengan fitur subnet kalkulator, manajemen IPv4/IPv6, integrasi DHCP/DNS, hingga REST API untuk koneksi ke sistem lain [2].

Selain phpIPAM, penelitian oleh Alzakwani et al. [3] menekankan pentingnya sistem otomatisasi dalam proses subnetting dan alokasi IP. Sistem otomatis tersebut mampu membagi blok IP secara lebih akurat, mengurangi kesalahan kalkulasi, serta mempercepat proses perencanaan kebutuhan jaringan.

Selain aspek pengelolaan, keamanan akses menjadi faktor yang tidak dapat dipisahkan dalam pengembangan aplikasi jaringan. Banyak penelitian mengadopsi Role-Based Access Control (RBAC) sebagai mekanisme pembatasan hak akses. RBAC memberikan struktur akses yang jelas dan terkontrol, di mana setiap pengguna hanya dapat mengakses fitur yang sesuai dengan peran atau tanggung jawabnya. Sistem berdasarkan RBAC terbukti efektif dalam mencegah akses tidak sah dan mengurangi risiko kebocoran data pada aplikasi jaringan [4].

Integrasi sistem juga menjadi kebutuhan penting dalam manajemen jaringan modern. Salah satu teknologi yang banyak digunakan adalah REST API, sebuah metode pertukaran data yang sederhana dan fleksibel. Sistem monitoring seperti LibreNMS menyediakan REST API yang memungkinkan aplikasi eksternal mengakses data perangkat dan status IP Address secara real-time [5]. Dari sisi keamanan komunikasi data, penggunaan protokol HTTPS (SSL/TLS) menjadi standar utama dalam pengembangan aplikasi berbasis web, mencegah penyadapan, modifikasi data, dan serangan man-in-the-middle.

III. DASAR TEORI

A. React JS

React JS adalah sebuah library JavaScript yang dikembangkan oleh Facebook (Meta) untuk membangun antarmuka pengguna (User Interface) berbasis web. React menggunakan pendekatan component-based architecture, di mana setiap bagian tampilan dibangun dalam bentuk komponen yang dapat digunakan kembali. Keunggulan React JS antara lain penggunaan Virtual DOM yang membuat proses rendering lebih cepat dan efisien, reusable components untuk mengurangi duplikasi kode, serta unidirectional data flow yang memudahkan pengelolaan alur data. Dalam aplikasi Booking Management IP Address, React JS digunakan sebagai front-end untuk menampilkan dashboard, data IP Address, status ketersediaan IP, serta form booking secara dinamis.

B. Node.js

Node.js adalah runtime environment JavaScript yang memungkinkan JavaScript dijalankan di sisi server. Node.js dikembangkan oleh Ryan Dahl dan dibangun di atas mesin JavaScript Google V8. Node.js dikenal memiliki performa tinggi karena menggunakan model asynchronous dan non-blocking I/O. Dalam aplikasi Booking Management IP Address, Node.js digunakan sebagai back-end untuk menangani permintaan dari front-end React JS, mengelola logika aplikasi, melakukan validasi data, serta berinteraksi dengan database MySQL.

C. MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang bersifat open-source dan menggunakan bahasa Structured Query Language (SQL). Dalam aplikasi Booking Management IP Address, MySQL digunakan untuk menyimpan data pengguna, data IP Address, status ketersediaan IP, serta riwayat pemesanan IP. Dengan struktur tabel yang terorganisasi, MySQL mendukung pengelolaan data secara konsisten dan memudahkan proses pencarian maupun pelaporan data.

D. RBAC (Role-Based Access Control)

Role-Based Access Control (RBAC) merupakan metode pengaturan hak akses pengguna berdasarkan peran (role) yang dimilikinya. Setiap peran memiliki hak akses tertentu terhadap fitur dan data dalam sistem. Penerapan RBAC bertujuan untuk meningkatkan keamanan serta memudahkan pengelolaan akses pengguna. Dalam aplikasi Booking Management IP Address, RBAC digunakan untuk membedakan hak akses antara Admin, Teknisi, dan User. Admin memiliki hak akses penuh terhadap sistem, Teknisi memiliki akses terbatas pada pengelolaan IP, sedangkan User hanya dapat melakukan booking dan melihat status IP.

