



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Landasan Teori

##### 2.1.1 Sistem Pemantauan (Monitoring)

Pemantauan layanan web (*web service monitoring*) merupakan mekanisme pengawasan berkelanjutan yang dirancang untuk memastikan ketersediaan, keandalan, dan responsivitas pertukaran data pada suatu infrastruktur jaringan. Dalam konteks sistem keamanan seperti pintu cerdas, *monitoring* tidak sekadar berfungsi sebagai alat peninjau lalu lintas *log*, melainkan bertindak sebagai panel kendali utama yang memberikan visibilitas *real-time* kepada *administrator* mengenai status perangkat keras di lapangan. Melalui integrasi layanan web, perangkat kontrol akses tidak lagi beroperasi secara pasif dan terisolasi, melainkan mampu melaporkan setiap *anomali* operasional maupun histori interaksi akses pengguna langsung ke pusat *server* (Tiwari & Waoo, 2024). Pemantauan terpusat ini menjadi fondasi yang mutlak diperlukan untuk merespon insiden keamanan jauh lebih cepat dibandingkan dengan metode pengecekan riwayat lokal pada memori mikrokontroler.

##### 2.1.2 Manajemen Log Akses

Efektivitas implementasi sistem pemantauan berbasis layanan web (*web service*) pada ekosistem pengunci pintu cerdas sangat bergantung pada sinergi mutakhir antara ketahanan keamanan *Server* dan tata kelola rekam jejak digital. Dalam arsitektur ini, manajemen log bertindak sebagai instrumen vital yang mengarsipkan kronologi akses secara terpusat guna memenuhi kebutuhan audit *forensik*, sementara infrastruktur keamanan bertugas membentengi pangkalan data tersebut dari berbagai vektor serangan destruktif seperti *SQL Injection* dan *Brute Force* (Praptomo, Suryanto & Kurniawan, 2023). Menurut Ramadhan et al. (2024), integrasi protokol proteksi berlapis termasuk mekanisme autentikasi *API Key* dan pembatasan laju permintaan (*Rate Limiting*) menjadi jaminan untuk memelihara kerahasiaan serta integritas data log secara *real-time*.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

### 2.1.3 Web Service

*Web service* merupakan layanan yang digunakan untuk menghubungkan pertukaran data antarsistem melalui jaringan, sehingga beberapa komponen seperti perangkat IoT, *backend*, basis data, dan antarmuka pengguna dapat saling terintegrasi. Dalam sistem monitoring berbasis IoT, *web service* berperan sebagai penghubung antara perangkat utama dengan server agar data yang dikirimkan dapat diproses, disimpan, dan ditampilkan secara terpusat melalui aplikasi web. Pada penelitian ini, *web service* digunakan untuk mendukung proses pengiriman dan pengelolaan data log akses pintu dari sistem utama menuju *backend* Flask, basis data Supabase PostgreSQL, dan *frontend* React JS, sehingga administrator dapat memantau aktivitas akses pintu secara *real-time* (Novianto & Munir, 2022; Amrulloh et al., 2024).

### 2.1.4 Decoupled Architecture

*Decoupled architecture* merupakan pendekatan arsitektur sistem yang memisahkan setiap komponen utama, seperti *frontend*, *backend*, basis data, dan layanan pendukung, agar dapat dikembangkan, dikelola, dan diuji secara lebih mandiri melalui komunikasi API; pada penelitian ini, pendekatan tersebut digunakan untuk memisahkan antarmuka React JS, layanan *backend* Flask, basis data Supabase PostgreSQL, dan sistem utama *smart door lock*, sehingga proses monitoring dan manajemen log akses pintu dapat berjalan lebih fleksibel, mudah diintegrasikan, serta lebih mudah dianalisis dari sisi fungsi, keamanan, dan performa sistem (Lercher et al., 2023; Bogner et al., 2023).

### 2.1.5 Framework Flask



Gambar 2. 1 Framework *Flask*



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Arsitektur perangkat lunak yang ringan sekaligus modular merupakan identitas utama *Flask*. Bertindak sebagai *microframework* berbasis Python, ekosistem ini dinilai sangat ideal guna menopang pengembangan layanan *backend* beserta *endpoint RESTful API* (Kusuma & Hidayat, 2024). Ketika sistem mengemas seluruh pustaka bawaan sejak awal, *Flask* justru membebaskan pengembang dari tumpukan kode yang tidak perlu. Pemrogram memiliki otonomi penuh untuk menyematkan dependensi yang benar-benar esensial bagi sistem, semisal konektor komputasi awan atau modul keamanan ekstra (Putri dkk., 2023).

### 2.1.6 Framework React JS



Gambar 2. 2 Framework React JS

Pengembangan *Single Page Application* (SPA) yang dinamis dan interaktif pada sisi antarmuka pengguna jamak memanfaatkan React JS, sebuah pustaka *frontend* berbasis JavaScript yang mengusung *Component-Based Architecture* (Santoso & Arifin, 2023). Efisiensi komputasinya ditopang oleh *Virtual DOM* (*Document Object Model*). Teknologi ini memangkas beban kerja *browser* karena perenderan ulang (*re-rendering*) hanya dieksekusi pada komponen antarmuka yang mengalami pembaruan data, tanpa harus memuat kembali seluruh halaman secara utuh (Setiawan dkk., 2025). Sifat perenderan asinkron yang responsif tersebut mempermudah koordinasi data antar-arsitektur. Berkat karakteristik ini, React JS memiliki kompatibilitas tinggi saat disandingkan dengan arsitektur *backend* terpisah (*Decoupled Architecture*), yang pada akhirnya memungkinkan administrator memantau log keamanan serta status pintu cerdas secara instan tanpa kendala jeda pemrosesan visual (Fauzi & Lestari, 2024).



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

### 2.1.7 PostgreSQL



Gambar 2. 3 PostgreSQL

Ketangguhan performa server bermuara pada ketepatan penentuan mesin pengelola basis data. Guna menjawab kebutuhan tersebut, PostgreSQL menyediakan sistem manajemen basis data objek-relasional (*Object-Relational Database Management System / ORDBMS*) berskala *enterprise* yang berbasis *open-source*. Reputasi sistem ini bersandar pada tingkat keandalan yang tinggi, kepatuhan ketat terhadap standar SQL, serta ketangkasannya dalam mempertahankan integritas data di tengah beban kerja yang masif (Wahyudi & Utomo, 2022). Menariknya, kapabilitas PostgreSQL tidak terbatas pada model relasional konvensional semata. Ia juga adaptif terhadap struktur data non-relasional seperti JSONB, sebuah fitur esensial yang menjembatani lalu lintas pertukaran data dalam arsitektur API modern. Guna menggaransi keamanan, teknologi ini menerapkan mekanisme *Multi-Version Concurrency Control* (MVCC). Melalui MVCC, instruksi pembacaan dan penulisan transaksi dapat dieksekusi secara serempak tanpa memicu risiko konflik data atau *data deadlock* (Hidayat & Kurniawan, 2024).

### 2.1.8 Supabase



Gambar 2. 4 Supabase

Penyediaan infrastruktur *server* kontemporer saat ini mengalami simplifikasi signifikan berkat ekspansi platform *Backend-as-a-Service* (BaaS) seperti Supabase, yang dikembangkan sebagai alternatif *open-source* bagi Firebase. Tanpa menuntut pengembang untuk merancang seluruh ekosistem *server* dari titik awal, Supabase berhasil mengombinasikan ketangguhan basis data relasional tingkat lanjut dengan



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

kelenturan arsitektur web modern (Lestari & Saputra, 2023). Platform ini secara praktis membungkus pangkalan data dengan beragam utilitas terpadu, seperti manajemen autentikasi pengguna, penyimpanan berkas (*storage*), serta REST API yang diproduksi secara otomatis. Fleksibilitas tersebut kian optimal karena performanya yang responsif. (Pratama et al., 2024) menggarisbawahi bahwa keunggulan utama Supabase terletak pada kapabilitas sinkronisasi secara *real-time*. Aliran data dari level basis data akan langsung diteruskan ke aspek antarmuka aplikasi saat itu juga, sebuah mekanisme instan yang menjadikannya opsi paling ideal untuk menyokong sistem pemantauan dengan tingkat akurasi tinggi.

### 2.1.9 Virtual Private Server (VPS)



Gambar 2. 5 *Virtual Private Server (VPS)*

Aksesibilitas global tanpa interupsi bagi layanan *web service* dan dashbor pemantauan sangat bergantung pada keandalan infrastruktur *Virtual Private Server (VPS)*. Teknologi virtualisasi tingkat *server* ini mengompartemenkan satu perangkat keras fisik menjadi beberapa lingkungan virtual yang terisolasi secara mandiri. Setiap lingkungan virtual bekerja dengan alokasi sumber daya terdedikasi seperti CPU, RAM, serta kapasitas penyimpanan dan menjalankan sistem operasinya sendiri, sebuah mekanisme yang memberikan kendali administratif penuh (*root access*) bagi pengembang (Sanjaya & Budiman, 2023). Integrasi VPS dalam ekosistem IoT dan *web service* diakui oleh Nugroho (2024) mampu memberikan fleksibilitas tinggi, khususnya untuk kebutuhan konfigurasi keamanan jaringan, manajemen port, hingga instalasi dependensi perangkat lunak seperti *server Flask*. Keunggulan ini disempurnakan oleh adanya lapisan isolasi yang kokoh. Efeknya, fluktuasi beban komputasi ataupun gangguan keamanan dari *server* virtual lain tidak akan mengganggu stabilitas operasional sistem pemantauan yang tengah berjalan.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

### 2.1.10 Keamanan Sistem Web

Keamanan sistem web merupakan aspek penting dalam pengembangan aplikasi berbasis jaringan karena sistem yang terhubung dengan internet memiliki risiko terhadap akses tidak sah, pencurian data, manipulasi informasi, kesalahan konfigurasi, hingga gangguan layanan. Dalam konteks aplikasi *web service*, keamanan tidak hanya berfokus pada perlindungan halaman antarmuka, tetapi juga mencakup pengamanan *endpoint*, autentikasi pengguna, otorisasi akses, validasi input, pengelolaan token, pencatatan aktivitas, serta penanganan kesalahan agar tidak membocorkan informasi sensitif. OWASP Top Ten 2025 menempatkan risiko seperti *Broken Access Control*, *Security Misconfiguration*, *Injection*, dan *Authentication Failures* sebagai ancaman utama yang perlu diperhatikan dalam pengujian keamanan aplikasi web, sehingga penerapan mekanisme perlindungan pada sistem monitoring dan manajemen log akses pintu menjadi diperlukan untuk menjaga kerahasiaan, integritas, dan ketersediaan data.

### 2.1.11 Autentikasi API Key

Autentikasi API Key merupakan mekanisme pengamanan yang digunakan untuk memverifikasi identitas aplikasi, perangkat, atau layanan yang melakukan permintaan data ke *backend* melalui API. API Key umumnya berbentuk kode unik yang dikirimkan bersama *request*, biasanya melalui *header*, sehingga server dapat menentukan apakah permintaan tersebut berasal dari klien yang sah atau tidak. Dalam sistem berbasis IoT dan *web service*, API Key dapat digunakan sebagai lapisan awal untuk membatasi akses dari perangkat atau layanan yang tidak memiliki izin, namun penggunaannya perlu dikelola dengan aman karena API Key yang bocor dapat disalahgunakan untuk mengakses layanan secara tidak sah. OWASP API Security Top Ten 2023 juga menegaskan bahwa API Key lebih tepat digunakan untuk autentikasi klien API, bukan sebagai autentikasi utama pengguna manusia, sehingga pada penelitian ini API Key digunakan bersama mekanisme lain seperti JWT untuk memperkuat pengamanan akses sistem.

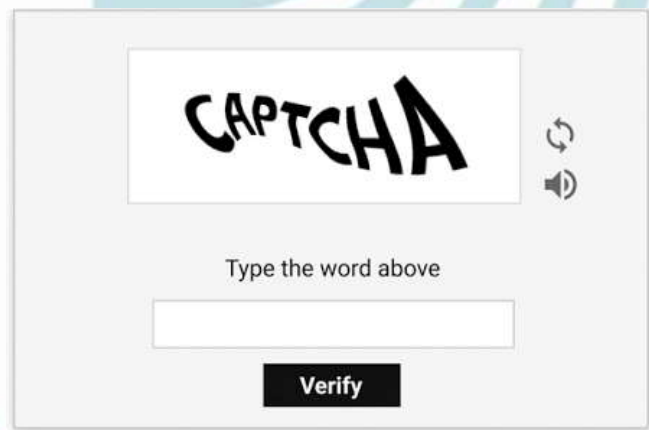
### 2.1.12 JSON Web Token / JWT

JSON Web Token atau JWT merupakan mekanisme autentikasi berbasis token yang digunakan untuk mengelola sesi pengguna secara lebih ringan dan fleksibel

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

pada aplikasi berbasis web maupun *web service*. Setelah pengguna berhasil melakukan login, server akan menghasilkan token yang berisi informasi tertentu dan digunakan kembali oleh klien untuk mengakses *endpoint* yang membutuhkan otorisasi. Pada sistem modern, JWT banyak digunakan karena mendukung komunikasi yang bersifat *stateless*, sehingga server tidak perlu menyimpan sesi pengguna secara langsung pada sisi server. Penerapan JWT pada sistem *web service* dapat membantu meningkatkan keamanan proses autentikasi dan pertukaran data, terutama apabila dikombinasikan dengan algoritma penandatanganan seperti HMAC SHA-256 atau SHA-512 serta pengelolaan token yang aman. Dalam penelitian ini, JWT digunakan untuk memastikan hanya administrator yang telah terautentikasi yang dapat mengakses halaman dan data log pada sistem monitoring.

**2.1.13 CAPTCHA**

Gambar 2. 6 Captcha

*CAPTCHA* atau *Completely Automated Public Turing test to tell Computers and Humans Apart* merupakan salah satu mekanisme verifikasi yang digunakan dalam keamanan sistem informasi untuk membedakan pengguna manusia dari program otomatis. Sistem ini bekerja dengan memberikan tantangan tertentu, baik berupa teks, gambar, audio, maupun pola interaksi, yang umumnya masih dapat dipahami oleh manusia, tetapi tidak mudah diselesaikan oleh bot atau skrip otomatis. Prinsip kerja tersebut menjadikan *CAPTCHA* sebagai bentuk *reverse Turing test*, karena sistem tidak hanya menerima masukan dari pengguna, melainkan juga menguji apakah aktivitas tersebut benar-benar berasal dari manusia. Penerapannya banyak digunakan untuk menekan risiko penyalahgunaan layanan digital, seperti spam,



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

pembuatan akun massal, pengisian formulir secara otomatis, *scraping* data, dan akses tidak sah yang dilakukan melalui otomatisasi. Meski demikian, perkembangan kecerdasan buatan, *computer vision*, dan pembelajaran mesin membuat desain *CAPTCHA* perlu terus disesuaikan, sebab beberapa model komputasi modern telah mampu mengenali pola visual maupun audio dengan tingkat akurasi yang semakin baik. Oleh karena itu, *CAPTCHA* modern tidak cukup hanya mengandalkan distorsi karakter atau tantangan visual sederhana, tetapi juga perlu mempertimbangkan keseimbangan antara keamanan, kemudahan penggunaan, aksesibilitas, dan ketahanan terhadap serangan otomatis berbasis AI (Acien et al., 2021; Hossen & Hei, 2022; Searles et al., 2023; Uysal, 2025).

#### 2.1.14 Rate Limiting

*Rate limiting* merupakan mekanisme pembatasan jumlah permintaan atau *request* yang dapat dilakukan oleh pengguna, perangkat, atau alamat IP dalam rentang waktu tertentu untuk menjaga kestabilan layanan dan mencegah penyalahgunaan sistem. Pada aplikasi berbasis *web service*, mekanisme ini penting diterapkan pada endpoint yang sensitif, seperti login administrator atau pengiriman data ke server, karena percobaan akses yang terlalu banyak dalam waktu singkat dapat menyebabkan beban berlebih pada CPU, memori, dan jaringan server. OWASP API Security Top Ten 2023 juga menempatkan konsumsi sumber daya yang tidak dibatasi sebagai salah satu risiko keamanan API, sehingga *rate limiting* menjadi salah satu perlindungan yang relevan untuk mengurangi risiko *brute force*, *request flooding*, dan gangguan ketersediaan layanan.

#### 2.1.15 Parameterized Query

*Parameterized query* merupakan teknik pengamanan basis data yang digunakan untuk memisahkan struktur perintah SQL dengan data input dari pengguna, sehingga input yang dikirimkan ke sistem tidak langsung diperlakukan sebagai bagian dari perintah SQL. Teknik ini menjadi penting dalam pengembangan aplikasi web karena dapat mengurangi risiko serangan SQL Injection, yaitu serangan yang memanfaatkan celah input untuk menyisipkan perintah SQL berbahaya ke dalam basis data. OWASP Top Ten 2025 menjelaskan bahwa salah satu cara utama untuk mencegah *injection* adalah menjaga agar data tetap terpisah



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

dari perintah atau *query*, sedangkan penelitian Sidik et al. (2023) juga menunjukkan bahwa *parameterized query* membantu database driver mengeksekusi input secara lebih aman sebagai parameter, bukan sebagai instruksi SQL.

