

Rancang Bangun Sistem Monitoring Real-Time dan Manajemen Log Akses Pintu Berbasis Web Service pada Smart Door Lock IoT

Kurniawan Sandi

Program Studi Teknik Multimedia dan Jaringan, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta
Depok, Jawa Barat

Kurniawan.sandi.tik22@mhs.wj.ac.id | +62 895-1857-3420

Abstrak - Perkembangan smart door lock berbasis Internet of Things menuntut ketersediaan layanan pemantauan yang dapat menampilkan aktivitas akses secara real-time, menyimpan log secara terpusat, serta memiliki mekanisme keamanan dan kualitas halaman yang terukur. Penelitian ini merancang dan membangun sistem monitoring real-time dan manajemen log akses pintu berbasis web service dengan pendekatan decoupled architecture, menggunakan Flask sebagai backend, React JS sebagai frontend, Supabase PostgreSQL sebagai basis data, dan Virtual Private Server sebagai lingkungan deployment. Sistem dilengkapi mekanisme keamanan berupa API Key, JSON Web Token, CAPTCHA, rate limiting, parameterized query, dan Row Level Security. Pengujian dilakukan melalui tiga pendekatan, yaitu pengujian fungsional berbasis test script, pengujian keamanan berdasarkan sepuluh kategori OWASP Top Ten menggunakan Postman, OWASP ZAP, npm audit, pip-audit, dan SQLMap, serta pengujian kualitas halaman menggunakan Lighthouse. Hasil pengujian menunjukkan seluruh 11 skenario fungsional memperoleh status Pass dengan tingkat keberhasilan 100%. Pada pengujian keamanan, enam kategori memperoleh status Memenuhi, dua Memenuhi Sebagian, satu Memenuhi Terbatas, dan satu Belum Memenuhi. Pengujian Lighthouse menghasilkan rata-rata skor 80,50 pada halaman login dan 81,25 pada dashboard, keduanya berkategori Perlu Peningkatan dengan skor Best Practices tertinggi sebesar 100. Sistem telah menjalankan fungsi utama dan memiliki perlindungan dasar, tetapi masih memerlukan penguatan pada konfigurasi security header, HTTPS/TLS, mekanisme alert otomatis, serta optimasi performance, accessibility, dan SEO.

Kata Kunci - web service, monitoring real-time, manajemen log akses, OWASP Top Ten, Lighthouse, smart door lock.

I. PENDAHULUAN

Perkembangan teknologi Internet of Things telah mendorong penerapan sistem keamanan fisik yang lebih modern melalui smart door lock, yang tidak hanya berfungsi sebagai pengganti kunci konvensional tetapi juga dapat diintegrasikan dengan autentikasi, pemantauan, dan pencatatan aktivitas akses secara digital [1], [2]. Pada penerapannya, sistem tersebut membutuhkan layanan berbasis web agar administrator dapat memantau status akses pintu, melihat riwayat aktivitas, serta mengelola data pengguna secara terpusat dan real-time.

Implementasi smart door lock pada umumnya lebih banyak berfokus pada sisi perangkat keras dan metode autentikasi, seperti sensor biometrik atau pengenalan wajah, sementara aspek pengelolaan data akses, ketersediaan log, keamanan layanan backend, serta performa sistem belum menjadi perhatian utama. Penyimpanan data akses secara lokal pada perangkat juga berisiko hilang apabila perangkat rusak, memori penuh, atau terjadi gangguan penyimpanan, sehingga diperlukan sistem monitoring dan manajemen log akses yang menyimpan data secara terpusat melalui web service.

Integrasi smart door lock dengan web service mempermudah pengelolaan data, tetapi juga memunculkan risiko keamanan aplikasi web, seperti akses endpoint tanpa otorisasi, penyalahgunaan token, kesalahan konfigurasi, manipulasi input, kelemahan autentikasi, serta gangguan integritas data log. OWASP Top Ten 2025 menempatkan Broken Access Control, Security Misconfiguration, Injection, Authentication Failures, dan Security Logging and Alerting Failures sebagai risiko kritis pada aplikasi web yang perlu diperhatikan dalam pengembangan dan pengujian sistem [3].