E. REST API

REST API (Representational State Transfer Application Programming Interface) merupakan metode komunikasi antara front-end dan back-end menggunakan protokol HTTP dengan format data JSON. REST API memungkinkan pertukaran data secara terstruktur dan fleksibel. Pada aplikasi Booking Management IP Address, REST API berfungsi sebagai penghubung antara React JS dan back-end Node.js serta database MySQL. Melalui REST API, proses pengambilan data, penyimpanan, validasi, dan pembaruan data dapat dilakukan secara cepat dan terintegrasi.

F. LibreNMS

LibreNMS merupakan salah satu aplikasi Network Monitoring System (NMS) berbasis open-source yang digunakan untuk memantau kondisi dan status perangkat jaringan secara real-time. Aplikasi ini dapat digunakan untuk memonitor berbagai perangkat jaringan seperti router, switch, server, maupun perangkat jaringan lainnya yang terhubung dalam suatu infrastruktur jaringan. LibreNMS menyediakan REST API yang dapat digunakan aplikasi eksternal untuk mengambil data perangkat secara langsung.

IV. ARSITEKTUR DAN PERANCANGAN SISTEM

A. Gambaran Sistem

Arsitektur sistem pada aplikasi Booking Management IP Address ini terdiri dari beberapa lapisan yang saling terintegrasi untuk mendukung proses autentikasi, pengelolaan role, hingga penyajian data kepada pengguna. Setelah pengguna memasukkan kredensial, sistem akan melakukan proses autentikasi dan menentukan peran (role) pengguna, yaitu Admin, Teknisi, atau User, sesuai dengan data yang tersimpan dalam database. Setiap role memiliki hak akses yang berbeda dan dibatasi oleh mekanisme Role-Based Access Control (RBAC).

Setelah role diverifikasi, frontend React.js menampilkan dashboard sesuai peran pengguna. Admin mendapatkan akses ke Admin Dashboard untuk mengelola seluruh data IP Address dan pengguna. Teknisi

diarahkan ke Teknisi Dashboard untuk memproses dan memperbarui status IP Address, sedangkan user umum hanya dapat mengakses User Dashboard untuk melakukan booking IP dan memantau status pemesanan mereka.

Backend Node.js kemudian memproses permintaan tersebut, melakukan validasi hak akses, dan terhubung ke sistem penyimpanan data yaitu MySQL. Hasil proses tersebut dikembalikan ke frontend dalam format JSON, sehingga aplikasi dapat menampilkan data terbaru secara akurat kepada pengguna.

B. Blok Diagram Sistem

Blok diagram sistem terdiri dari tiga bagian utama, yaitu Input, Proses, dan Output. Pada bagian Input, data dimasukkan oleh pengguna ke dalam sistem melalui antarmuka web, meliputi data login pengguna, peran (role) pengguna, data IP Address, serta data pemesanan atau pelepasan IP Address. Bagian Proses mencakup autentikasi dan validasi pengguna, pengolahan data melalui REST API, penerapan Role-Based Access Control (RBAC), serta penyimpanan dan pengambilan data dari database MySQL. Bagian Output menghasilkan informasi status IP Address, riwayat booking, dan data monitoring perangkat.

C. Entity-Relationship Diagram (ERD)

Entity-Relationship Diagram (ERD) menggambarkan struktur data dalam sistem dengan empat entitas utama yang saling terhubung:

- Entitas Admin: memiliki otoritas penuh mengelola data master sistem. Memiliki hubungan one-to-many dengan entitas IP Address.
- Entitas User: merepresentasikan pengguna atau teknisi yang akan meminjam IP Address. Memiliki hubungan one-to-many dengan entitas Booking.
- Entitas IP Address: menyimpan semua data alamat IP yang ada. Memiliki hubungan one-to-many dengan entitas Booking.
- Entitas Booking: berfungsi sebagai tabel histori yang mencatat setiap transaksi peminjaman dan pelepasan IP, menyimpan informasi tanggal peminjaman, durasi, dan status.