#### 2.1.16 Row Level Security (RLS)

Sebagai paradigma proteksi mutakhir pada RDBMS sekelas *PostgreSQL*, *Row Level Security* (RLS) menawarkan kontrol akses granular melalui pembatasan hak baca dan tulis langsung di tingkat baris data, alih-alih mengunci keseluruhan tabel secara kaku (Pratama & Susanto, 2023). Secara arsitektural, instruksi eksekusi data disaring secara dinamis lewat evaluasi kebijakan keamanan yang bersandar pada identitas pengguna atau konteks sesi aktif (Rahmawati dkk., 2022). Peran RLS bertransformasi menjadi pilar pertahanan krusial saat diintegrasikan pada arsitektur *Backend-as-a-Service* (BaaS) seperti Supabase guna mengamankan ekosistem IoT dari infiltrasi langsung via API. Dampak konkretnya, sistem akan memblokir operasi pembacaan maupun modifikasi log secara otomatis jika peminta tidak mengantongi otorisasi *JSON Web Token* (JWT) yang valid, sebuah pembatasan ketat yang menjamin otentisitas rekam jejak (*audit trail*) dari target manipulasi peretas (Nugroho & Wijaya, 2024).

#### 2.1.17 OWASP Top Ten

OWASP Top Ten merupakan dokumen acuan yang digunakan untuk memahami risiko keamanan paling kritis pada aplikasi web, sehingga dapat dijadikan dasar dalam proses pengujian dan evaluasi keamanan sistem. Pada penelitian ini, OWASP Top Ten digunakan sebagai pendekatan untuk menguji keamanan sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service*, terutama karena sistem memiliki beberapa komponen yang terhubung melalui jaringan, seperti *frontend* React JS, *backend* Flask, basis data Supabase PostgreSQL, dan layanan pada VPS. OWASP Top Ten 2025 memuat sepuluh kategori risiko utama, mulai dari kelemahan kontrol akses, kesalahan konfigurasi, kelemahan autentikasi, risiko *injection*, hingga penanganan kondisi kesalahan yang tidak tepat, sehingga kategori tersebut relevan digunakan sebagai dasar penyusunan skenario pengujian keamanan pada sistem ini.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Tabel 2. 1 Tabel Kategori OWASP Top Ten

| Kode | Kategori OWASP Top Ten                     | Kaitan dengan Sistem                                                                                                                            |
|------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| A01  | <i>Broken Access Control</i>               | Berkaitan dengan pengujian akses endpoint, halaman dashboard, dan data log agar hanya dapat diakses oleh administrator yang memiliki otorisasi. |
| A02  | <i>Security Misconfiguration</i>           | Berkaitan dengan konfigurasi Flask, VPS, CORS, mode debug, serta respons error agar tidak membocorkan informasi sensitif.                       |
| A03  | <i>Software Supply Chain Failures</i>      | Berkaitan dengan keamanan dependensi atau package yang digunakan pada React JS, Flask, dan library pendukung lainnya.                           |
| A04  | <i>Cryptographic Failures</i>              | Berkaitan dengan perlindungan data sensitif, token JWT, API Key, serta pengamanan komunikasi data antara klien dan server.                      |
| A05  | <i>Injection</i>                           | Berkaitan dengan pengujian input berbahaya, terutama SQL Injection pada endpoint login, pencarian, atau pengelolaan log akses.                  |
| A06  | <i>Insecure Design</i>                     | Berkaitan dengan rancangan alur sistem, validasi akses pada backend, serta pencegahan bypass melalui akses langsung ke API.                     |
| A07  | <i>Authentication Failures</i>             | Berkaitan dengan pengujian login administrator, CAPTCHA, <i>rate limiting</i> , jumlah login gagal, dan pencegahan percobaan brute force.       |
| A08  | <i>Software or Data Integrity Failures</i> | Berkaitan dengan perlindungan integritas data log akses agar tidak dapat diubah atau dihapus oleh pihak yang tidak berwenang.                   |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|     |                                               |                                                                                                                                                                 |
|-----|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A09 | <i>Security Logging and Alerting Failures</i> | Berkaitan dengan pencatatan aktivitas penting, seperti login gagal, akses tanpa token, dan percobaan akses tidak sah.                                           |
| A10 | <i>Mishandling of Exceptional Conditions</i>  | Berkaitan dengan cara sistem menangani kondisi tidak normal, seperti JSON rusak, field kosong, method salah, atau error server tanpa membocorkan detail teknis. |

### 2.1.18 Postman



Gambar 2. 7 Postman

Postman merupakan salah satu aplikasi yang banyak digunakan dalam proses pengembangan perangkat lunak, terutama untuk membantu pengujian dan pengelolaan *Application Programming Interface* (API). Melalui platform ini, pengembang dapat merancang, mendokumentasikan, menguji, serta memantau layanan API dengan lebih mudah tanpa harus langsung menghubungkannya ke antarmuka utama aplikasi (Pujiyanto dkk., 2024). Postman bekerja dengan cara mengirimkan permintaan HTTP ke *endpoint* server, misalnya menggunakan metode GET, POST, PUT, atau DELETE, kemudian menampilkan respons yang diberikan server dalam format data terstruktur seperti JSON maupun XML (Karmelia, 2024). Dengan cara kerja tersebut, pengembang dapat memeriksa apakah fungsi pada sisi *backend* sudah berjalan sesuai kebutuhan. Selain itu, Postman juga membantu proses pencarian kesalahan karena pengujian dapat dilakukan secara terpisah, sehingga bagian yang belum sesuai dapat diketahui lebih awal sebelum sistem diintegrasikan secara penuh ke dalam aplikasi (Nugraha, 2024).



### 2.1.19 SQLMap

Sebagai kerangka kerja *open-source* yang esensial, SQLmap mengotomatiskan identifikasi sekaligus eksploitasi celah *SQL Injection* pada platform berbasis web selama pelaksanaan audit siber maupun *penetration testing* (Kurniawan dkk., 2025). Berbekal mesin analitik tingkat lanjut, utilitas ini mampu memetakan arsitektur sistem manajemen basis data (DBMS) secara jarak jauh. Ia bahkan sanggup mengekstraksi catatan rahasia dari tabel penyimpanan hingga menjalankan instruksi komando langsung pada sistem operasi *server* lewat serangkaian manuver pengujian yang masif (Salsabila & Fithra, 2025). Kapasitas teknis alat ini ditopang oleh kapabilitasnya dalam mengadopsi berbagai teknik injeksi, mulai dari *Boolean-based blind*, *Time-based blind*, *Error-based*, sampai kueri *UNION*. Rangkaian metode tersebut dieksekusi dengan menembakkan muatan data (*payload*) secara kontinu guna mengevaluasi seberapa ketat sebuah situs memvalidasi input penggunaanya (Rahman dkk., 2024). Integrasi fungsionalitas yang komprehensif ini pada akhirnya menempatkan SQLmap sebagai instrumen mutlak. Para analis keamanan memanfaatkannya untuk melacak risiko kebocoran informasi, menguji imunitas basis data dari anomali siber, dan membangun perimeter mitigasi yang presisi sebelum sebuah sistem resmi diluncurkan ke ranah publik (Nugraha dkk., 2024).

### 2.1.20 Hydra

Sebagai utilitas peretas kata sandi lintas protokol, Hydra memegang peranan sentral saat mengeksekusi simulasi *brute-force* dan serangan berbasis kamus secara serentak pada ranah audit siber maupun *penetration testing* (Pratama, 2024). Alat bantu ini secara khusus mengusung arsitektur *multithreaded* berintensitas tinggi. Struktur tersebut memfasilitasi para penguji untuk mengevaluasi ketangguhan mekanisme autentikasi secara kilat pada aneka rupa layanan yang berjalan paralel, mencakup protokol SSH, FTP, HTTP/HTTPS, hingga sistem basis data (Saputra dkk., 2025). Fleksibilitas Hydra dalam menyerap rujukan *wordlist* eksternal menjadikannya instrumen mutlak bagi riset keamanan jaringan modern. Melalui metodologi pengujian ini, analis dapat mendeteksi titik lemah pada kredensial pengguna dan menakar daya tahan pertahanan *server* terhadap gempuran

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

permintaan masuk yang masif; sebuah langkah preventif esensial guna memastikan skenario mitigasi beroperasi presisi sebelum celah keamanan yang ada telanjur dieksploitasi oleh peretas (Yusuf & Ramadhan, 2026).

### 2.1.21 Pengujian Fungsional Berbasis Test Script

Pengujian fungsional berbasis *test script* merupakan proses pengujian yang dilakukan dengan mengikuti skenario uji yang telah disusun secara terstruktur untuk memastikan setiap fungsi pada sistem berjalan sesuai kebutuhan. Dalam pengujian ini, setiap fitur diuji berdasarkan data uji, langkah pengujian, hasil yang diharapkan, hasil aktual, serta status keberhasilan pengujian. Pendekatan ini membantu proses validasi menjadi lebih rapi dan mudah ditelusuri karena setiap hasil pengujian terdokumentasi secara jelas. Pada penelitian ini, *test script* digunakan untuk menguji fungsi utama website monitoring, seperti login administrator, dashboard, daftar wajah pengguna, kelola pengguna, log akses pintu, pencarian data log, dan logout (Olsina et al., 2022; ISTQB, 2024).

### 2.1.22 Pengujian Kualitas Website

Pengujian kualitas website merupakan proses evaluasi terhadap halaman web untuk mengetahui apakah website telah memiliki kualitas yang baik dari sisi *performance*, *accessibility*, *best practices*, dan SEO. Pengujian ini tidak diarahkan untuk mengukur kemampuan server atau VPS dalam menerima banyak *request*, melainkan untuk menilai kualitas halaman yang digunakan oleh pengguna, seperti kecepatan pemuatan halaman, kemudahan akses, penerapan praktik pengembangan web yang baik, serta struktur halaman agar lebih mudah dipahami oleh pengguna maupun sistem pencarian. Nufus dan Saputri (2026) menggunakan Google Lighthouse untuk melakukan pengujian teknis website dengan menilai aspek *performance*, *accessibility*, *best practices*, dan SEO, sedangkan penelitian Faizin, Nevin, dan Yuhana (2024) menunjukkan bahwa evaluasi website menggunakan Lighthouse dapat digunakan untuk menilai kualitas halaman dari sisi *performance* dan *accessibility*. Pada penelitian ini, pengujian kualitas website digunakan untuk mengevaluasi halaman website monitoring dan manajemen log akses pintu agar sistem tidak hanya berjalan sesuai fungsi, tetapi juga memiliki kualitas halaman yang layak digunakan oleh administrator.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

### 2.1.23 Lighthouse

*Lighthouse* merupakan alat audit otomatis yang digunakan untuk mengevaluasi kualitas teknis halaman website berdasarkan beberapa aspek utama, yaitu *performance*, *accessibility*, *best practices*, dan SEO. Dalam pengujian website, *Lighthouse* membantu pengembang memperoleh gambaran objektif mengenai kualitas halaman melalui skor pengujian, sehingga bagian yang masih perlu diperbaiki dapat diidentifikasi dengan lebih jelas. Nufus dan Saputri (2026) menjelaskan bahwa pengujian teknis menggunakan *Google Lighthouse* dapat digunakan untuk menilai kualitas website melalui aspek *performance*, *accessibility*, *best practices*, dan *search engine optimization* atau SEO. Hal serupa juga digunakan oleh Setiawan (2025) dalam evaluasi website sistem informasi, di mana *Lighthouse* digunakan untuk mengukur empat kategori utama tersebut sebagai representasi kualitas teknis dan pengalaman pengguna pada website. Pada penelitian ini, *Lighthouse* digunakan untuk mengevaluasi kualitas halaman website monitoring dan manajemen log akses pintu agar halaman yang dikembangkan tidak hanya berfungsi, tetapi juga memiliki kualitas teknis yang layak digunakan oleh administrator.

### 2.1.24 Performance

*Performance* merupakan aspek pengujian kualitas website yang digunakan untuk menilai kecepatan dan efisiensi halaman saat dimuat oleh pengguna. Pada pengujian menggunakan *Lighthouse*, aspek ini dapat menggambarkan sejauh mana halaman website mampu menampilkan konten dengan baik tanpa memberikan waktu tunggu yang terlalu lama kepada pengguna. Dalam konteks sistem monitoring, nilai *performance* menjadi penting karena administrator membutuhkan akses halaman yang cepat dan responsif untuk melihat informasi log akses pintu secara *real-time*. Nufus dan Saputri (2026) menunjukkan bahwa aspek *performance* pada pengujian *Lighthouse* dapat digunakan untuk mengidentifikasi kebutuhan perbaikan pada kecepatan pemuatan halaman dan stabilitas akses website



### 2.1.25 Accessibility

*Accessibility* merupakan aspek yang menilai kemudahan halaman website untuk diakses dan digunakan oleh berbagai pengguna, termasuk pengguna yang membutuhkan struktur halaman yang jelas, label formulir yang tepat, kontras tampilan yang memadai, serta navigasi yang mudah dipahami. Pengujian *accessibility* penting karena website yang baik tidak hanya dilihat dari tampilan visual, tetapi juga dari sejauh mana pengguna dapat mengakses informasi dan fitur tanpa hambatan. Pada penelitian ini, aspek *accessibility* digunakan untuk menilai apakah halaman website monitoring dan manajemen log akses pintu memiliki struktur yang mudah digunakan oleh administrator. Evaluasi website menggunakan *Lighthouse* juga sering menempatkan *accessibility* sebagai salah satu parameter utama dalam menilai kualitas teknis halaman web.

### 2.1.26 Best Practices

*Best practices* merupakan aspek pengujian yang digunakan untuk menilai apakah halaman website telah mengikuti praktik pengembangan web yang baik dari sisi teknis, keamanan dasar, dan kestabilan halaman. Aspek ini dapat membantu pengembang mengetahui apakah terdapat bagian pada website yang masih berpotensi menurunkan kualitas pengalaman pengguna, seperti penggunaan teknologi yang kurang sesuai, masalah pada kode halaman, atau penerapan praktik teknis yang belum optimal. Dalam penelitian ini, *best practices* digunakan untuk mengevaluasi apakah website monitoring telah dikembangkan dengan struktur dan praktik teknis yang layak sebelum digunakan oleh administrator. Nufus dan Saputri (2026) juga memasukkan *best practices* sebagai salah satu aspek penilaian teknis dalam evaluasi kualitas website menggunakan *Google Lighthouse*.

### 2.1.27 SEO

SEO atau *Search Engine Optimization* merupakan aspek yang berkaitan dengan struktur halaman website agar lebih mudah dikenali, dibaca, dan dianalisis oleh mesin pencari. Meskipun website monitoring pada penelitian ini bersifat administratif dan tidak ditujukan sebagai website publik, aspek SEO tetap relevan untuk menilai kerapian struktur dasar halaman, seperti penggunaan judul halaman, deskripsi, tautan, dan elemen pendukung lainnya. Dengan struktur SEO yang baik,

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

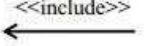
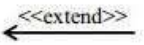


**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

halaman website menjadi lebih terorganisir dan sesuai dengan standar teknis dasar pengembangan web. Dalam evaluasi kualitas website, SEO termasuk salah satu kategori yang digunakan pada *Lighthouse* untuk menilai kelengkapan dan keterbacaan struktur halaman.

### 2.1.28 Use Case Diagram

| Simbol                                                                              | Keterangan                                                                                                                      |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|    | Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>                           |
|    | <i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor                                                               |
|    | <i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>                                              |
|  | <i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>                          |
|  | Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya                        |
|  | Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi |

Gambar 2. 8 Use Case Diagram

Pemodelan interaksi fungsional antara sistem dan entitas eksternalnya (*actor*) mutlak membutuhkan *use case diagram* sebagai medium visualisasi utama pada ranah rekayasa perangkat lunak berbasis objek. Komponen penyusun *Unified Modeling Language* (UML) ini sama sekali tidak ditujukan untuk menguliti kerumitan logika atau alur internal program. Fokus utamanya justru menyoroti sudut pandang pengguna terkait batasan operasional serta layanan apa saja yang mampu difasilitasi oleh sistem tersebut (Nugroho & Pratama, 2023). Arsitektur rancang bangun ini kemudian dapat dipetakan secara presisi sedari fase awal



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

analisis. Proyeksi grafisnya secara gamblang memadukan elemen *use case* yang mendeskripsikan unit fungsionalitas dengan aktor, baik yang berwujud entitas manusia maupun instrumen perangkat keras. Menurut Handayani et al. (2024), pengaplikasian diagram ini memegang peran strategis guna menautkan kebutuhan spesifik klien (*user requirements*) dengan cetak biru perancangan *server*.