Berdasarkan kondisi tersebut, terdapat kesenjangan penelitian pada pengembangan smart door lock, yaitu belum optimalnya pembahasan mengenai sistem monitoring dan manajemen log akses pintu yang berjalan secara real-time sekaligus diuji dari sisi fungsionalitas, keamanan, dan performa secara terukur. Penelitian sebelumnya cenderung berfokus pada mekanisme autentikasi atau perangkat utama [1], [2], sedangkan pengujian layanan web umumnya masih dibatasi pada aspek tertentu, seperti pengujian celah basis data menggunakan SQLMap [4] atau mekanisme logging, deteksi anomali, dan pemblokiran otomatis pada layanan web [5], tanpa mencakup pengujian fungsional, keamanan, dan kualitas halaman secara bersamaan.

Untuk menjawab permasalahan tersebut, penelitian ini merancang dan mengimplementasikan sistem monitoring real-time dan manajemen log akses pintu berbasis web service dengan konsep decoupled architecture, menggunakan Flask sebagai backend, React JS sebagai frontend, Supabase PostgreSQL sebagai basis data, dan Virtual Private Server sebagai lingkungan deployment, yang dilengkapi mekanisme keamanan API Key, JSON Web Token, CAPTCHA, rate limiting, parameterized query, dan Row Level Security. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem tersebut, serta menguji dan menganalisisnya dari aspek fungsionalitas menggunakan test script, keamanan berdasarkan OWASP Top Ten, dan kualitas halaman berdasarkan audit Lighthouse.

II. METODOLOGI PENELITIAN

A. Rancangan dan Objek 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. Objek penelitian adalah produk rekayasa perangkat lunak berupa sistem monitoring real-time dan manajemen log akses pintu berbasis *web service*, mencakup *backend* Flask yang berjalan pada Virtual Private Server, basis data PostgreSQL yang dikelola melalui Supabase, serta *dashboard* administrator berbasis React JS.

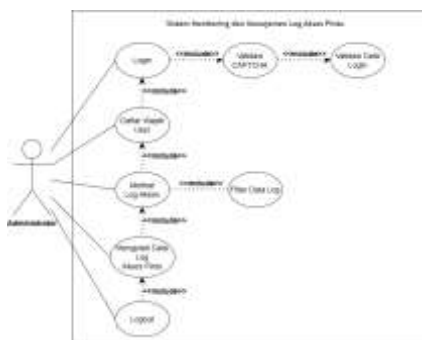
B. Perancangan Sistem

Sistem dirancang menggunakan konsep *decoupled architecture*, yaitu pemisahan antara *frontend*, *backend*, basis data, dan lingkungan *deployment*. Administrator mengakses sistem melalui *web browser* yang berkomunikasi dengan *frontend* React JS; *frontend* selanjutnya berkomunikasi dengan *backend* Flask melalui REST API. *Backend* memproses autentikasi dan validasi akses, serta menghubungkan sistem dengan basis data Supabase PostgreSQL, sedangkan data log akses pintu dari sistem utama diterima melalui *endpoint* API yang dilengkapi validasi API Key sebelum disimpan dan ditampilkan pada *dashboard monitoring*. Gambar 1 menunjukkan arsitektur sistem yang dirancang.



Gambar 1. Arsitektur Sistem

Interaksi Administrator dengan fitur utama sistem digambarkan melalui *use case diagram* pada Gambar 2, yang mencakup login, *dashboard monitoring*, daftar dan registrasi wajah pengguna, kelola pengguna, log akses pintu, pencarian log, serta logout.



Gambar 2. Use Case Diagram Sistem

C. Teknik Pengumpulan dan Pengujian Data

Data dikumpulkan melalui studi literatur, perancangan, dan implementasi sistem. Pengujian dilakukan melalui tiga pendekatan. Pertama, pengujian fungsional menggunakan *test script* terhadap 11 skenario yang mewakili fitur utama sistem, dengan status *Pass* atau *Fail* berdasarkan kesesuaian hasil aktual terhadap hasil yang diharapkan [6]. Kedua, pengujian keamanan berdasarkan sepuluh kategori OWASP Top Ten [3], meliputi kontrol akses, konfigurasi keamanan, keamanan dependensi, kriptografi, injeksi, desain sistem,