D. Use Case Diagram

Use case diagram menggambarkan interaksi antara tiga aktor utama dalam sistem IP Booking Management, yaitu Admin, Teknisi, dan User, yang masing-masing memiliki hak akses berbeda sesuai penerapan RBAC. Admin berperan sebagai pengelola utama sistem dengan kewenangan melakukan manajemen user, manajemen RBAC, menyetujui atau menolak booking IP, melihat laporan sistem, serta memantau seluruh data booking IP. Teknisi berfokus pada aspek operasional dengan melihat data booking IP serta melakukan manajemen IP. Sementara itu, User memiliki akses terbatas untuk melakukan booking IP serta melihat data IP Pool yang tersedia.

E. Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini berfungsi untuk mendukung proses pengembangan, pengujian, dan pengoperasian aplikasi. Berikut adalah kebutuhan perangkat lunak yang digunakan:

TABEL 1

Kebutuhan Perangkat Lunak

Nama Perangkat	Deskripsi
Visual Studio Code	Code editor untuk penulisan dan pengelolaan kode program

Nama Perangkat	Deskripsi
React JS	Library JavaScript untuk membangun antarmuka pengguna (front-end)
Node.js	Runtime environment JavaScript untuk menjalankan kode di sisi server (back-end)
MySQL/PostgreSQL	Sistem manajemen basis data relasional untuk menyimpan data aplikasi
Postman	Tools pengujian REST API (GET, POST, PUT, DELETE)
LibreNMS	Sistem monitoring jaringan open-source terintegrasi melalui REST API
Laragon	Platform pengembangan lokal untuk menjalankan server aplikasi
Browser Chrome	Peramban web untuk pengujian tampilan dan debugging

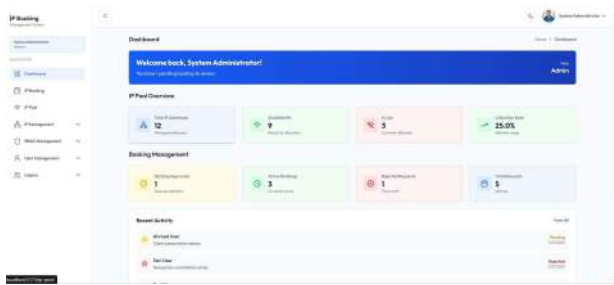
V. IMPLEMENTASI

Pada tahap implementasi aplikasi Booking Management IP Address berbasis web, pengembangan halaman antarmuka (frontend) menggunakan framework React JS, yang dipilih karena kemampuannya dalam membangun tampilan interaktif, modular, dan mudah dikembangkan. Untuk sisi backend, sistem dibangun menggunakan Node.js dengan Express.js sebagai REST API server. Seluruh data disimpan dalam database MySQL yang terstruktur untuk memudahkan pengelolaan data booking, data pengguna, dan data IP address. Integrasi antara frontend dan backend dilakukan melalui komunikasi REST API berbasis protokol HTTPS, sehingga data yang dikirimkan lebih aman dan mendukung proses otentikasi serta otorisasi berbasis Role-Based Access Control (RBAC).