### 2.1.29 Flowchart

|  |                                                                                                                                                                           |  |                                                                                                                                     |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-------------------------------------------------------------------------------------------------------------------------------------|
|  | <b>Flow Direction symbol</b><br>Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line. |  | <b>Simbol Manual Input</b><br>Simbol untuk pemasukan data secara manual on-line keyboard                                            |
|  | <b>Terminator Symbol</b><br>Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan                                                                    |  | <b>Simbol Preparation</b><br>Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage. |
|  | <b>Connector Symbol</b><br>Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.                                                   |  | <b>Simbol Predefine Proses</b><br>Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure                                     |
|  | <b>Connector Symbol</b><br>Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.                                                 |  | <b>Simbol Display</b><br>Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.       |
|  | <b>Processing Symbol</b><br>Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer                                                                               |  | <b>Simbol disk and On-line Storage</b><br>Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.                |
|  | <b>Simbol Manual Operation</b><br>Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer                                                                   |  | <b>Simbol magnetik tape Unit</b><br>Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.  |
|  | <b>Simbol Decision</b><br>Simbol pemilihan proses berdasarkan kondisi yang ada.                                                                                           |  | <b>Simbol Punch Card</b><br>Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu                      |
|  | <b>Simbol Input-Output</b><br>Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya                                                   |  | <b>Simbol Dokumen</b><br>Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.       |

Gambar 2. 9 Flowchart

Bagan alir memegang peranan krusial sebagai medium representasi visual yang memetakan algoritma, alur kerja, beserta logika sekuensial perangkat lunak secara metodis (Suryanto & Hidayatullah, 2023). Model grafis ini dibangun menggunakan deretan notasi geometris baku yang saling dihubungkan oleh garis penunjuk arah. Tujuannya sangat terarah. Kerumitan operasional komputasi dapat direduksi secara



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

drastis, sehingga arsitektur sistem menjadi jauh lebih intuitif untuk dipahami dan dievaluasi secara mendalam (Mahendra dkk., 2022). Pada ranah pengembangan keamanan *Internet of Things* (IoT) terdistribusi, instrumen ini diaplikasikan layaknya cetak biru fundamental. Rangkaian eksekusinya dijabarkan dengan tingkat presisi tinggi. Pemodelan tersebut mencakup prosedur autentikasi kredensial, rute transmisi paket data menuju *server*, hingga logika pengambilan keputusan manakala sistem merespons adanya anomali akses. Pendekatan strategis ini memberi ruang bagi pengembang guna memvalidasi kesahihan logika program jauh sebelum fase penulisan kode dimulai (Wibowo & Kartika, 2024). Merujuk pada ilustrasi yang telah disajikan sebelumnya, ragam simbol standar yang menyusun diagram tersebut secara efektif merangkum gambaran umum dari operasional sistem menjadi satu kesatuan alur yang terintegrasi.

## 2.2 Penelitian Terdahulu

Tabel 2. 2 Perbandingan Penelitian Terdahulu

| Jurnal                                                                               | Hasil                                                                                                                                                                                                  | Perbedaan                                                                                                                             |
|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Penulis:</b><br>Setiawan,<br>dkk.<br><br><b>Tahun :</b> 2025                      | Membangun sistem monitoring <i>real-time</i> berbasis IoT.<br>Menggunakan arsitektur RESTful Web Service untuk menghubungkan <i>hardware</i> (sensor) ke antarmuka web secara responsif dan fleksibel. | Membangun Pladen ( <i>Backend</i> ) layanan web untuk manajemen <i>log</i> keamanan pintu cerdas dan riwayat aktivitas administrator. |
| <b>Penulis:</b><br>Rahmanda,<br>A., &<br>Kusuma, W.<br>A.<br><br><b>Tahun :</b> 2025 | Menguji keamanan komunikasi <i>Server backend</i> .<br>Membuktikan bahwa penggunaan token keamanan sangat efektif mencegah pencetakan dan manipulasi pertukaran data JSON.                             | Mengamankan <i>endpoint</i> log aktivitas perangkat keras dan mengintegrasikannya dengan sistem <i>cloud</i> PostgreSQL (Supabase).   |



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|                                                                                  |                                                                                                                                                                        |                                                                                                         |
|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| <b>Penulis:</b><br>Fikri, M. N.,<br>& Setiawan,<br>A.<br><br><b>Tahun :</b> 2022 | Melakukan pengujian penetrasi dengan alat SQLMap.<br>Mengonfirmasi bahwa perlindungan sistem pada kueri berhasil mencegah pencurian paksa struktur tabel basis data.   | Menguji celah basis data pada lingkungan <i>Decoupled Architecture</i> berbasis <i>framework</i> Flask. |
| <b>Penulis:</b><br>Pratama, R.,<br>& Anwar, K.<br><br><b>Tahun :</b> 2026        | Melakukan simulasi serangan <i>Brute Force</i> dengan Hydra.<br>Berhasil memblokir serangan menggunakan konfigurasi pembatasan lalu lintas pada lapisan jaringan.      | Melakukan mitigasi di tingkat aplikasi lunak melalui <i>Rate Limiting</i> pada <i>endpoint login</i> .  |
| <b>Penulis:</b><br>Situmorang et al.<br><br><b>Tahun :</b> 2026                  | Menerapkan tantangan CAPTCHA pada antarmuka <i>login</i> . Implementasi ini terbukti akurat dalam menyaring skrip bot otomatis sehingga mencegah pengambilalihan akun. | Memadukan CAPTCHA dan <i>JSON Web Token (JWT)</i> khusus untuk mengunci sesi tunggal administrator.     |



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB III

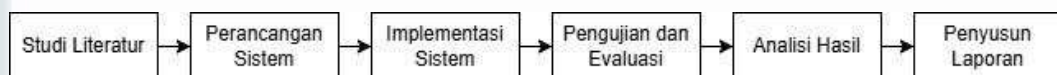
### METODE PENELITIAN

#### 3.1 Rancangan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimental karena sistem yang dikembangkan tidak hanya dirancang dan diimplementasikan, tetapi juga diuji secara terukur untuk mengetahui kesesuaian fungsi, keamanan, dan kualitas halaman website. Sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service* dirancang menggunakan konsep *decoupled architecture*, yaitu dengan memisahkan komponen *backend* Flask, *frontend* React JS, basis data Supabase PostgreSQL, dan Virtual Private Server sebagai lingkungan *deployment*. Pengujian dilakukan melalui tiga pendekatan, yaitu pengujian fungsional website menggunakan *test script* untuk memastikan fitur utama berjalan sesuai kebutuhan, pengujian keamanan berdasarkan OWASP Top Ten untuk mengevaluasi potensi risiko pada layanan *web service*, serta pengujian kualitas website menggunakan Lighthouse untuk menilai aspek *performance*, *accessibility*, *best practices*, dan SEO. Hasil dari pengujian tersebut kemudian dianalisis secara deskriptif untuk menilai apakah sistem yang dibangun mampu berjalan sesuai fungsi, memiliki mekanisme keamanan yang memadai, serta memiliki kualitas halaman website yang layak digunakan oleh administrator.

#### 3.2 Tahapan Penelitian

Tahapan penelitian disusun untuk memberikan gambaran alur pelaksanaan penelitian mulai dari pengumpulan referensi hingga penarikan kesimpulan. Pada penelitian ini, tahapan penelitian terdiri dari enam tahap utama, yaitu sebagai berikut:



Gambar 3. 1 Tahapan Penelitian



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1. Studi Literatur

Tahap studi literatur dilakukan dengan mengumpulkan dan mempelajari referensi yang berkaitan dengan sistem monitoring *real-time*, manajemen log akses pintu, *web service*, *decoupled architecture*, Flask, React JS, Supabase PostgreSQL, Virtual Private Server, serta keamanan aplikasi web. Selain itu, studi literatur juga mencakup pembahasan mengenai pengujian fungsional berbasis *test script*, OWASP Top Ten, dan pengujian performa sistem.

2. Perancangan Sistem

Tahap perancangan sistem dilakukan untuk menyusun rancangan arsitektur, alur kerja sistem, rancangan antarmuka, struktur basis data, serta mekanisme komunikasi antar komponen. Sistem dirancang menggunakan konsep *decoupled architecture* dengan memisahkan komponen *backend* Flask, *frontend* React JS, basis data Supabase PostgreSQL, dan layanan pada Virtual Private Server.

3. Implementasi Sistem

Tahap implementasi sistem dilakukan dengan membangun sistem berdasarkan rancangan yang telah dibuat. Implementasi meliputi pengembangan *backend web service* menggunakan Flask, pembuatan antarmuka website menggunakan React JS, konfigurasi basis data Supabase PostgreSQL, konfigurasi VPS, serta penerapan mekanisme keamanan seperti API Key, JSON Web Token, CAPTCHA, *rate limiting*, *parameterized query*, dan Row Level Security.

4. Pengujian dan Evaluasi

Tahap pengujian dan evaluasi dilakukan untuk memastikan sistem berjalan sesuai kebutuhan. Pengujian yang dilakukan meliputi pengujian fungsional website menggunakan *test script*, pengujian keamanan berdasarkan kategori OWASP Top Ten, serta pengujian kualitas website menggunakan Lighthouse. Pengujian fungsional digunakan untuk memastikan fitur utama website berjalan sesuai skenario, pengujian keamanan digunakan untuk mengevaluasi risiko pada layanan *web service*, sedangkan pengujian



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

kualitas website digunakan untuk menilai aspek *performance*, *accessibility*, *best practices*, dan SEO pada halaman website monitoring.

5. Analisis Hasil

Tahap analisis hasil dilakukan dengan meninjau dan mengolah data hasil pengujian untuk mengetahui tingkat keberhasilan sistem dari sisi fungsionalitas, keamanan, dan kualitas website. Hasil analisis digunakan untuk mengevaluasi apakah sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service* telah berjalan sesuai tujuan penelitian, memiliki mekanisme keamanan yang memadai, serta memiliki kualitas halaman website yang layak digunakan oleh administrator.

6. Penyusunan Laporan

Tahap penyusunan laporan dilakukan dengan menyusun seluruh hasil penelitian ke dalam bentuk laporan skripsi. Laporan disusun mulai dari pendahuluan, tinjauan pustaka, metode penelitian, hasil dan pembahasan, hingga kesimpulan dan saran berdasarkan hasil implementasi serta pengujian sistem.

### 3.3 Objek Penelitian

Merujuk pada fokus perancangan yang dilakukan, objek penelitian ini adalah produk rekayasa teknologi informasi berupa sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service*. Secara spesifik, objek yang diteliti mencakup arsitektur perangkat lunak yang terdiri dari *backend* Flask yang berjalan pada Virtual Private Server (VPS), basis data PostgreSQL yang dikelola melalui platform Supabase, serta antarmuka *dashboard* administrator berbasis React JS. Sistem ini digunakan untuk menerima, mengelola, menyimpan, dan menampilkan data log akses pintu secara terpusat dan *real-time*. Objek penelitian tersebut dievaluasi melalui tiga aspek utama, yaitu pengujian fungsional website menggunakan *test script*, pengujian keamanan berdasarkan kategori OWASP Top Ten, serta pengujian kualitas website menggunakan Lighthouse pada aspek *performance*, *accessibility*, *best practices*, dan SEO. Dengan demikian, penelitian ini tidak hanya meninjau keberhasilan implementasi sistem, tetapi juga menilai kesesuaian fungsi, ketahanan keamanan, dan kualitas halaman website yang digunakan oleh administrator.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi komponen, fungsi, dan standar kualitas yang diperlukan dalam pengembangan sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service*. Tahapan ini menjadi dasar dalam menentukan perangkat lunak yang digunakan, fitur utama yang harus tersedia, serta kebutuhan non-fungsional yang berkaitan dengan keamanan dan kualitas halaman website. Pada penelitian ini, analisis kebutuhan disusun berdasarkan konsep *decoupled architecture*, yaitu pemisahan antara *backend* Flask, *frontend* React JS, basis data Supabase PostgreSQL, dan Virtual Private Server sebagai lingkungan *deployment*. Selain itu, analisis kebutuhan juga disesuaikan dengan metode pengujian yang digunakan, yaitu pengujian fungsional berbasis *test script*, pengujian keamanan berdasarkan OWASP Top Ten, dan pengujian kualitas website menggunakan Lighthouse.

##### 4.1.1 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak merupakan daftar aplikasi, *framework*, *library*, dan alat pendukung yang digunakan dalam proses perancangan, implementasi, serta pengujian sistem. Perangkat lunak yang digunakan tidak hanya mendukung pembangunan sistem, tetapi juga mendukung proses evaluasi dari sisi fungsionalitas, keamanan, dan performa. Rincian kebutuhan perangkat lunak ditunjukkan pada Tabel 4.1.

Tabel 4. 1 Kebutuhan Perangkat Lunak

| No | Perangkat Lunak | Fungsi dalam Sistem                                                                                                                |
|----|-----------------|------------------------------------------------------------------------------------------------------------------------------------|
| 1  | <i>Python</i>   | Digunakan sebagai bahasa pemrograman utama untuk membangun <i>backend web service</i> .                                            |
| 2  | <i>Flask</i>    | Digunakan sebagai <i>framework backend</i> untuk membuat endpoint REST API, mengelola request, dan memproses data log akses pintu. |



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|    |                               |                                                                                                                                            |
|----|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 3  | <i>React JS</i>               | Digunakan untuk membangun antarmuka <i>dashboard</i> monitoring berbasis web yang menampilkan data log akses secara <i>real-time</i> .     |
| 4  | <i>Node.js</i> dan <i>NPM</i> | Digunakan sebagai lingkungan pengembangan frontend serta pengelola dependensi <i>React JS</i> .                                            |
| 5  | Supabase                      | Digunakan sebagai layanan <i>backend-as-a-service</i> untuk mengelola koneksi basis data, autentikasi, dan integrasi data berbasis cloud.  |
| 6  | PostgreSQL                    | Digunakan sebagai sistem manajemen basis data relasional untuk menyimpan data <i>access_logs</i> secara terstruktur.                       |
| 7  | Visual Studio Code            | Digunakan sebagai code editor dalam proses penulisan dan pengelolaan source code <i>backend</i> maupun <i>frontend</i> .                   |
| 8  | Postman                       | Digunakan untuk menguji endpoint API, validasi API Key, serta memastikan <i>respons server</i> sesuai dengan skenario pengujian.           |
| 9  | Kali Linux                    | Digunakan sebagai sistem operasi pengujian keamanan untuk menjalankan <i>tools penetration testing</i> seperti SQLMap dan Hydra.           |
| 10 | SQLMap                        | Digunakan sebagai alat bantu pengujian keamanan untuk menguji potensi kerentanan <i>SQL Injection</i> pada endpoint sistem.                |
| 11 | Hydra                         | Digunakan sebagai alat bantu pengujian keamanan untuk mensimulasikan serangan <i>brute force</i> pada halaman <i>login administrator</i> . |
| 12 | Ubuntu Server 22.04 LTS       | Digunakan sebagai sistem operasi pada <i>Virtual Private Server</i> (VPS) untuk menjalankan layanan <i>backend</i> .                       |
| 13 | Git                           | Digunakan untuk membantu pengelolaan <i>versi source code</i> selama proses pengembangan sistem.                                           |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|    |                                            |                                                                                                                                                                                     |
|----|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14 | Microsoft Excel /<br>Google<br>Spreadsheet | Digunakan untuk menyusun dan mendokumentasikan pengujian fungsional berbasis <i>test script</i> , meliputi skenario uji, hasil yang diharapkan, hasil aktual, dan status pengujian. |
| 15 | Web Browser                                | Digunakan untuk menjalankan dan memeriksa tampilan website monitoring serta memastikan fitur website dapat diakses oleh Administrator.                                              |
| 16 | Chrome<br>DevTools                         | Digunakan untuk memeriksa respons halaman, aktivitas jaringan, penyimpanan token, serta menjalankan audit kualitas website.                                                         |
| 17 | Lighthouse                                 | Digunakan untuk menguji kualitas halaman website berdasarkan aspek <i>performance</i> , <i>accessibility</i> , <i>best practices</i> , dan SEO.                                     |
| 18 | npm audit                                  | Digunakan untuk memeriksa potensi kerentanan dependensi pada <i>frontend</i> React JS sebagai bagian dari pengujian keamanan dependensi.                                            |
| 19 | pip-audit / Safety                         | Digunakan untuk memeriksa potensi kerentanan dependensi pada <i>backend</i> Python/Flask sebagai bagian dari pengujian keamanan dependensi.                                         |

#### 4.1.2 Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan kemampuan utama yang harus dimiliki oleh sistem agar proses monitoring dan manajemen log akses pintu dapat berjalan sesuai rancangan. Pada sistem ini, kebutuhan fungsional berfokus pada kemampuan sistem dalam menerima, memvalidasi, menyimpan, menampilkan, dan mengelola data log akses pintu secara terpusat. Kebutuhan fungsional ini juga menjadi dasar dalam penyusunan pengujian fungsional berbasis *test script*. Rincian kebutuhan fungsional ditunjukkan pada Tabel 4.2.

Tabel 4. 2 Kebutuhan Fungsional Sistem

| No | Modul Sistem                   | Deskripsi Kebutuhan Fungsional                                                                                                                                                |
|----|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Login Administrator            | Sistem harus menyediakan halaman login agar administrator dapat masuk ke dalam sistem menggunakan kredensial yang valid.                                                      |
| 2  | Validasi CAPTCHA               | Sistem harus memvalidasi CAPTCHA pada halaman login untuk membedakan akses manusia dengan percobaan akses otomatis.                                                           |
| 3  | Dashboard Monitoring           | Sistem harus mampu menampilkan ringkasan data log akses pintu secara terstruktur melalui halaman dashboard.                                                                   |
| 4  | Logging dan Auditing           | Sistem harus mampu menangkap dan merekam aktivitas akses pintu secara <i>real-time</i> , meliputi identitas pengguna, status akses, waktu kejadian, dan keterangan aktivitas. |
| 5  | Secure REST API                | Sistem harus menyediakan endpoint API yang dapat digunakan untuk menerima data log akses pintu secara aman melalui validasi API Key atau token.                               |
| 6  | Manajemen Log Akses            | Sistem harus mampu menampilkan data log akses pintu yang tersimpan pada basis data agar dapat dipantau oleh administrator.                                                    |
| 7  | Pencarian atau Penyaringan Log | Sistem harus menyediakan fitur pencarian atau penyaringan data log agar administrator dapat menelusuri data berdasarkan parameter tertentu.                                   |
| 8  | Daftar Wajah Pengguna          | Sistem harus mampu menampilkan data daftar wajah pengguna yang terintegrasi dengan sistem utama.                                                                              |
| 9  | Kelola Pengguna                | Sistem harus menyediakan fitur pengelolaan data pengguna yang berkaitan dengan akses pintu.                                                                                   |
| 10 | Logout Administrator           | Sistem harus menyediakan fungsi logout untuk mengakhiri sesi administrator dan mengembalikan pengguna ke halaman login.                                                       |

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|    |                                |                                                                                                                                                                        |
|----|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11 | Access Control                 | Sistem harus membatasi akses halaman dan endpoint tertentu agar hanya administrator yang telah terautentikasi yang dapat mengakses data sistem.                        |
| 12 | Deteksi Aktivitas Mencurigakan | Sistem harus mampu merespons aktivitas mencurigakan, seperti akses tanpa token, API Key tidak valid, login gagal berulang, dan input yang berpotensi menyerang sistem. |

#### 4.1.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional menjelaskan standar kualitas yang harus dipenuhi agar sistem dapat berjalan secara aman, stabil, responsif, dan dapat dievaluasi secara terukur. Pada penelitian ini, kebutuhan non-fungsional tidak hanya mencakup keamanan sistem, tetapi juga performa layanan dan penggunaan sumber daya VPS. Rincian kebutuhan non-fungsional ditunjukkan pada Tabel 4.3.

Tabel 4. 3 Kebutuhan Non-Fungsional Sistem

| No | Komponen Kebutuhan   | Spesifikasi Teknis                                                                                              | Target Pengujian                                                                                |
|----|----------------------|-----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| 1  | Keamanan             | Sistem menerapkan API Key, JWT, CAPTCHA, <i>rate limiting, parameterized query</i> , dan Row Level Security.    | Endpoint API, login Administrator, data log akses, dan basis data.                              |
| 2  | Validitas Fungsional | Sistem diuji menggunakan <i>test script</i> untuk memastikan fitur utama berjalan sesuai hasil yang diharapkan. | Login, dashboard, log akses, daftar wajah pengguna, kelola pengguna, pencarian log, dan logout. |



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|   |                        |                                                                                                                 |                                                                                                                    |
|---|------------------------|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| 3 | Access Control         | Sistem membatasi akses halaman dan endpoint berdasarkan token atau otorisasi yang valid.                        | Endpoint dashboard, endpoint log akses, dan halaman administrator.                                                 |
| 4 | Authentication         | Sistem mewajibkan administrator melakukan login dan validasi CAPTCHA sebelum mengakses halaman utama.           | Halaman login administrator dan sesi pengguna.                                                                     |
| 5 | Integrity              | Sistem menggunakan <i>parameterized query</i> dan Row Level Security untuk menjaga integritas data log akses.   | Data log akses pada PostgreSQL/Supabase.                                                                           |
| 6 | Availability           | Sistem menerapkan <i>rate limiting</i> untuk mengurangi risiko permintaan berulang yang dapat membebani server. | Endpoint login dan endpoint API.                                                                                   |
| 7 | OWASP Security Testing | Sistem diuji berdasarkan kategori OWASP Top Ten yang relevan dengan aplikasi <i>web service</i> .               | Kontrol akses, konfigurasi keamanan, autentikasi, <i>injection</i> , integritas data, logging, dan error handling. |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|   |                  |                                                                                                                                    |                                                                                 |
|---|------------------|------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| 8 | Kualitas Website | Sistem diuji menggunakan Lighthouse berdasarkan aspek <i>performance</i> , <i>accessibility</i> , <i>best practices</i> , dan SEO. | Halaman website monitoring dan manajemen log akses pintu.                       |
| 9 | Error Handling   | Sistem memberikan respons kesalahan yang aman tanpa menampilkan detail teknis yang sensitif.                                       | Respons API saat menerima data tidak valid, method salah, atau token tidak sah. |

#### 4.2 Perancangan Sistem

Perancangan sistem merupakan tahap lanjutan setelah analisis kebutuhan dilakukan. Pada tahap ini, kebutuhan perangkat lunak, kebutuhan fungsional, dan kebutuhan non-fungsional diterjemahkan ke dalam rancangan teknis yang menjadi dasar dalam proses implementasi. Perancangan sistem bertujuan untuk menggambarkan hubungan antar komponen, alur komunikasi data, batas akses pengguna, serta mekanisme keamanan yang diterapkan pada sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service*.

Sistem yang dirancang menggunakan pendekatan *decoupled architecture*, yaitu pemisahan antara *frontend*, *backend*, basis data, dan layanan pendukung lainnya. *Frontend* berfungsi sebagai antarmuka *dashboard monitoring* yang digunakan oleh administrator, *backend* berperan sebagai pusat pemrosesan *request* dan validasi keamanan, sedangkan basis data digunakan untuk menyimpan seluruh data log akses pintu secara terpusat. Dengan pemisahan tersebut, setiap komponen dapat dikembangkan, diuji, dan dievaluasi secara lebih terarah, baik dari sisi fungsionalitas, keamanan, maupun performa sistem.



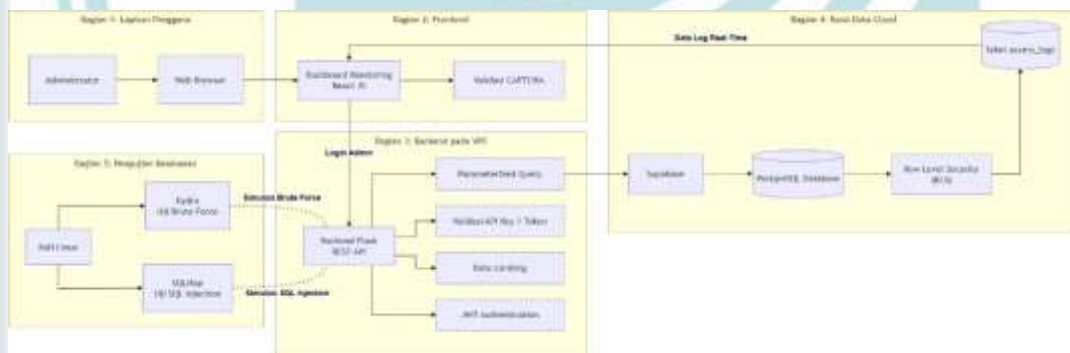
#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Perancangan pada penelitian ini terdiri dari perancangan arsitektur sistem, perancangan *use case diagram*, dan perancangan *flowchart* sistem. Ketiga rancangan tersebut digunakan untuk memberikan gambaran menyeluruh mengenai bagaimana sistem bekerja, bagaimana administrator berinteraksi dengan aplikasi, serta bagaimana data log akses pintu diproses dari sistem utama hingga ditampilkan pada *dashboard monitoring*.

#### 4.2.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem bertujuan untuk menggambarkan susunan komponen utama yang digunakan dalam sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service*. Arsitektur ini dirancang dengan pendekatan *decoupled architecture*, yaitu pemisahan antara lapisan pengguna, antarmuka *frontend*, layanan *backend*, basis data, dan lingkungan pengujian. Pemisahan tersebut dilakukan agar setiap komponen memiliki fungsi yang jelas dan dapat diuji secara terpisah sesuai kebutuhan sistem.



Gambar 4. 1 Arsitektur Sistem

Berdasarkan Gambar 4.1, sistem terdiri dari beberapa komponen utama. Administrator berperan sebagai pengguna utama yang mengakses sistem melalui *web browser*. Permintaan dari administrator diteruskan ke *frontend* React JS sebagai antarmuka sistem, kemudian berkomunikasi dengan *backend* Flask melalui REST API. *Backend* berfungsi untuk memproses permintaan, melakukan validasi akses, mengelola autentikasi, serta menghubungkan sistem dengan basis data Supabase PostgreSQL. Data log akses pintu yang diterima dari sistem utama disimpan secara terpusat pada basis data, kemudian ditampilkan kembali pada *dashboard monitoring* agar dapat dipantau oleh administrator.



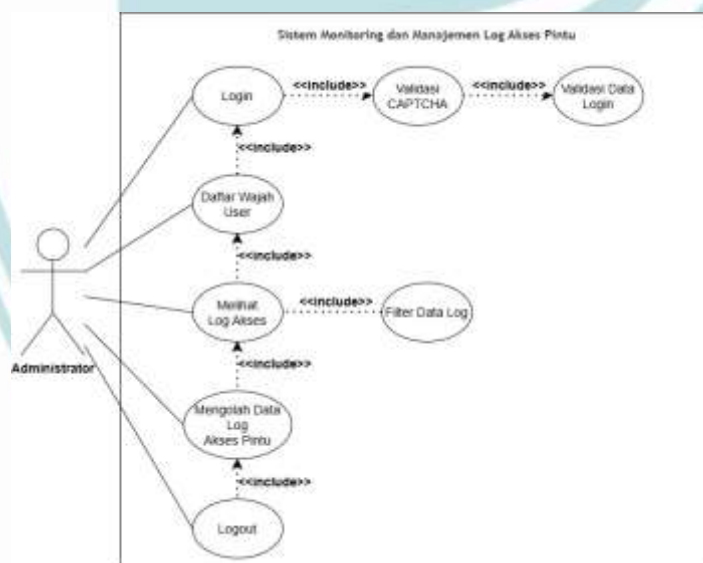
#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pada rancangan ini, mekanisme keamanan diterapkan pada beberapa titik penting, seperti penggunaan API Key untuk validasi akses dari sistem utama, JSON Web Token untuk sesi administrator, CAPTCHA dan *rate limiting* pada proses login, *parameterized query* untuk mengurangi risiko *injection*, serta Row Level Security untuk membatasi akses data pada basis data. Selain itu, arsitektur sistem juga dirancang agar dapat mendukung proses pengujian fungsional berbasis *test script*, pengujian keamanan berdasarkan OWASP Top Ten, dan pengujian performa kuantitatif terhadap layanan *web service*.

#### 4.2.2 Perancangan Use Case Diagram

Perancangan *use case diagram* digunakan untuk menggambarkan interaksi antara pengguna dengan fitur-fitur utama yang tersedia pada sistem. Pada sistem ini, aktor utama yang terlibat adalah Administrator. Administrator memiliki hak akses untuk melakukan login, melihat *dashboard monitoring*, melihat daftar wajah pengguna, mengelola data pengguna, melihat log akses pintu, melakukan pencarian atau penyaringan data log, serta melakukan logout dari sistem.



Gambar 4. 2 Use Case Diagram Sistem *Monitoring dan Manajemen Log Akses Pintu*.

Berdasarkan Gambar 4.2, seluruh fitur utama pada sistem hanya dapat diakses oleh Administrator setelah berhasil melakukan proses autentikasi. Proses login menjadi tahap awal sebelum pengguna dapat masuk ke halaman utama sistem. Setelah

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

berhasil masuk, Administrator dapat memantau data log akses pintu, melihat informasi pengguna, serta mengelola data yang berkaitan dengan sistem monitoring. Rancangan *use case* ini juga menjadi dasar dalam penyusunan pengujian fungsional berbasis *test script*, karena setiap fitur yang ditampilkan pada diagram dapat dijadikan sebagai skenario pengujian untuk memastikan sistem berjalan sesuai kebutuhan.

#### 4.2.3 Perancangan *Flowchart* Sistem

Perancangan *flowchart* sistem digunakan untuk menggambarkan alur kerja sistem secara berurutan, mulai dari administrator mengakses halaman login hingga sistem menampilkan data log akses pintu pada *dashboard monitoring*. Alur ini membantu menjelaskan proses autentikasi, validasi akses, pengambilan data, penyimpanan log, serta respons sistem terhadap kondisi akses yang sah maupun tidak sah.



POLITEKNIK  
NEGERI  
JAKARTA

- [illegible]

Berdasarkan Gambar 4.3, proses dimulai ketika administrator membuka halaman website dan melakukan login dengan memasukkan kredensial serta validasi CAPTCHA. Sistem kemudian mengirimkan data login ke *backend* untuk dilakukan pemeriksaan. Apabila data login valid, sistem akan menghasilkan token autentikasi dan mengarahkan administrator ke halaman *dashboard monitoring*. Apabila login tidak valid, sistem akan menolak akses dan menampilkan pesan kesalahan. Pada sisi lain, data log akses pintu dari sistem utama dikirimkan ke *backend* melalui endpoint API yang telah dilengkapi validasi API Key. Data yang valid akan diproses dan disimpan ke dalam basis data Supabase PostgreSQL, kemudian ditampilkan pada halaman log akses pintu.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Alur sistem juga mencakup mekanisme keamanan untuk menolak akses tanpa otorisasi, membatasi percobaan login berulang melalui *rate limiting*, serta menangani permintaan yang tidak sesuai agar tidak membocorkan informasi sensitif. Dengan rancangan alur tersebut, sistem tidak hanya dirancang untuk menjalankan fungsi monitoring dan manajemen log, tetapi juga mendukung kebutuhan pengujian keamanan berdasarkan OWASP Top Ten serta pengujian kualitas website menggunakan Lighthouse.

### 4.3 Implementasi Sistem

Implementasi sistem merupakan tahap penerapan dari rancangan yang telah dibuat sebelumnya ke dalam bentuk aplikasi yang dapat dijalankan dan diuji. Pada tahap ini, komponen sistem yang telah dirancang, seperti *backend*, *frontend*, basis data, serta infrastruktur server, mulai dikonfigurasi dan diintegrasikan agar membentuk sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service*. Sistem ini dibangun menggunakan pendekatan *decoupled architecture*, sehingga setiap komponen dikembangkan secara terpisah sesuai dengan fungsinya. *Backend* dikembangkan menggunakan *framework* Flask, *frontend* dibangun menggunakan React JS, basis data dikelola melalui Supabase PostgreSQL, sedangkan proses *deployment* dilakukan pada Virtual Private Server (VPS). Selain itu, sistem juga dilengkapi dengan mekanisme keamanan seperti API Key, JSON Web Token (JWT), CAPTCHA, *rate limiting*, *parameterized query*, dan Row Level Security untuk mendukung perlindungan akses dan menjaga integritas data log akses pintu.

#### 4.3.1 Implementasi Lingkungan *Backend* Flask

Implementasi lingkungan *backend* Flask dilakukan untuk menyiapkan sisi server yang berfungsi sebagai pusat pemrosesan *request* pada sistem monitoring *real-time* dan manajemen log akses pintu. *Backend* dikembangkan menggunakan bahasa pemrograman Python dengan *framework* Flask. Pada sistem ini, *backend* digunakan untuk menangani autentikasi Administrator, validasi token, komunikasi dengan *frontend*, pengambilan data dari Supabase PostgreSQL, pengelolaan data pengguna, serta pengamanan endpoint melalui beberapa *library* pendukung.



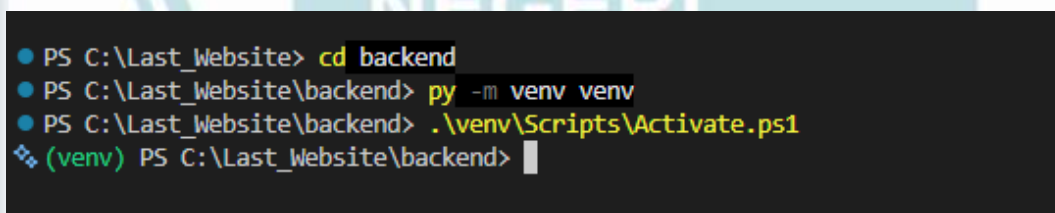
#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4. 4 Struktur Direktori *Backend Flask*

Berdasarkan Gambar 4.4, struktur direktori *backend Flask* disusun agar setiap komponen program tersimpan secara terpisah dan lebih mudah dikelola. Direktori *backend* memuat file utama aplikasi, konfigurasi *environment*, *virtual environment*, serta file pendukung lain yang digunakan untuk menjalankan layanan *backend*. Penyusunan struktur direktori ini bertujuan agar proses pengembangan, konfigurasi, dan pemeliharaan sistem dapat dilakukan secara lebih teratur. Selain itu, pemisahan file konfigurasi juga membantu menjaga data sensitif seperti API Key, token, dan konfigurasi koneksi basis data agar tidak ditulis secara langsung pada *source code* utama.