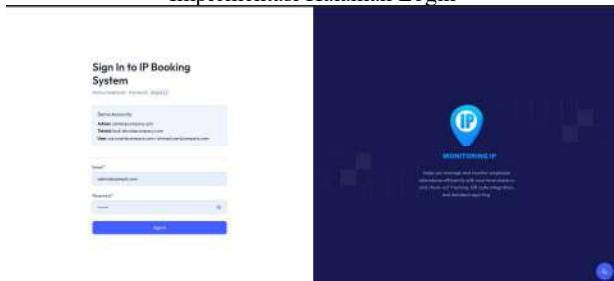
A. Implementasi Halaman Website

Pada tahap implementasi halaman website, seluruh antarmuka pengguna (user interface) dikembangkan menggunakan React JS yang memungkinkan pembuatan halaman yang dinamis, interaktif, serta mudah dipelihara. Setiap halaman pada aplikasi dirancang menggunakan pendekatan component-based, sehingga setiap fungsi seperti formulir booking, tabel data, notifikasi, dan menu navigasi dikelola dalam komponen terpisah. Tampilan website dibuat dengan memperhatikan aspek kemudahan penggunaan (user friendly), konsistensi desain, dan responsivitas agar dapat diakses melalui berbagai perangkat.

Secara keseluruhan, implementasi halaman website mencakup halaman login dan autentikasi, halaman dashboard yang menampilkan ringkasan data booking dan status IP address, halaman booking untuk melakukan pemesanan IP address, halaman approval khusus admin dan teknisi, halaman manajemen IP address serta pengguna, dan halaman RBAC Dashboard.



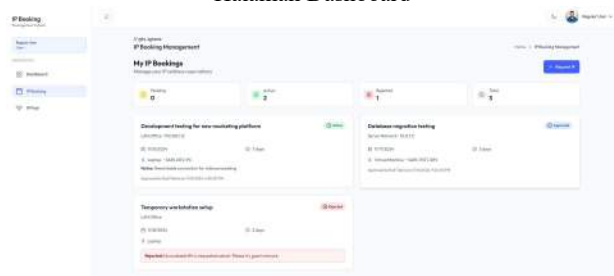
GAMBAR 1
Implementasi Halaman Login



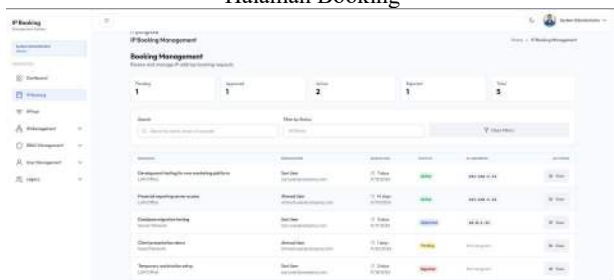
GAMBAR 2
Halaman Login & Autentikasi



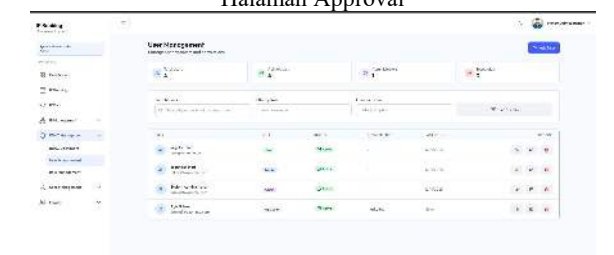
GAMBAR 3
Halaman Dashboard



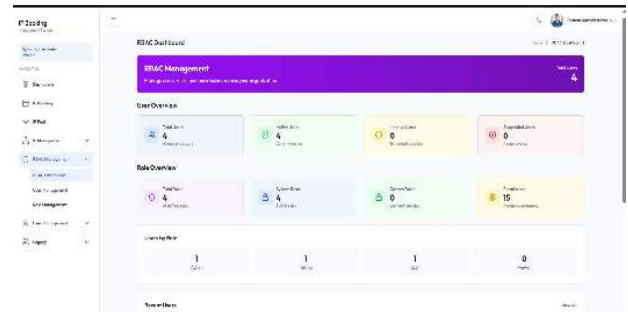
GAMBAR 4
Halaman Booking



GAMBAR 5
Halaman Approval



GAMBAR 6
Halaman Manajemen IP Address



GAMBAR 7
Halaman RBAC Dashboard

Setiap halaman tersebut telah berhasil diuji dan berfungsi sesuai tujuan, yaitu memudahkan proses booking IP address serta memastikan seluruh aktivitas dalam aplikasi memiliki alur yang jelas dan terstruktur.