Gambar 4. 5 Pembuatan *Virtual Environment Backend*

Gambar 4.5 menunjukkan proses pembuatan *virtual environment* pada lingkungan *backend*. *Virtual environment* digunakan untuk memisahkan dependensi proyek dari konfigurasi Python utama pada sistem operasi. Dengan adanya *virtual environment*, seluruh *library* yang digunakan oleh aplikasi Flask dapat dikelola secara khusus pada lingkungan proyek, sehingga mengurangi risiko konflik versi dengan proyek atau aplikasi lain yang berada pada server.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
(.venv) PS C:\Last_Website\backend> python -c "import flask; print(flask.__version__)"
<string>:1: DeprecationWarning: The '__version__' attribute is deprecated and will be removed
in Flask 3.2. Use feature detection or 'importlib.metadata.version("flask")' instead.
3.1.3
```

Gambar 4. 6 Hasil Instalasi Dependensi *Backend Flask*

Berdasarkan Gambar 4.6, dependensi *backend* Flask telah berhasil diinstal. Dependensi tersebut digunakan untuk mendukung fungsi utama sistem, seperti pembuatan REST API, komunikasi dengan Supabase, pengelolaan autentikasi JWT, validasi *request*, serta penerapan mekanisme keamanan pada endpoint. Setelah struktur direktori dibuat, *virtual environment* diaktifkan, dan seluruh dependensi berhasil diinstal, lingkungan *backend* Flask dinyatakan siap digunakan untuk tahap implementasi fungsi sistem. Lingkungan ini menjadi dasar bagi pengembangan endpoint yang digunakan pada proses login Administrator, validasi CAPTCHA, pembuatan JWT, pengambilan data *dashboard*, pengelolaan data pengguna, dan pengambilan riwayat log akses pintu.

#### 4.3.2 Implementasi Lingkungan *Frontend* React JS

Implementasi lingkungan *frontend* React JS dilakukan untuk membangun antarmuka web yang digunakan oleh Administrator dalam mengakses sistem monitoring *real-time* dan manajemen log akses pintu. *Frontend* berfungsi sebagai lapisan tampilan yang menghubungkan pengguna dengan layanan *backend* Flask melalui REST API. Pada tahap ini, *frontend* dikembangkan menggunakan React JS karena mampu membangun antarmuka yang responsif, terstruktur, dan dapat berkomunikasi dengan *backend* secara dinamis.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

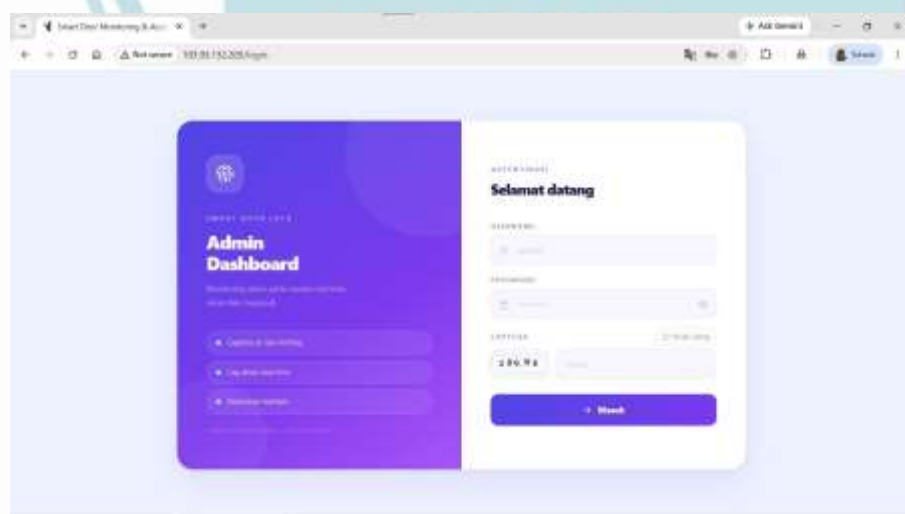
```

(.venv) PS C:\Last_Website\frontend> npm fund
frontend@0.0.0
├─┬ https://opencollective.com/postcss/
│   └─┬ autoprefixer@10.5.0, postcss@8.5.15
│       └─┬ https://opencollective.com/browserslist
│           └─┬ browserslist@4.28.2, update-browserslist-db@1.2.3, caniuse-lite@1.0.30001799
│               └─┬ https://github.com/sponsors/rnwify
│                   └─┬ fraction.js@5.3.4
│                       └─┬ https://opencollective.com/react-hook-form
│                           └─┬ react-hook-form@7.80.0
│                               └─┬ https://github.com/vitejs/vite/sponsor=1
│                                   └─┬ vite@8.0.16
│                                       └─┬ https://opencollective.com/parcel
│                                           └─┬ lightningcss@1.32.0, lightningcss-win32-x64-msvc@1.32.0
│                                               └─┬ https://github.com/sponsors/jonshlinkert
│                                                   └─┬ picomatch@4.0.4
│                                                       └─┬ https://github.com/sponsors/SuperchupuDev
│                                                           └─┬ timmglubby@0.2.17
│                                                               └─┬ https://github.com/sponsors/Bashen
│                                                                   └─┬ @oxc-project/types@0.133.0
│                                                                       └─┬ https://github.com/sponsors/collinhacks
│                                                                           └─┬ zod@4.4.3
│                                                                               └─┬ https://github.com/sponsors/RubenWebbonghi
│                                                                                   └─┬ follow-redirects@1.16.0
│                                                                                       └─┬ https://github.com/sponsors/Ijhaeb
│                                                                                           └─┬ get-intrinsic@1.3.0, function-bind@1.1.2, gopd@1.2.0, has-symbol@1.1.0, has-tostringtag@1.0.2
│                                                                                               └─┬ https://opencollective.com/enpreis
│                                                                                                   └─┬ cookie@1.1.1
│                                                                                                       └─┬ https://github.com/sponsors/sindresorhus
│                                                                                                           └─┬ @alloc/quick-lru@5.2.0
│                                                                                                               └─┬ https://opencollective.com/webpack
│                                                                                                                   └─┬ tapable@2.3.3

```

Gambar 4. 7 Hasil Instalasi Dependensi *Frontend*

Berdasarkan Gambar 4.7, proses instalasi dependensi *frontend* telah berhasil dilakukan. Dependensi yang digunakan pada *frontend* berfungsi untuk mendukung proses pembangunan halaman web, pengelolaan komponen tampilan, serta komunikasi antara *frontend* dan *backend*. Dengan dependensi tersebut, sistem dapat menampilkan beberapa halaman utama, seperti halaman login Administrator, *dashboard monitoring*, daftar wajah pengguna, kelola pengguna, dan log akses pintu.



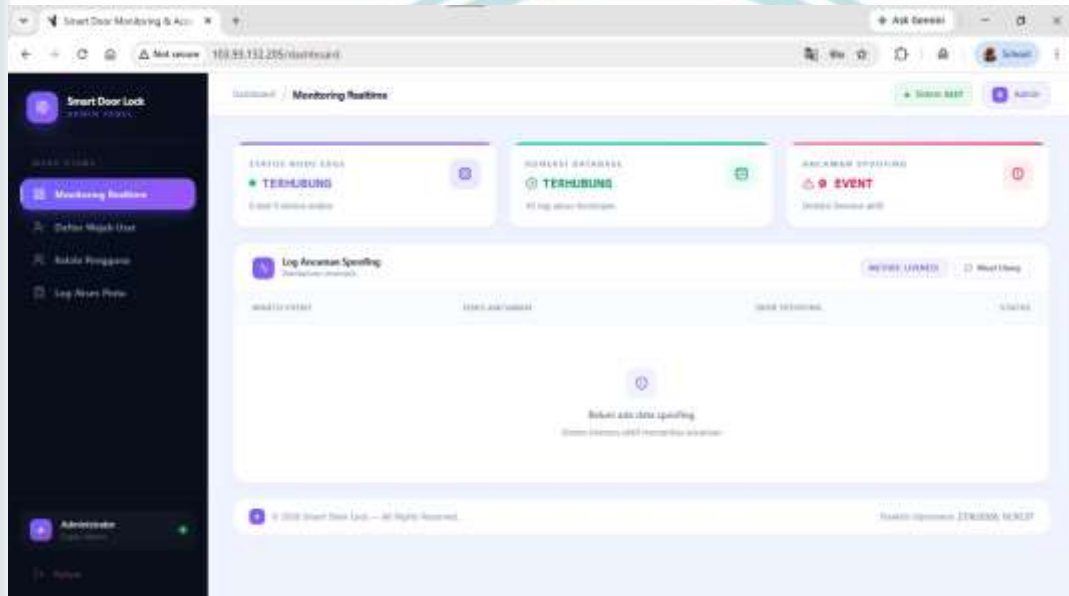
Gambar 4. 8 Tampilan Halaman Login Administrator



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.8 menunjukkan tampilan halaman login Administrator. Halaman ini menjadi pintu masuk utama sebelum pengguna dapat mengakses fitur sistem. Pada halaman login, Administrator diminta memasukkan kredensial yang valid dan menyelesaikan validasi CAPTCHA. Penerapan validasi pada halaman login bertujuan untuk memastikan bahwa akses ke sistem hanya dapat dilakukan oleh pengguna yang sah.



Gambar 4. 9 Tampilan *Dashboard Monitoring Realtime*

Gambar 4.9 menunjukkan tampilan *dashboard monitoring real-time* yang digunakan oleh Administrator untuk memantau kondisi sistem. Halaman ini menampilkan informasi utama yang berkaitan dengan status sistem, data akses, dan informasi pendukung lain yang diperoleh dari *backend* dan basis data Supabase. Dengan adanya halaman *dashboard*, Administrator dapat melihat kondisi sistem secara terpusat melalui antarmuka web.

#### 4.3.3 Konfigurasi Database Supabase

Konfigurasi *database* Supabase dilakukan untuk menyiapkan media penyimpanan data pada sistem monitoring *real-time* dan manajemen log akses pintu. Pada penelitian ini, Supabase digunakan sebagai layanan basis data berbasis *cloud* yang menyediakan PostgreSQL dan dapat diintegrasikan dengan *backend* Flask melalui API maupun *Software Development Kit* (SDK). Penggunaan Supabase bertujuan



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

agar data log akses pintu, data pengguna, dan data wajah yang telah terdaftar dapat disimpan secara terpusat serta dapat diakses oleh *backend* secara fleksibel.

```
SUPABASE_URL=https://gwxebdmavlmxxcdrlge.supabase.co
SUPABASE_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Imd3d3
```

Gambar 4. 10 Konfigurasi *Project URL* dan *API Key Supabase*

Berdasarkan Gambar 4.10, konfigurasi awal pada Supabase dilakukan dengan mengambil *Project URL* dan *API Key* yang dihasilkan oleh sistem. *Project URL* digunakan sebagai alamat utama untuk menghubungkan *backend* Flask dengan proyek Supabase, sedangkan *API Key* digunakan sebagai kredensial akses agar *backend* dapat melakukan komunikasi dengan basis data. Informasi tersebut menjadi bagian penting dalam proses integrasi karena *backend* membutuhkan konfigurasi koneksi yang valid untuk menjalankan proses baca dan tulis data pada tabel Supabase.

```
backend > cat .env
1 SECRET_KEY=smartdoor_super_rahasia_12345!
2 JWT_SECRET_KEY=jwt_smartdoor_amangila_67890@
3 FLASK_ENV=production
4 FLASK_APP=wsgi.py
5
6 DATABASE_URL=postgresql://postgres.gwxebdmavlmxxcdrlge:ocIbauketek123@aws-1-ap-sou
7 CORS_ORIGINS=*
8
9 REDIS_URL=redis://localhost:6379/0
10
11 ADMIN_USERNAME=admin
12 ADMIN_PASSWORD_HASH=scrypt:32768:B:1$epN21L2DsGLMwKs$8d20f3ee67d115841b7B141haF97
13
14 EDGE_BASE_URL=http://100.101.204.86:8000
15 DEVICE_NAME=raspberry_pi_01
16
17 DEVICE_SECRET_KEY=e9438d1214312f06b4a7fa31e5ed8dcad409cab70b3de38993cd15d896b00d9
18 DEVICE_TIMEOUT=10
19 DEVICE_POLL_INTERVAL=1
20
21 SUPABASE_URL=https://gwxebdmavlmxxcdrlge.supabase.co
22 SUPABASE_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6Imd3d3
```

Gambar 4. 11 Konfigurasi File *Environment Backend*

Gambar 4.11 menunjukkan konfigurasi file *environment* pada *backend* Flask. File *environment* digunakan untuk menyimpan variabel konfigurasi yang bersifat sensitif, seperti Supabase URL, Supabase Key, JWT Secret, dan token lain yang digunakan oleh sistem. Penempatan konfigurasi pada file *environment* bertujuan agar data rahasia tidak ditulis secara langsung di dalam *source code* utama. Dengan cara ini, sistem menjadi lebih aman dan lebih mudah dikonfigurasi apabila terjadi perubahan kredensial atau perpindahan lingkungan server.



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4. 12 Struktur Tabel pada Supabase

Gambar 4.12 menunjukkan struktur tabel yang digunakan pada Supabase. Pada sistem ini, terdapat beberapa tabel utama yang digunakan untuk menyimpan data, yaitu tabel *access\_logs*, *profiles*, dan *registered\_faces*. Tabel *access\_logs* digunakan untuk menyimpan riwayat akses pintu, seperti identitas pengguna, status akses, waktu akses, dan hasil deteksi dari sistem utama. Tabel *profiles* digunakan untuk menyimpan data profil pengguna, sedangkan tabel *registered\_faces* digunakan untuk menyimpan informasi wajah pengguna yang telah didaftarkan pada sistem.

#### 4.3.4 Implementasi Row Level Security pada Supabase

Implementasi *Row Level Security* dilakukan untuk memberikan pembatasan akses terhadap data pada tingkat baris di basis data *Supabase PostgreSQL*. Mekanisme ini digunakan agar data tidak dapat dibaca, ditambahkan, diperbarui, atau dimodifikasi secara bebas oleh pihak yang tidak memiliki otorisasi. Pada sistem monitoring *real-time* dan manajemen log akses pintu, Row Level Security menjadi salah satu lapisan keamanan pada sisi basis data, sehingga perlindungan data tidak hanya bergantung pada validasi di *backend*.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4. 13 Row Level Security pada Supabase

Gambar 4.13 menunjukkan bahwa fitur *Row Level Security* telah diaktifkan pada tabel *Supabase*. Pengaktifan RLS bertujuan untuk memastikan bahwa setiap proses akses data mengikuti aturan yang telah ditentukan oleh sistem. Dengan demikian, permintaan yang tidak sesuai dengan aturan akses tidak dapat langsung membaca atau mengubah data pada basis data.



Gambar 4. 14 Konfigurasi *Policy RLS* pada Tabel *Database*

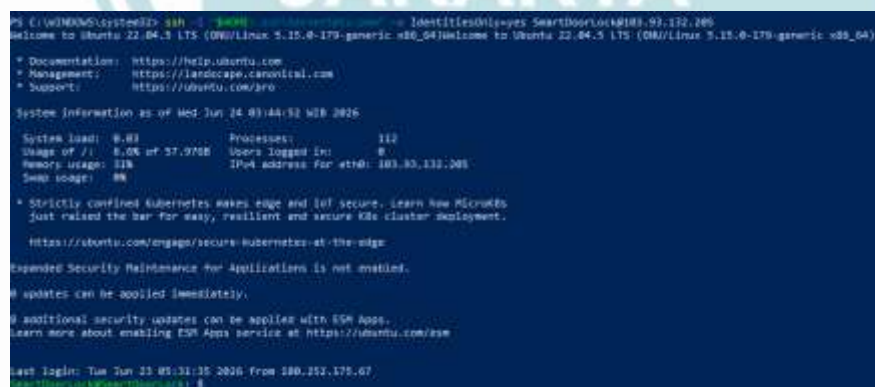
Gambar 4.14 menunjukkan konfigurasi *policy RLS* pada tabel *database Supabase*. *Policy* tersebut digunakan untuk menentukan aturan akses terhadap data, baik untuk proses membaca, menambahkan, maupun memperbarui data. Pada implementasi sistem ini, *policy* dibuat agar akses data hanya dapat dilakukan melalui mekanisme yang telah divalidasi oleh sistem. Dengan demikian, *request* yang tidak sesuai dengan aturan akses tidak akan diberikan izin untuk membaca atau memodifikasi data pada tabel. Penerapan RLS menjadi salah satu lapisan keamanan tambahan selain validasi API Key, JSON Web Token, dan pengamanan endpoint pada *backend* Flask. Kombinasi ini bertujuan untuk menjaga agar data log akses pintu tetap terlindungi dari akses langsung yang tidak sah maupun potensi manipulasi data.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Implementasi dan konfigurasi Virtual Private Server dilakukan untuk menempatkan sistem *backend* dan *frontend* pada lingkungan server yang dapat diakses secara daring. VPS digunakan sebagai tempat *deployment* aplikasi agar layanan *backend* Flask dan *frontend* React JS dapat berjalan secara terpusat. Pada tahap ini, konfigurasi dilakukan mulai dari pengecekan spesifikasi server, akses SSH, pembaruan paket sistem, instalasi dependensi, hingga proses menjalankan aplikasi sebagai layanan pada server.



Gambar 4.15 menunjukkan spesifikasi VPS yang digunakan sebagai lingkungan *deployment* sistem. Spesifikasi server menjadi bagian penting karena berpengaruh terhadap kemampuan sistem dalam menjalankan layanan *backend*, *frontend*, serta menangani *request* dari pengguna. Informasi spesifikasi VPS juga digunakan sebagai acuan dalam pengujian performa, terutama untuk mengamati penggunaan CPU dan *memory* selama sistem menerima beban permintaan.



Gambar 4. 16 Akses SSH ke VPS



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.16 menunjukkan proses akses ke VPS menggunakan SSH. Akses SSH digunakan untuk melakukan konfigurasi server, mengunggah atau mengambil *source code*, menjalankan perintah instalasi, serta mengelola layanan aplikasi. Melalui akses ini, pengembang dapat mengatur lingkungan server secara langsung sesuai kebutuhan sistem.

```
SmartDoorLock@SmartDoorLock: $ sudo apt update
Hit:1 http://mirror.biznetgio.com/ubuntu jammy InRelease
Hit:2 http://mirror.biznetgio.com/ubuntu jammy-updates InRelease
Hit:3 http://mirror.biznetgio.com/ubuntu jammy-backports InRelease
Hit:4 http://mirror.biznetgio.com/ubuntu jammy-security InRelease
Get:5 https://pkgs.tailscale.com/stable/ubuntu jammy InRelease
Hit:6 https://deb.nodesource.com/node_20.x nodistro InRelease
Fetched 6581 B in 1s (6700 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
SmartDoorLock@SmartDoorLock: $ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
  redis-server python3-pip-whl liblua5.1-0 lua-bitop redis-tools python3-pip
  lua-cjson
Learn more about Ubuntu Pro at https://ubuntu.com/pro
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
SmartDoorLock@SmartDoorLock: $ sudo apt install python3 python3-pip python3-venv git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.17).
python3 is already the newest version (3.10.6-1~22.04.1).
python3 set to manually installed.
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.7).
python3-venv is already the newest version (3.10.6-1~22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Gambar 4. 17 *Update Package dan Instalasi Dependency* pada VPS

Gambar 4.17 menunjukkan proses pembaruan paket sistem dan instalasi dependensi yang dibutuhkan pada VPS. Pembaruan paket dilakukan untuk memastikan lingkungan server berada dalam kondisi terbaru dan siap digunakan untuk menjalankan aplikasi. Setelah itu, dependensi pendukung diinstal agar server dapat menjalankan layanan *backend* dan *frontend* dengan baik.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
SmartDoorLock@SmartDoorLock:~$ curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
2023-04-24 03:07:30 - Installing pre-requisites

Hit:1 http://mirror.idnweb.id/ubuntu jammy InRelease
Hit:2 http://mirror.idnweb.id/ubuntu jammy-updates InRelease
Hit:3 http://mirror.idnweb.id/ubuntu jammy-backports InRelease
Hit:4 http://mirror.idnweb.id/ubuntu jammy-security InRelease
Get:5 https://ppa.launchpad.net/ansible/ubuntu-jammy InRelease
Hit:6 https://deb.nodesource.com/node_20.x nodistro InRelease
Fetched 6081 B in 1s (7674 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240226-02.04.1).
curl is already the newest version (7.81.0-1ubuntu1.24).
gnupg is already the newest version (2.2.37-1ubuntu1.3).
git is already the newest version (2.43.0-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Hit:1 http://mirror.idnweb.id/ubuntu jammy InRelease
Hit:2 http://mirror.idnweb.id/ubuntu jammy-updates InRelease
Hit:3 http://mirror.idnweb.id/ubuntu jammy-backports InRelease
Hit:4 http://mirror.idnweb.id/ubuntu jammy-security InRelease
Get:5 https://ppa.launchpad.net/ansible/ubuntu-jammy InRelease
Fetched 6081 B in 1s (7205 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.

2023-04-24 03:07:40 - To install Node.js, run: apt install nodejs -y
2023-04-24 03:07:40 - You can use Nixpacks Runtime as a node.js alternative
2023-04-24 03:07:40 - To install Nixpacks Runtime, run: apt install nixpacks -y

SmartDoorLock@SmartDoorLock:~$ sudo apt install -y nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nodejs is already the newest version (20.20.2-1nodesource).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
SmartDoorLock@SmartDoorLock:~$
```

Gambar 4. 18 Instalasi Python, pip, Node.js, NPM, dan Git pada VPS

Gambar 4.18 menunjukkan proses instalasi perangkat lunak utama yang dibutuhkan pada VPS, yaitu Python dan pip untuk menjalankan *backend* Flask, Node.js dan NPM untuk menjalankan *frontend* React JS, serta Git untuk membantu pengelolaan *source code*. Instalasi komponen tersebut menjadi dasar agar aplikasi dapat dijalankan pada lingkungan server.

```
SmartDoorLock@SmartDoorLock:~$ git clone https://github.com/kurniawanSandi23/smartdoor.git
fatal: destination path 'smartdoor' already exists and is not an empty directory.
SmartDoorLock@SmartDoorLock:~$ ls -la smartdoor
total 20
drwxr-xr-x 5 SmartDoorLock SmartDoorLock 4096 Jun 23 00:10 .
drwxr-xr-x 9 SmartDoorLock SmartDoorLock 4096 Jun 23 04:44 ..
drwxr-xr-x 8 SmartDoorLock SmartDoorLock 4096 Jun 23 00:10 .git
drwxr-xr-x 6 SmartDoorLock SmartDoorLock 4096 Jun 23 05:35 backend
drwxr-xr-x 6 SmartDoorLock SmartDoorLock 4096 Jun 23 03:17 frontend
```

Gambar 4. 19 Proses Clone atau Upload Source Code ke VPS

Gambar 4.19 menunjukkan proses pemindahan *source code* aplikasi ke VPS. Proses ini dilakukan agar kode program yang telah dikembangkan dapat ditempatkan pada lingkungan *deployment*. Setelah *source code* tersedia pada VPS, proses konfigurasi lanjutan dilakukan, seperti aktivasi *virtual environment* pada *backend*, instalasi *library* Flask, instalasi *package frontend*, serta penyesuaian konfigurasi koneksi ke Supabase.



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
SmartDoorLock@SmartDoorLock:~$ pm2 restart smartdoor-frontend
```

```
[PM2] Applying action restartProcessId on app [smartdoor-frontend](ids: [ 0 ])
[PM2] [smartdoor-frontend]:(0) B
```

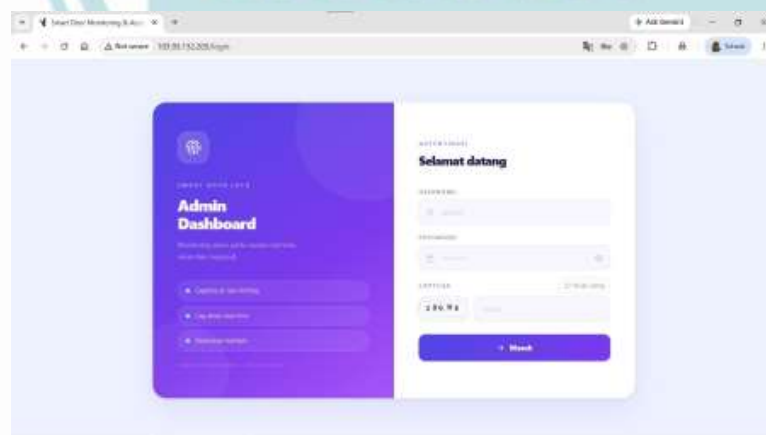
| id | name               | namespace | version | mode | pid   | uptime | #  | status | cpu  | mem     | user          | watching     |
|----|--------------------|-----------|---------|------|-------|--------|----|--------|------|---------|---------------|--------------|
| 1  | smartdoor-backend  | default   | N/A     | fork | 13479 | 1s     | 28 | online | 100% | 72.1mb  | SmartDoorLock | disconnected |
| 0  | smartdoor-frontend | default   | N/A     | fork | 13489 | 0s     | 15 | online | 0%   | 708.0kb | SmartDoorLock | disconnected |

Gambar 4. 20 Running Backend dan Frontend pada VPS

Gambar 4.20 menunjukkan bahwa *backend* Flask dan *frontend* React JS telah berhasil dijalankan pada VPS menggunakan PM2. Berdasarkan output PM2, proses *smartdoor-backend* dan *smartdoor-frontend* berada pada status *online*, sehingga kedua layanan telah aktif dan dapat berjalan sebagai proses latar belakang. Penggunaan PM2 bertujuan untuk mempermudah pengelolaan proses aplikasi, seperti memantau status layanan, melakukan *restart* aplikasi, serta menjaga layanan tetap berjalan secara lebih stabil pada lingkungan server. Setelah seluruh tahapan konfigurasi selesai dilakukan, VPS dinyatakan siap digunakan sebagai lingkungan *deployment* sistem.

#### 4.3.6 Hasil Implementasi Antarmuka Sistem

Hasil implementasi antarmuka sistem menunjukkan tampilan web yang digunakan oleh Administrator untuk mengakses fitur monitoring *real-time* dan manajemen log akses pintu. Antarmuka ini dibangun menggunakan React JS dan terhubung dengan *backend* Flask melalui REST API. Setiap halaman dirancang untuk mendukung kebutuhan Administrator dalam melakukan autentikasi, memantau kondisi sistem, mengelola data pengguna, melakukan registrasi wajah, serta melihat riwayat akses pintu.



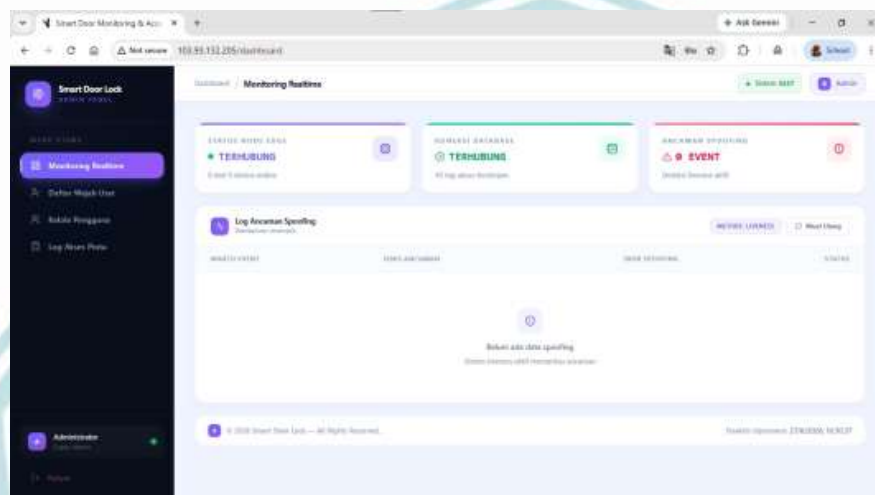
Gambar 4. 21 Tampilan Halaman Login Administrator



### Hak Cipta :

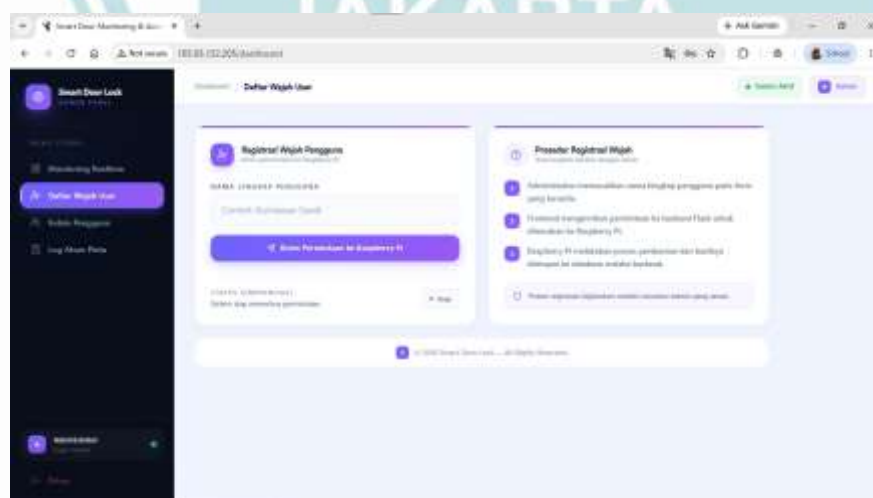
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.21 menunjukkan halaman login Administrator sebagai halaman awal sebelum pengguna dapat mengakses sistem. Pada halaman ini, Administrator memasukkan username, password, dan menyelesaikan validasi CAPTCHA. Jika data login sesuai, sistem akan memberikan akses menuju halaman utama. Jika data tidak sesuai, sistem akan menolak akses dan menampilkan pesan kesalahan.



Gambar 4. 22 Tampilan *Dashboard Monitoring Realtime*

Gambar 4.22 menunjukkan halaman *dashboard monitoring real-time*. Halaman ini digunakan untuk menampilkan informasi utama sistem, seperti status perangkat, status koneksi, serta ringkasan data yang berkaitan dengan aktivitas akses pintu. Informasi yang ditampilkan pada *dashboard* diperoleh dari *backend* Flask yang terhubung dengan Supabase PostgreSQL.



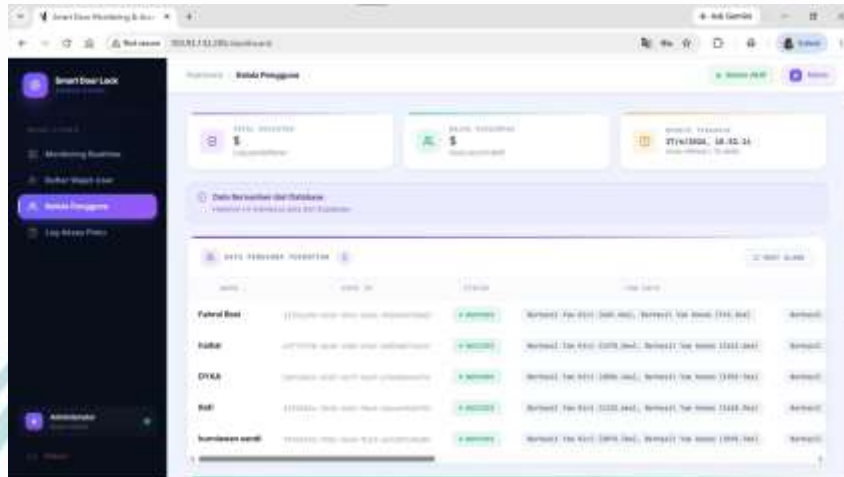
Gambar 4. 23 Tampilan Menu Daftar Wajah User



### Hak Cipta :

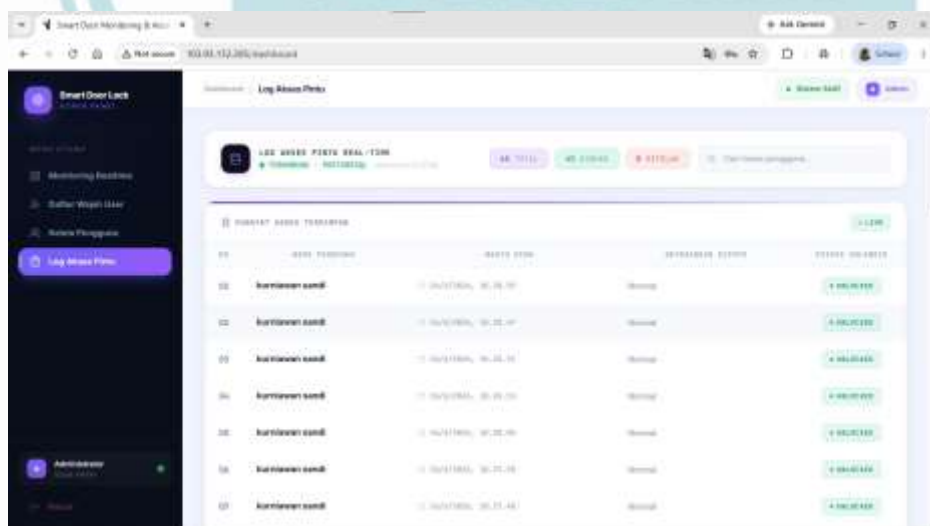
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.23 menunjukkan menu daftar wajah pengguna. Menu ini digunakan untuk membantu proses pendaftaran data wajah pengguna yang terhubung dengan sistem utama. Melalui halaman ini, Administrator dapat memasukkan data pengguna dan melakukan proses registrasi wajah sesuai kebutuhan sistem.



Gambar 4. 24 Tampilan Menu Kelola Pengguna

Gambar 4.24 menunjukkan menu kelola pengguna. Menu ini digunakan untuk menampilkan dan mengelola data pengguna yang tersimpan pada sistem. Data pengguna yang dikelola pada menu ini berkaitan dengan informasi profil pengguna dan data yang digunakan dalam proses akses pintu.



Gambar 4. 25 Tampilan Menu Log Akses Pintu



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.25 menunjukkan tampilan menu Log Akses Pintu. Menu ini digunakan untuk menampilkan riwayat akses pintu yang tersimpan pada *database*. Data yang ditampilkan mencakup identitas pengguna, status akses, waktu akses, serta hasil deteksi dari sistem utama. Selain menampilkan seluruh riwayat akses, halaman ini juga menyediakan fitur pencarian berdasarkan nama pengguna agar Administrator dapat menelusuri data log tertentu dengan lebih cepat. Berdasarkan hasil implementasi antarmuka yang telah dibuat, sistem berhasil menyediakan tampilan web yang mendukung kebutuhan Administrator dalam melakukan monitoring dan pengelolaan data akses pintu.

#### 4.3.7 Implementasi Mekanisme Keamanan Sistem

Implementasi mekanisme keamanan sistem dilakukan untuk melindungi layanan *backend*, *frontend*, dan basis data dari akses tidak sah maupun percobaan penyalahgunaan endpoint aplikasi. Pada sistem monitoring *real-time* dan manajemen log akses pintu ini, mekanisme keamanan yang diterapkan meliputi API Key untuk validasi akses dari sistem utama, JSON Web Token untuk pengelolaan sesi Administrator, CAPTCHA pada halaman login, *rate limiting* untuk membatasi percobaan login berulang, *parameterized query* untuk mengurangi risiko *injection*, serta Row Level Security untuk membatasi akses data pada Supabase PostgreSQL. Mekanisme keamanan ini menjadi dasar sebelum sistem diuji lebih lanjut menggunakan pendekatan OWASP Top Ten pada bagian pengujian.