B. Integrasi REST API

Integrasi REST API pada aplikasi Booking Management IP Address berperan sebagai jembatan utama antara sisi frontend dan backend. Seluruh proses pengambilan data (fetching), penyimpanan data, pembaruan, dan penghapusan dilakukan melalui endpoint API yang disediakan oleh server Node.js. REST API dibangun mengikuti struktur arsitektur RESTful, yang berarti setiap endpoint memiliki fungsi spesifik sesuai metode HTTP yang digunakan.

API autentikasi digunakan untuk proses login dan validasi akses pengguna. Jika valid, server membuat JSON Web Token (JWT) berisi data user dan role. Token dikembalikan ke frontend untuk digunakan pada request berikutnya. Token ini wajib disertakan oleh frontend pada header setiap permintaan API lainnya.

```
const { data: response, status } = await post("auth/login", formData)
console.log('response', response)
```

GAMBAR 8
Code API Autentikasi



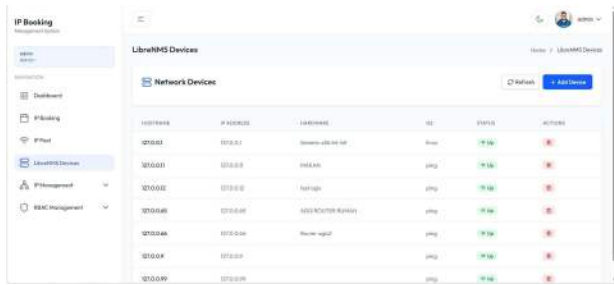
GAMBAR 9

Contoh Respon API Login Berhasil (Token JWT)

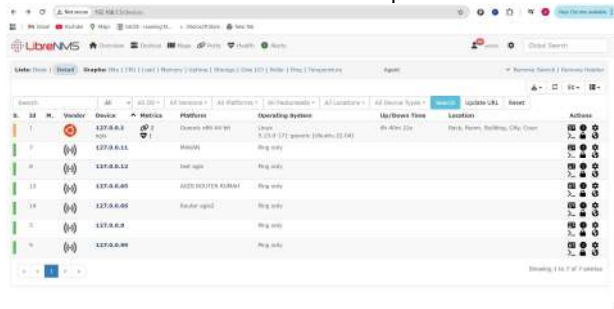
C. Integrasi LibreNMS

Pada tahap implementasi sistem, aplikasi Booking Management IP Address juga diintegrasikan dengan sistem monitoring jaringan menggunakan LibreNMS. Integrasi ini bertujuan untuk menampilkan informasi perangkat jaringan yang sedang dimonitor sehingga administrator jaringan dapat melihat status perangkat secara langsung melalui aplikasi.

Backend Node.js mengakses API LibreNMS menggunakan endpoint /api/v0/devices untuk mengambil data perangkat jaringan yang telah terdaftar pada sistem LibreNMS. Informasi perangkat yang ditampilkan meliputi hostname perangkat, IP Address perangkat, hardware perangkat, sistem operasi perangkat, serta status perangkat (Up atau Down).



GAMBAR 10
Halaman LibreNMS Devices pada Website



GAMBAR 11

Dashboard Monitoring Perangkat pada LibreNMS

Sistem juga menyediakan fitur refresh device yang memungkinkan administrator memperbarui data perangkat dari server LibreNMS sehingga informasi yang ditampilkan selalu dalam kondisi terbaru. Dengan adanya fitur ini, aplikasi Booking Management IP Address tidak hanya berfungsi untuk mengelola pemesanan IP Address, tetapi juga membantu administrator jaringan dalam memantau perangkat yang menggunakan alamat IP tersebut.