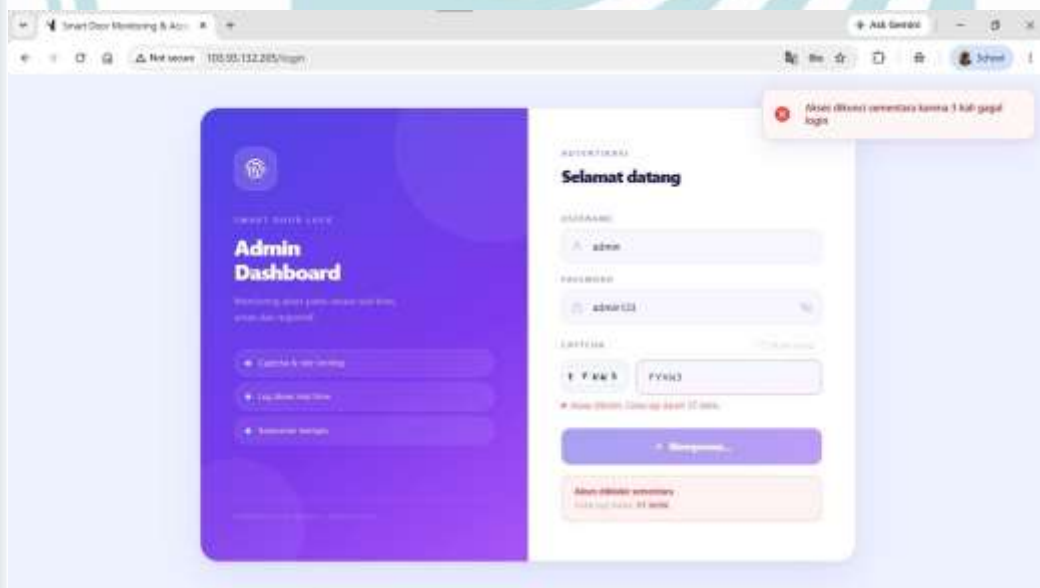
Gambar 4. 26 Implementasi CAPTCHA pada Halaman Login

Gambar 4.26 menunjukkan implementasi CAPTCHA pada halaman login Administrator. CAPTCHA digunakan sebagai validasi tambahan untuk memastikan bahwa proses login dilakukan oleh pengguna manusia, bukan oleh

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

[illegible]

Gambar 4.27 menunjukkan respons token JWT yang dihasilkan ketika proses login berhasil dilakukan. Token JWT digunakan sebagai identitas sesi Administrator setelah autentikasi berhasil. Token tersebut kemudian digunakan oleh *frontend* untuk mengakses endpoint yang membutuhkan otorisasi. Dengan adanya JWT, sistem dapat memastikan bahwa halaman dan data tertentu hanya dapat diakses oleh Administrator yang telah berhasil login.



Gambar 4.28 menunjukkan implementasi *rate limiting* pada halaman login. Mekanisme ini digunakan untuk membatasi jumlah percobaan login dalam rentang waktu tertentu. Pada sistem ini, apabila percobaan login gagal dilakukan secara berulang hingga melewati batas yang ditentukan, sistem akan memblokir akses sementara sebagai bentuk perlindungan terhadap percobaan login yang tidak wajar.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Penerapan *rate limiting* juga mendukung kebutuhan pengujian performa dan keamanan karena dapat digunakan untuk mengukur jumlah login gagal dan jumlah *request* yang diblokir.

Selain mekanisme tersebut, sistem juga menerapkan API Key pada endpoint tertentu agar permintaan dari sistem utama dapat divalidasi sebelum diproses oleh *backend*. Permintaan yang tidak memiliki API Key yang valid akan ditolak oleh sistem. Pada sisi basis data, Row Level Security diterapkan untuk membatasi akses data secara langsung, sedangkan *parameterized query* digunakan untuk memisahkan input pengguna dari perintah basis data. Dengan kombinasi mekanisme keamanan tersebut, sistem memiliki lapisan perlindungan pada sisi akses aplikasi, endpoint API, sesi pengguna, dan basis data. Hasil pengujian terhadap mekanisme keamanan ini tidak dibahas pada bagian implementasi, melainkan disajikan pada bagian 4.4 Pengujian Sistem agar sesuai dengan pendekatan pengujian berdasarkan OWASP Top Ten.

#### 4.4 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui apakah sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service* telah berjalan sesuai kebutuhan dari sisi fungsionalitas, keamanan, dan kualitas halaman website. Pengujian fungsional dilakukan menggunakan pendekatan *test script* untuk memastikan fitur utama website berjalan sesuai skenario yang telah ditentukan. Pengujian keamanan dilakukan berdasarkan kategori OWASP Top Ten untuk mengevaluasi potensi risiko keamanan pada layanan *web service*. Selain itu, pengujian kualitas website dilakukan menggunakan Lighthouse untuk menilai aspek *performance*, *accessibility*, *best practices*, dan SEO pada halaman website monitoring.

##### 4.4.1 Skenario Pengujian

Skenario pengujian disusun sebagai acuan dalam pelaksanaan pengujian sistem. Pada penelitian ini, skenario pengujian dibagi menjadi tiga bagian, yaitu pengujian fungsional berbasis *test script*, pengujian keamanan berdasarkan OWASP Top Ten, dan pengujian performa sistem.



Tabel 4. 4 Skenario Pengujian Fungsional Berbasis *Test Script*

| Test Case ID | Fitur yang Diuji      | Skenario Pengujian                                                    | Hasil yang Diharapkan                                          |
|--------------|-----------------------|-----------------------------------------------------------------------|----------------------------------------------------------------|
| TS-01        | Login Administrator   | Administrator memasukkan username, password, dan CAPTCHA yang valid.  | Sistem berhasil login dan menampilkan halaman dashboard.       |
| TS-02        | Login Gagal           | Administrator memasukkan username atau password yang tidak valid.     | Sistem menolak login dan menampilkan pesan kesalahan.          |
| TS-03        | Validasi CAPTCHA      | Administrator mengisi CAPTCHA yang salah atau kosong.                 | Sistem menolak login karena CAPTCHA tidak valid.               |
| TS-04        | Dashboard Monitoring  | Administrator membuka halaman dashboard setelah login.                | Sistem menampilkan informasi monitoring secara terpusat.       |
| TS-05        | Daftar Wajah User     | Administrator membuka menu daftar wajah user.                         | Sistem menampilkan data wajah pengguna yang terdaftar.         |
| TS-06        | Registrasi Wajah User | Administrator memasukkan data pengguna untuk proses registrasi wajah. | Sistem memproses data registrasi wajah sesuai input.           |
| TS-07        | Kelola Pengguna       | Administrator membuka menu kelola pengguna.                           | Sistem menampilkan data pengguna yang tersimpan pada database. |

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|       |                  |                                                                     |                                                            |
|-------|------------------|---------------------------------------------------------------------|------------------------------------------------------------|
| TS-08 | Log Akses Pintu  | Administrator membuka menu log akses pintu.                         | Sistem menampilkan riwayat akses pintu secara terstruktur. |
| TS-09 | Pencarian Log    | Administrator mencari log berdasarkan nama atau parameter tertentu. | Sistem menampilkan data log sesuai kata kunci pencarian.   |
| TS-10 | Proteksi Halaman | Pengguna mencoba mengakses halaman utama tanpa login.               | Sistem menolak akses dan mengarahkan ke halaman login.     |
| TS-11 | Logout           | Administrator menekan tombol logout.                                | Sistem mengakhiri sesi dan kembali ke halaman login.       |

Tabel 4. 5 Skenario Pengujian Keamanan Berdasarkan OWASP Top Ten

| Kode OWASP | Kategori                         | Skenario Pengujian                                                                                                                                              | Hasil yang Diharapkan                                                                                                               |
|------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| A01        | <i>Broken Access Control</i>     | Mengakses endpoint dashboard dan log akses menggunakan token valid, tanpa token, dan token yang salah.                                                          | Endpoint hanya dapat diakses dengan token valid, sedangkan request tanpa token atau token salah ditolak.                            |
| A02        | <i>Security Misconfiguration</i> | Mengakses endpoint yang tidak tersedia, menggunakan metode HTTP yang tidak sesuai, memanipulasi Origin, memeriksa header respons, dan meninjau alert OWASP ZAP. | Sistem memberikan respons error yang sesuai tanpa menampilkan traceback, source code, versi server detail, atau informasi sensitif. |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|     |                                       |                                                                                                                                            |                                                                                                                                                        |
|-----|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| A03 | <i>Software Supply Chain Failures</i> | Melakukan audit dependensi frontend menggunakan <i>npm audit</i> dan <i>backend</i> menggunakan <i>pip-audit</i> .                         | Hasil audit dependensi terdokumentasi dan tidak ditemukan kerentanan kritis pada dependensi yang digunakan.                                            |
| A04 | <i>Cryptographic Failures</i>         | Memeriksa isi JWT, mencoba mengakses file <i>.env</i> melalui browser, dan mencari keberadaan <i>secret key</i> pada source code frontend. | JWT tidak memuat data sensitif, file <i>.env</i> tidak dapat diakses melalui browser, dan <i>secret key</i> tidak ditemukan pada source code frontend. |
| A05 | <i>Injection</i>                      | Mengirim payload SQL Injection pada endpoint login serta melakukan pengujian tambahan menggunakan SQLMap.                                  | Sistem tidak mengeksekusi input sebagai perintah SQL dan tidak mengizinkan bypass autentikasi melalui payload berbahaya.                               |
| A06 | <i>Insecure Design</i>                | Mencoba mengakses API atau halaman dashboard secara langsung tanpa melalui proses login.                                                   | Sistem tetap melakukan validasi pada sisi <i>backend</i> dan menolak akses tanpa autentikasi.                                                          |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|     |                                               |                                                                                                                     |                                                                                                           |
|-----|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| A07 | <i>Authentication Failures</i>                | Menguji login dengan CAPTCHA salah, password salah, serta percobaan login gagal berulang.                           | Sistem menolak login yang tidak valid dan mengunci akses sementara setelah beberapa kali percobaan gagal. |
| A08 | <i>Software or Data Integrity Failures</i>    | Mencoba menghapus data log akses pintu melalui request <i>DELETE</i> pada endpoint log.                             | Sistem menolak request penghapusan log sehingga data log tidak dapat dihapus tanpa otorisasi.             |
| A09 | <i>Security Logging and Alerting Failures</i> | Memeriksa pencatatan aktivitas login gagal, CAPTCHA tidak sesuai, password salah, dan request yang tidak diizinkan. | Aktivitas penting tercatat pada log sistem sebagai bukti aktivitas keamanan.                              |
| A10 | <i>Mishandling of Exceptional Conditions</i>  | Menguji login dengan username/password salah, CAPTCHA salah, CAPTCHA kedaluwarsa, dan kondisi akses diblokir.       | Sistem menampilkan pesan kesalahan yang aman dan tidak membocorkan detail teknis internal.                |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Tabel 4. 6 Skenario Pengujian Kualitas Website Menggunakan Lighthouse

| Kode Uji | Jenis Pengujian       | Skenario Pengujian                                                 | Hasil yang Diharapkan                                                                            |
|----------|-----------------------|--------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| LH-01    | <i>Performance</i>    | Melakukan audit halaman website monitoring menggunakan Lighthouse. | Website memperoleh skor yang menunjukkan kualitas pemuatan halaman yang layak digunakan.         |
| LH-02    | <i>Accessibility</i>  | Melakukan audit aksesibilitas halaman website monitoring.          | Website memiliki struktur halaman, label, kontras, dan navigasi yang mendukung kemudahan akses.. |
| LH-03    | <i>Best Practices</i> | Melakukan audit praktik pengembangan website.                      | Website memenuhi praktik pengembangan web yang baik dari sisi teknis dan keamanan dasar.         |
| LH-04    | SEO                   | Melakukan audit SEO pada halaman website monitoring.               | Website memiliki struktur dasar halaman yang rapi dan mudah dianalisis.                          |

#### 4.4.2 Prosedur Pengujian

Prosedur pengujian dilakukan secara bertahap sesuai dengan jenis pengujian yang telah ditentukan. Tahap pertama adalah pengujian fungsional berbasis *test script*. Pada tahap ini, setiap fitur website diuji berdasarkan skenario yang telah disusun, meliputi login Administrator, validasi CAPTCHA, dashboard monitoring, daftar wajah user, registrasi wajah, kelola pengguna, log akses pintu, pencarian log, proteksi halaman tanpa login, dan logout. Setiap hasil pengujian dicatat dalam format *test script* yang memuat *test case ID*, skenario pengujian, hasil yang diharapkan, hasil aktual, status pengujian, tanggal eksekusi, dan nama penguji.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Tahap kedua adalah pengujian keamanan berdasarkan OWASP Top Ten. Pengujian dilakukan terhadap endpoint, token, API Key, halaman login, konfigurasi server, dependensi perangkat lunak, validasi input, integritas data log, pencatatan aktivitas keamanan, dan penanganan kesalahan. Alat bantu yang digunakan dalam pengujian ini meliputi Postman, SQLMap, Hydra, serta alat pendukung lain sesuai kebutuhan pengujian.

Tahap ketiga adalah pengujian kualitas website menggunakan Lighthouse. Pengujian ini dilakukan untuk menilai kualitas halaman website monitoring dan manajemen log akses pintu berdasarkan aspek *performance*, *accessibility*, *best practices*, dan SEO. Proses pengujian dilakukan dengan membuka halaman website melalui browser, kemudian menjalankan audit Lighthouse pada halaman yang diuji. Hasil audit berupa skor pada setiap aspek digunakan untuk mengetahui kualitas pemuatan halaman, kemudahan akses, penerapan praktik pengembangan web yang baik, serta kelengkapan struktur dasar SEO. Pengujian ini tidak diarahkan untuk mengukur kemampuan VPS dalam menangani banyak *request*, melainkan untuk mengevaluasi kualitas halaman website yang digunakan oleh administrator.

POLITEKNIK  
NEGERI  
JAKARTA

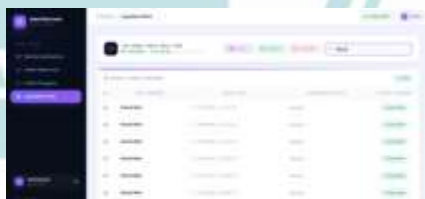
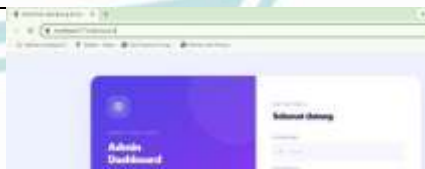


#### 4.4.3 Hasil Pengujian Fungsional Berbasis *Test Script*

Pengujian fungsional berbasis *test script* dilakukan terhadap 11 skenario pengujian yang mewakili fitur utama pada website monitoring dan manajemen log akses pintu. Berdasarkan hasil pengujian, seluruh skenario memperoleh status **Pass**, sehingga fitur utama pada sistem dinyatakan berjalan sesuai dengan hasil yang diharapkan.

Tabel 4. 7 Pengujian Fungsional Berbasis Test Script

| Test Case ID | Fitur yang Diuji     | Expected Results                                         | Actual Results                                                   | Share Screen |
|--------------|----------------------|----------------------------------------------------------|------------------------------------------------------------------|--------------|
| TS-01        | Login Administrator  | Sistem berhasil login dan menampilkan halaman dashboard. | Sistem berhasil login dan menampilkan dashboard Administrator.   |              |
| TS-02        | Login Gagal          | Sistem menolak login apabila kredensial tidak valid.     | Sistem menolak login dan menampilkan pesan kesalahan.            |              |
| TS-03        | Validasi CAPTCHA     | Sistem menolak login apabila CAPTCHA tidak valid.        | Sistem tidak mengizinkan login ketika CAPTCHA salah atau kosong. |              |
| TS-04        | Dashboard Monitoring | Sistem menampilkan informasi monitoring.                 | Halaman dashboard berhasil ditampilkan.                          |              |

|       |                       |                                                     |                                                               |                                                                                       |
|-------|-----------------------|-----------------------------------------------------|---------------------------------------------------------------|---------------------------------------------------------------------------------------|
| TS-05 | Daftar Wajah User     | Sistem menampilkan data wajah pengguna.             | Data wajah pengguna berhasil ditampilkan.                     |    |
| TS-06 | Registrasi Wajah User | Sistem memproses input registrasi wajah.            | Sistem berhasil memproses data registrasi wajah.              |    |
| TS-07 | Kelola Pengguna       | Sistem menampilkan data pengguna.                   | Data pengguna berhasil ditampilkan pada menu kelola pengguna. |   |
| TS-08 | Log Akses Pintu       | Sistem menampilkan riwayat akses pintu.             | Riwayat akses pintu berhasil ditampilkan.                     |  |
| TS-09 | Pencarian Log         | Sistem menampilkan log sesuai kata kunci pencarian. | Sistem berhasil menampilkan log sesuai pencarian.             |  |
| TS-10 | Proteksi Halaman      | Sistem menolak akses halaman tanpa login.           | Sistem mengarahkan pengguna ke halaman login.                 |  |

Hak Cipta :

© Hak Cipta milik Politeknik Negeri Jakarta

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :  
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.  
 b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta  
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|       |        |                                                            |                                                      |                                                                                     |
|-------|--------|------------------------------------------------------------|------------------------------------------------------|-------------------------------------------------------------------------------------|
| TS-11 | Logout | Sistem mengakhiri sesi Administrator dan kembali ke login. | Sistem berhasil logout dan kembali ke halaman login. |  |
|-------|--------|------------------------------------------------------------|------------------------------------------------------|-------------------------------------------------------------------------------------|

Tabel 4. 8 Rekapitulasi Hasil Pengujian Fungsional

| Keterangan              | Jumlah |
|-------------------------|--------|
| Total Test Case         | 11     |
| Pass                    | 11     |
| Fail                    | 0      |
| Not Tested              | 0      |
| Persentase Keberhasilan | 100%   |

Berdasarkan Tabel 4.8, hasil pengujian fungsional menunjukkan bahwa seluruh test case memperoleh status **Pass**. Persentase keberhasilan pengujian diperoleh dari perbandingan antara jumlah test case yang berhasil dengan jumlah seluruh test case yang diuji, yaitu 11 dari 11 test case atau sebesar **100%**. Hasil tersebut menunjukkan bahwa fitur utama pada website monitoring dan manajemen log akses pintu telah berjalan sesuai dengan skenario pengujian yang telah ditentukan.