D. Implementasi RBAC

Role-Based Access Control (RBAC) merupakan mekanisme pengaturan hak akses yang digunakan untuk memastikan bahwa setiap pengguna hanya dapat mengakses fitur sesuai dengan peran (role) yang dimilikinya. Pada implementasi RBAC dalam proyek ini, terdapat tiga jenis role, yaitu Admin, Teknisi, dan User. Penerapan RBAC ini dilakukan mulai dari proses autentikasi, penyimpanan role di database, hingga pembatasan akses pada endpoint API dan tampilan interface aplikasi.

Salah satu contoh penerapan code untuk mengatur halaman RBAC adalah dengan menggunakan variabel `canManageRoles` yang mengambil data role user. Jika data `canManageRoles` tidak sama dengan admin, maka terdapat pembatasan halaman pada tampilan antarmuka.

```
const canManageRoles = currentUser && hasPermission(currentUser.role, ['admin']);

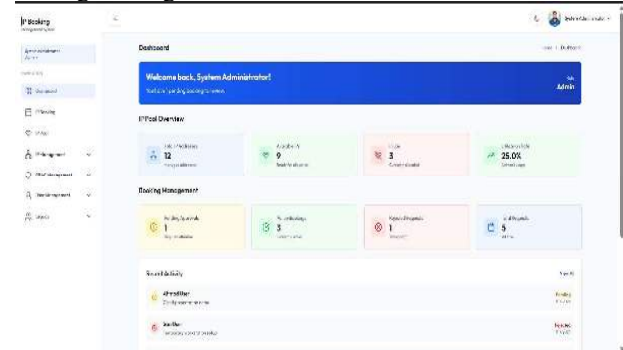
if (!canManageRoles) {
  return (
    <div className="text-center py-12">
      <Shield size={48} className="mx-auto text-gray-400 mb-4" />
      <p className="text-gray-500 dark:text-gray-400">
        You don't have permission to access role management.
      </p>
    </div>
  );
}
```

GAMBAR 12
Code Mengatur Halaman RBAC

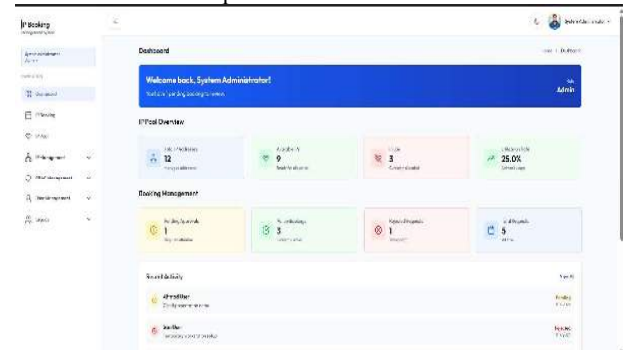


GAMBAR 13
Halaman Manage Role

Berikut adalah tampilan antarmuka berdasarkan masing-masing role:



GAMBAR 14
Tampilan Akses Role Admin



GAMBAR 15
Tampilan Akses Role Teknisi



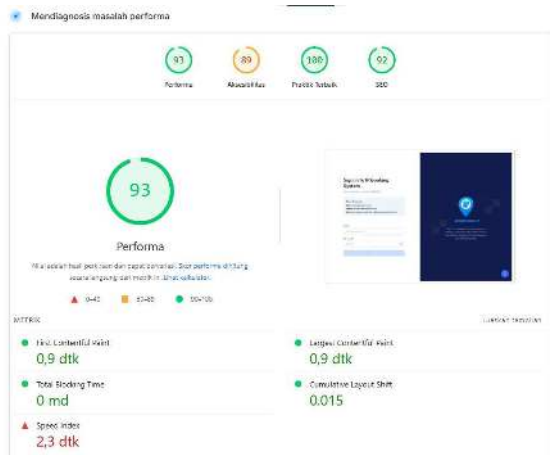
GAMBAR 16
Tampilan Akses Role User

VI. PENGUJIAN

Pengujian ini bertujuan untuk memvalidasi fungsionalitas dan kinerja sistem yang telah dibangun. Metodologi yang digunakan adalah Blackbox Testing, di mana pengujian dilakukan dari sudut pandang pengguna tanpa melihat kode sumber. Terdapat dua jenis pengujian yang dilakukan: Pengujian Kinerja Web dan Pengujian Fungsionalitas. Hasil pengujian ini akan membuktikan bahwa aplikasi dapat diimplementasikan dengan baik dan mudah digunakan.

A. Pengujian Kinerja Web

Pengujian kinerja web dilakukan untuk memastikan bahwa aplikasi IP Booking Management dapat berjalan secara optimal, cepat, dan responsif pada berbagai perangkat. Pengujian ini menggunakan Google PageSpeed Insights, yaitu alat analisis performa yang secara otomatis mengukur kecepatan, stabilitas tampilan, serta pengalaman pengguna berdasarkan metrik-metrik kinerja standar industri. Pengujian ini dilakukan pada halaman Login sebagai halaman yang pertama kali diakses oleh pengguna.



GAMBAR 17
Tampilan Mendiagnosis Performa Website (Google PageSpeed Insights)

First Contentful Paint (FCP)	0,9 detik	Konten pertama muncul sangat cepat, pengguna langsung melihat tampilan awal.
Largest Contentful Paint (LCP)	0,9 detik	Elemen terbesar halaman termuat sangat cepat (ideal < 2,5 detik).
Total Blocking Time (TBT)	0 ms	Tidak ada JavaScript berat yang menghambat interaksi pengguna.
Speed Index	2,3 detik	Proses tampilan keseluruhan cepat namun sedikit di atas batas ideal (< 1,8 detik).
Cumulative Layout Shift (CLS)	0,015	Layout sangat stabil tanpa pergeseran elemen mengganggu.

GAMBAR 18
Tabel Pengujian Metrik Performa
Hasil pengujian kinerja web disajikan pada Tabel II berikut ini:

TABEL 2 Pengujian Kinerja Web		
Kategori Pengujian	Nilai	Keterangan
Performa	93	Performa sangat baik (kategori hijau)
Aksesibilitas	89	Aksesibilitas baik, perlu sedikit perbaikan kecil
Praktik Terbaik (Best Practices)	100	Mengikuti standar praktik terbaik secara optimal
SEO	92	Optimasi SEO sangat baik

Berdasarkan hasil pengujian, dapat disimpulkan bahwa aplikasi Booking Management IP Address memiliki kinerja yang baik dengan waktu muat dan respons yang cepat. Ini memastikan pengalaman pengguna yang lancar, terutama bagi administrator jaringan yang memerlukan akses cepat untuk operasional harian.

B. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan untuk memverifikasi apakah setiap fitur dalam aplikasi berfungsi dengan benar sesuai dengan yang diharapkan. Hasil pengujian disajikan pada Tabel III berikut ini:

TABEL 3 Pengujian Fungsionalitas			
Uji Coba	Tujuan Pengujian	Hasil Pengujian	Kesimpulan
Autentikasi Pengguna	Login dengan username dan password yang benar	Berhasil masuk ke halaman dashboard	Fungsi login berjalan dengan baik dan aman

Uji Coba	Tujuan Pengujian	Hasil Pengujian	Kesimpulan
Penambahan Data IP	Menambahkan IP Address baru ke sistem	Data IP berhasil ditambahkan	Proses penambahan data berfungsi dengan baik
Peminjaman IP	Meminjam IP yang tersedia dari daftar	Status IP berhasil berubah dari tersedia menjadi terpakai	Fungsi peminjaman IP berjalan dengan baik dan status IP otomatis terperbarui
Pelepasan IP	Mengubah status IP dari terpakai menjadi tersedia	Status IP berhasil berubah menjadi tersedia	Fungsi pelepasan IP berjalan dengan baik
Pencarian Data	Mencari IP Address tertentu	IP Address ditemukan sesuai dengan data yang dicari	Fitur pencarian data berfungsi dengan baik

Hasil dari Pengujian Fungsionalitas menunjukkan bahwa semua fitur utama aplikasi, mulai dari autentikasi hingga pengelolaan IP, telah berfungsi dengan benar dan sesuai dengan kebutuhan operasional. Ini membuktikan bahwa aplikasi ini siap untuk diimplementasikan dan digunakan oleh tim teknis di PT Moratelindo.