#### 4.4.4 Hasil Pengujian Keamanan Berdasarkan OWASP Top Ten

Pengujian keamanan dilakukan untuk mengetahui kemampuan sistem dalam menghadapi risiko keamanan aplikasi web berdasarkan kategori OWASP Top Ten. Pengujian ini tidak hanya berfokus pada SQL Injection dan Brute Force, tetapi juga mencakup kontrol akses, konfigurasi keamanan, keamanan dependensi, perlindungan token, integritas data, pencatatan log keamanan, dan penanganan kesalahan.



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Tabel 4. 9 Hasil Pengujian Keamanan Berdasarkan OWASP Top Ten

| Code OWASP                                  | Komponen yang Diuji                                        | Skenario Pengujian                                                                                                                                                                                                                                                                                                                                       | Status         |
|---------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| A01 - <i>Broken Access Control</i>          | Endpoint dashboard dan log akses                           | Request menggunakan token valid menghasilkan respons <i>200 OK</i> , sedangkan request tanpa token dan token tidak valid ditolak dengan respons <i>401 Unauthorized</i> . Setelah proses logout, halaman dashboard tidak dapat diakses kembali.                                                                                                          | Memenuhi       |
| A02 - <i>Security Misconfiguration</i>      | Endpoint, metode HTTP, CORS, header respons, dan OWASP ZAP | Endpoint yang tidak tersedia menghasilkan respons <i>404 Not Found</i> , sedangkan penggunaan metode HTTP yang tidak sesuai menghasilkan <i>405 Method Not Allowed</i> tanpa menampilkan traceback. CORS menolak origin yang tidak terdaftar, tetapi header respons masih menampilkan versi Werkzeug/Python dan beberapa header keamanan belum tersedia. | Belum Memenuhi |
| A03 - <i>Software Supply Chain Failures</i> | Dependensi <i>frontend</i> dan <i>backend</i>              | Audit frontend menggunakan <i>npm audit</i> menunjukkan tidak ditemukan kerentanan,                                                                                                                                                                                                                                                                      | Memenuhi       |

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|                                        |                                                                    |                                                                                                                                                                                                                                                                                                              |                   |
|----------------------------------------|--------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
|                                        |                                                                    | sedangkan <i>pip-audit</i> tidak menemukan kerentanan yang diketahui pada dependensi backend yang diperiksa.                                                                                                                                                                                                 |                   |
| A04 -<br><i>Cryptographic Failures</i> | JWT, file <i>.env</i> ,<br><i>secret key</i> , dan komunikasi data | Payload JWT tidak memuat username, password, API Key, atau kredensial sensitif lainnya. File <i>.env</i> tidak dapat diakses melalui browser dan <i>secret key</i> tidak ditemukan pada source code frontend. Namun, penggunaan HTTPS/TLS belum dibuktikan dalam pengujian.                                  | Memenuhi Sebagian |
| A05 - <i>Injection</i>                 | Parameter input pada endpoint log akses pintu                      | SQLMap berhasil terhubung ke endpoint menggunakan JWT valid dan melakukan pengujian terhadap parameter yang digunakan pada proses pengambilan atau pencarian log. Hasil pemindaian menunjukkan bahwa parameter yang diuji tidak terindikasi dapat diinjeksi dan input tidak dieksekusi sebagai perintah SQL. | Memenuhi          |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|                                                  |                                                        |                                                                                                                                                                                                                                                                       |                   |
|--------------------------------------------------|--------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| A06 - <i>Insecure Design</i>                     | Alur akses API dan halaman <i>dashboard</i>            | Akses langsung ke API atau halaman dashboard tanpa melalui proses login tetap ditolak oleh <i>backend</i> dengan respons 401 <i>Unauthorized</i> , sehingga alur frontend tidak dapat digunakan untuk melewati <i>validasi autentikasi</i> .                          | Memenuhi          |
| A07 - <i>Authentication Failures</i>             | Login Administrator, CAPTCHA, dan <i>rate limiting</i> | Login menggunakan CAPTCHA atau password yang salah ditolak oleh sistem. Tiga percobaan login gagal memicu respons 429 <i>Too Many Requests</i> dan pemblokiran akses sementara.                                                                                       | Memenuhi          |
| A08 - <i>Software or Data Integrity Failures</i> | Integritas data log akses pintu                        | Request <i>DELETE</i> pada endpoint log ditolak dengan respons 405 <i>Method Not Allowed</i> . Hasil tersebut menunjukkan bahwa metode penghapusan log tidak tersedia, tetapi pengujian belum mencakup percobaan perubahan data melalui metode atau endpoint lainnya. | Memenuhi Terbatas |



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|                                                     |                                             |                                                                                                                                                                                                                                                                     |                   |
|-----------------------------------------------------|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| A09 - <i>Security Logging and Alerting Failures</i> | Log aktivitas dan peringatan keamanan       | Aktivitas login gagal, CAPTCHA atau password yang tidak sesuai, serta request yang tidak diizinkan tercatat pada log server. Namun, mekanisme pemberian alert otomatis kepada administrator belum dibuktikan.                                                       | Memenuhi Sebagian |
| A10 - <i>Mishandling of Exceptional Conditions</i>  | Penanganan kesalahan login dan akses sistem | Username atau password salah, CAPTCHA salah atau kedaluwarsa, serta kondisi akses yang diblokir ditangani menggunakan respons dan pesan kesalahan yang terkontrol tanpa menampilkan <i>500 Internal Server Error</i> , stack trace, atau informasi teknis internal. | Memenuhi          |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Berdasarkan Tabel 4.8, hasil pengujian keamanan berdasarkan OWASP Top Ten menunjukkan bahwa enam kategori memperoleh status Memenuhi, dua kategori memperoleh status Memenuhi Sebagian, satu kategori memperoleh status Memenuhi Terbatas, dan satu kategori memperoleh status Belum Memenuhi. Sistem telah mampu membatasi akses endpoint melalui token autentikasi, menolak input yang terindikasi sebagai SQL Injection, membatasi percobaan login tidak valid menggunakan CAPTCHA dan *rate limiting*, menjaga akses terhadap data log, mencatat aktivitas penting, serta menangani kondisi kesalahan tanpa menampilkan informasi teknis sensitif. Namun, sistem masih memerlukan penyempurnaan pada konfigurasi *security header*, penyembunyian informasi versi server, penerapan HTTPS/TLS, pengujian integritas data secara lebih menyeluruh, serta penerapan mekanisme peringatan keamanan otomatis.

#### 4.4.5 Hasil Pengujian Kualitas Website Menggunakan Lighthouse

Pengujian kualitas website dilakukan menggunakan Lighthouse pada dua halaman utama, yaitu halaman *Login Administrator* dan halaman *Dashboard Monitoring*. Pengujian ini bertujuan untuk memperoleh nilai kuantitatif terhadap kualitas halaman website berdasarkan empat aspek, yaitu *Performance*, *Accessibility*, *Best Practices*, dan *SEO*. Hasil pengujian ditampilkan pada Tabel 4.9.

Tabel 4. 10 Hasil Pengujian Kualitas Website Menggunakan Lighthouse

| No    | Halaman yang Diuji   | Performa | Accessibilitas | Best Practices | SEO | RataRata | Kategori          |
|-------|----------------------|----------|----------------|----------------|-----|----------|-------------------|
| LH-01 | Login Administrator  | 58       | 81             | 100            | 83  | 80,50    | Perlu Peningkatan |
| LH-02 | Dashboard Monitoring | 67       | 76             | 100            | 82  | 81,25    | Perlu Peningkatan |

Tabel 4. 11 Rekapitulasi Rata-Rata Skor Lighthouse

| No | Aspek Pengujian       | Halaman Login | Halaman Dashboard | Rata-Rata | Kategori          |
|----|-----------------------|---------------|-------------------|-----------|-------------------|
| 1  | <i>Performance</i>    | 58            | 67                | 62.50     | Perlu Peningkatan |
| 2  | <i>Accessibility</i>  | 81            | 76                | 78,50     | Perlu Peningkatan |
| 3  | <i>Best Practices</i> | 100           | 100               | 100,00    | Baik              |
| 4  | SEO                   | 83            | 82                | 82,50     | Perlu Peningkatan |

Tabel 4. 12 Kategori Skor Lighthouse

| Rentang Skor | Kategori          |
|--------------|-------------------|
| 90–100       | Baik              |
| 50–89        | Perlu Peningkatan |
| 0–49         | Buruk             |

Berdasarkan hasil pengujian menggunakan Lighthouse, halaman Login Administrator memperoleh rata-rata skor sebesar **80,50**, sedangkan halaman Dashboard Monitoring memperoleh rata-rata skor sebesar **81,25**. Hasil tersebut menunjukkan bahwa kualitas website secara umum berada pada kategori **Perlu Peningkatan**. Aspek *Best Practices* memperoleh nilai tertinggi, yaitu **100** pada kedua halaman, sehingga dapat dinyatakan bahwa website telah memenuhi praktik pengembangan web yang baik dari sisi audit Lighthouse.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pada aspek *Performance*, halaman Login memperoleh skor **58**, sedangkan halaman Dashboard memperoleh skor **67**. Nilai ini menunjukkan bahwa kecepatan dan efisiensi pemuatan halaman masih perlu ditingkatkan. Pada aspek *Accessibility*, halaman Login memperoleh skor **81** dan halaman Dashboard memperoleh skor **76**, sehingga masih terdapat beberapa bagian yang dapat diperbaiki agar halaman lebih mudah diakses. Pada aspek *SEO*, halaman Login memperoleh skor **83** dan halaman Dashboard memperoleh skor **82**, yang menunjukkan bahwa struktur dasar halaman sudah cukup baik, tetapi masih dapat ditingkatkan agar lebih sesuai dengan rekomendasi audit website.

Dengan demikian, hasil pengujian kualitas website menggunakan Lighthouse menunjukkan bahwa website monitoring dan manajemen log akses pintu telah memiliki nilai yang cukup baik pada aspek *Best Practices*, tetapi masih memerlukan peningkatan pada aspek *Performance*, *Accessibility*, dan *SEO*. Hasil ini menjadi dasar evaluasi kuantitatif terhadap kualitas halaman website sesuai dengan pendekatan penelitian yang digunakan.

#### 4.4.5 Analisis Hasil Pengujian

Berdasarkan hasil pengujian yang telah dilakukan, sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service* dianalisis berdasarkan tiga aspek utama, yaitu fungsionalitas website, keamanan sistem, dan kualitas website. Ketiga aspek tersebut digunakan untuk menilai kesesuaian fungsi sistem dengan kebutuhan yang telah ditetapkan, efektivitas mekanisme perlindungan yang diterapkan, serta kualitas halaman website ketika digunakan oleh Administrator.

Pada pengujian fungsional berbasis *test script*, seluruh fitur utama website diuji berdasarkan skenario yang telah ditentukan. Pengujian mencakup login Administrator, validasi CAPTCHA, dashboard monitoring, daftar wajah pengguna, registrasi wajah, kelola pengguna, log akses pintu, pencarian log, proteksi halaman tanpa login, dan logout. Berdasarkan hasil pengujian, sebanyak 11 *test case* memperoleh status *Pass*, dengan 0 *Fail* dan 0 *Not Tested*. Dengan demikian, tingkat keberhasilan pengujian fungsional mencapai 100%, sehingga fitur-fitur utama pada website dapat dinyatakan telah berjalan sesuai dengan hasil yang diharapkan.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pada pengujian keamanan berdasarkan OWASP Top Ten, sistem diuji terhadap sepuluh kategori risiko keamanan aplikasi web. Hasil pengujian menunjukkan bahwa sistem telah mampu menolak akses tanpa token atau menggunakan token yang tidak valid, membatasi akses langsung ke endpoint yang dilindungi, menangani endpoint yang tidak tersedia dan metode HTTP yang tidak sesuai tanpa menampilkan *traceback*, menjaga file konfigurasi agar tidak dapat diakses secara langsung melalui browser, membatasi percobaan login tidak valid melalui CAPTCHA dan *rate limiting*, serta menangani kondisi kesalahan menggunakan respons yang terkontrol. Audit terhadap dependensi juga telah dilakukan pada komponen frontend dan backend, sedangkan aktivitas login gagal dan request tertentu telah tercatat pada log sistem.

Meskipun demikian, belum seluruh kategori keamanan memperoleh status Memenuhi. Kategori A01, A03, A06, A07, dan A10 memperoleh status Memenuhi karena hasil pengujian telah sesuai dengan kriteria yang ditetapkan. Kategori A04 dan A09 memperoleh status Memenuhi Sebagian karena perlindungan file konfigurasi dan pencatatan aktivitas telah dibuktikan, tetapi penggunaan HTTPS/TLS dan mekanisme peringatan otomatis kepada Administrator belum ditunjukkan dalam pengujian. Kategori A08 memperoleh status Memenuhi Terbatas karena pengujian baru membuktikan bahwa metode penghapusan log tidak tersedia, tetapi belum mencakup seluruh kemungkinan perubahan data. Kategori A02 belum memenuhi seluruh kriteria karena informasi versi perangkat lunak masih dapat terlihat dan beberapa *security header* belum tersedia. Sementara itu, kategori A05 memerlukan pengujian ulang karena pengujian SQLMap sebelumnya belum berhasil terhubung ke endpoint dan belum dapat digunakan untuk menyimpulkan ketahanan sistem terhadap SQL Injection.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Berdasarkan hasil tersebut, mekanisme keamanan seperti JWT, API Key, CAPTCHA, *rate limiting*, validasi akses pada backend, audit dependensi, dan *error handling* telah memberikan perlindungan dasar terhadap akses tidak sah dan penyalahgunaan layanan. Namun, sistem masih memerlukan penyempurnaan pada konfigurasi *security header*, penyembunyian informasi versi server, penerapan HTTPS/TLS, pengujian integritas data yang lebih menyeluruh, penyediaan mekanisme *alert* keamanan otomatis, serta pengujian ulang SQL Injection menggunakan endpoint dan parameter yang valid. Oleh karena itu, hasil pengujian belum dapat digunakan untuk menyatakan bahwa seluruh kategori OWASP Top Ten memperoleh status *Pass*.

Pada pengujian kualitas website menggunakan Lighthouse, pengujian dilakukan terhadap dua halaman utama, yaitu halaman Login Administrator dan halaman Dashboard Monitoring. Halaman Login Administrator memperoleh skor *Performance* sebesar 58, *Accessibility* sebesar 81, *Best Practices* sebesar 100, dan SEO sebesar 83, dengan nilai rata-rata sebesar 80,50. Sementara itu, halaman Dashboard Monitoring memperoleh skor *Performance* sebesar 67, *Accessibility* sebesar 76, *Best Practices* sebesar 100, dan SEO sebesar 82, dengan nilai rata-rata sebesar 81,25. Berdasarkan kategori penilaian Lighthouse, kedua halaman berada pada kategori Perlu Peningkatan karena nilai rata-rata yang diperoleh berada pada rentang 50-89.

Hasil pengujian Lighthouse menunjukkan bahwa aspek *Best Practices* memperoleh nilai tertinggi, yaitu 100 pada kedua halaman. Hasil tersebut menunjukkan bahwa website telah menerapkan sejumlah praktik pengembangan web yang dinilai baik berdasarkan audit Lighthouse. Namun, aspek *Performance* masih menjadi bagian yang paling memerlukan perhatian karena hanya memperoleh skor 58 pada halaman Login Administrator dan 67 pada halaman Dashboard Monitoring. Nilai tersebut menunjukkan bahwa proses pemuatan halaman masih dapat dioptimalkan melalui pengurangan ukuran aset, optimasi gambar, pengelolaan file JavaScript dan CSS, pengurangan sumber daya yang menghambat proses pemuatan, serta peningkatan efisiensi pemanggilan data pada halaman.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Aspek *Accessibility* dan SEO juga masih dapat ditingkatkan. Perbaikan pada aspek *Accessibility* dapat dilakukan melalui penambahan label pada elemen formulir, peningkatan kontras warna, penyusunan struktur heading yang lebih teratur, dan pemberian keterangan pada elemen visual. Sementara itu, peningkatan aspek SEO dapat dilakukan melalui penyempurnaan judul halaman, deskripsi halaman, struktur elemen HTML, serta atribut pendukung lainnya. Meskipun website monitoring tidak ditujukan sebagai website publik, penyempurnaan tersebut tetap diperlukan untuk meningkatkan kerapian struktur dan kualitas teknis halaman.

Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem telah memenuhi kebutuhan utama dari sisi fungsionalitas karena seluruh *test case* memperoleh status *Pass*. Dari sisi keamanan, sebagian besar mekanisme perlindungan dasar telah berjalan, tetapi masih terdapat beberapa kategori yang perlu disempurnakan dan diuji ulang sebelum seluruh aspek keamanan dapat dinyatakan terpenuhi. Sementara itu, hasil pengujian Lighthouse menunjukkan bahwa kedua halaman memiliki kualitas yang cukup baik, terutama pada aspek *Best Practices*, tetapi masih memerlukan peningkatan pada aspek *Performance*, *Accessibility*, dan SEO. Dengan demikian, sistem monitoring *real-time* dan manajemen log akses pintu berbasis *web service* telah dapat menjalankan fungsi utama sesuai dengan tujuan penelitian, tetapi masih memiliki ruang pengembangan pada konfigurasi keamanan dan kualitas teknis halaman website.