VII. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa Aplikasi Booking Management IP Address berbasis web telah berhasil dibangun dan berjalan sesuai dengan tujuan penelitian. Aplikasi ini mampu membantu proses pengelolaan dan pemesanan IP Address yang sebelumnya dilakukan secara manual menjadi lebih terstruktur dan terkomputerisasi.

Penerapan React JS sebagai front-end dan Node.js dengan REST API sebagai back-end mampu menghasilkan sistem yang interaktif, responsif, serta mudah dikembangkan. Penggunaan database MySQL memungkinkan penyimpanan data IP Address, data pengguna, dan riwayat booking secara terpusat dan konsisten. Selain itu, penerapan Role-Based Access Control (RBAC) berhasil membatasi hak akses pengguna sesuai dengan peran masing-masing, sehingga keamanan dan pengelolaan data dalam sistem menjadi lebih terkontrol. Berdasarkan hasil pengujian fungsionalitas, seluruh fitur utama aplikasi dapat berjalan dengan baik sesuai dengan kebutuhan pengguna.

Adapun beberapa saran untuk pengembangan selanjutnya adalah menambahkan fitur notifikasi otomatis terkait status pemesanan IP Address, mengembangkan sistem agar dapat terintegrasi dengan aplikasi monitoring jaringan secara real-time yang lebih luas, meningkatkan aspek keamanan sistem dengan penerapan enkripsi data dan mekanisme autentikasi lanjutan, serta menyempurnakan tampilan antarmuka pengguna agar lebih informatif dan mudah digunakan.

DAFTAR PUSTAKA

- [1] DNSstuff, "IP Address Management: Challenges and Solutions," 2023. [Online]. Available: <https://www.dnsstuff.com>
- [2] phpIPAM, "phpIPAM OpenSource IP Address Management," 2024. [Online]. Available: <https://phpipam.net>
- [3] M. H. H. Alzakwani, H. A. A. Al-Maashri, A. K. Al-Hinai, and S. Al-Salmi, "A simulation-based approach to automating subnetting for IPv4 networks," *Int. J. Innovative Research and Scientific Studies*, vol. 8, no. 1, pp. 106–115, Jan. 2025.
- [4] NIST, "Role-Based Access Control," 2021. [Online]. Available: <https://csrc.nist.gov/projects/role-based-access-control>
- [5] LibreNMS, "LibreNMS API Documentation," 2024. [Online]. Available: <https://docs.librenms.org/API/>
- [6] ManageEngine, "IP Address Management for Enterprise Networks," 2023.
- [7] A. Pauley, C. Jeffries, A. Asrigo, R. Putman, and S. Ramesh, "Secure IP address allocation at cloud scale," *arXiv:2210.14999*, Oct. 2022.
- [8] Y. Wang, J. Zhao, and L. Xu, "Research and implementation of IP address management in medium and large-scale LANs," *Proc. ICCSIE*, Xi'an, 2022.
- [9] R. Fielding, "Architectural styles and the design of network-based software architectures," Univ. of California, Irvine, 2000.
- [10] D. Ferraiolo, D. Kuhn, and R. Chandramouli, *Role-Based Access Control*. Artech House, 2003.
- [11] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2016.
- [12] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th ed. McGraw-Hill, 2015.
- [13] T. Rooney and M. Dooley, *IP Address Management: Principles and Practice*. Wiley-IEEE Press, 2010.
- [14] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. Pearson, 2011.
- [15] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th ed. Pearson, 2017.