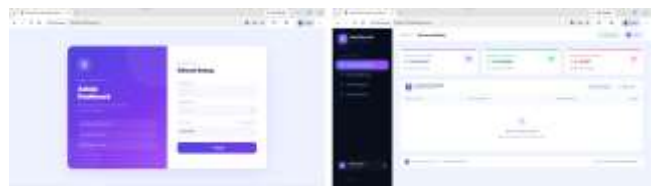
autentikasi, integritas data, pencatatan log, dan penanganan kesalahan, menggunakan Postman, OWASP ZAP, npm audit, pip-audit, SQLMap [4], dan Hydra [7]. Ketiga, pengujian kualitas halaman menggunakan Lighthouse pada aspek *performance*, *accessibility*, *best practices*, dan SEO, dengan kategori skor Baik (90–100), Perlu Peningkatan (50–89), dan Buruk (0–49). Persentase keberhasilan pengujian fungsional dihitung menggunakan rumus berikut:

$$\text{Persentase Keberhasilan (\%)} = (\text{Jumlah Skenario Berhasil} / \text{Total Skenario}) \times 100\% \quad (1)$$

III. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

Backend dikembangkan menggunakan Flask dengan Python di dalam *virtual environment* terpisah untuk mengelola dependensi, sedangkan *frontend* dibangun menggunakan React JS yang berkomunikasi dengan *backend* melalui REST API. Basis data dikelola melalui Supabase PostgreSQL dengan tiga tabel utama, yaitu *access_logs* untuk menyimpan riwayat akses pintu, *profiles* untuk data profil pengguna, dan *registered_faces* untuk data wajah terdaftar. *Backend* dan *frontend* di-deploy pada Virtual Private Server berbasis Ubuntu Server 22.04 LTS. Gambar 3 menunjukkan implementasi halaman login Administrator dan *dashboard monitoring real-time* yang menampilkan status sistem serta ringkasan data akses.



Gambar 3. Tampilan Halaman Login dan Dashboard Monitoring

B. Implementasi Mekanisme Keamanan

Sistem menerapkan beberapa lapisan keamanan, yaitu API Key untuk validasi akses dari sistem utama, JSON Web Token untuk pengelolaan sesi Administrator [8], CAPTCHA pada halaman login untuk membedakan akses manusia dari skrip otomatis [9], *rate limiting* untuk membatasi percobaan login berulang, *parameterized query* untuk mengurangi risiko injeksi, serta Row Level Security pada Supabase untuk membatasi akses data pada tingkat baris. Kombinasi mekanisme tersebut memberikan lapisan perlindungan pada sisi aplikasi, *endpoint* API, sesi pengguna, dan basis data sebelum sistem diuji lebih lanjut menggunakan pendekatan OWASP Top Ten.

C. Hasil Pengujian Fungsional Berbasis Test Script

Pengujian fungsional dilakukan terhadap 11 skenario yang mewakili fitur utama sistem, meliputi login Administrator, login gagal, 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, seluruh 11 skenario memperoleh status *Pass*, 0 *Fail*, dan 0 *Not Tested*, sehingga tingkat keberhasilan pengujian

fungsional mencapai 100%, sebagaimana dirangkum pada Tabel I.

Tabel I. Rekapitulasi Hasil Pengujian Fungsional

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

D. Hasil Pengujian Keamanan Berdasarkan OWASP Top Ten

Pengujian keamanan dilakukan terhadap sepuluh kategori OWASP Top Ten, tidak hanya berfokus pada SQL Injection dan *brute force*, tetapi juga mencakup kontrol akses, konfigurasi keamanan, keamanan dependensi, kriptografi, integritas data, pencatatan log, dan penanganan kesalahan. Ringkasan status pengujian ditunjukkan pada Tabel II.

Tabel II. Ringkasan Status Pengujian Keamanan OWASP Top Ten

Status	Jml.	Kode Kategori
Memenuhi	6	A01, A03, A05, A06, A07, A10
Memenuhi Sebagian	2	A04, A09
Memenuhi Terbatas	1	A08
Belum Memenuhi	1	A02

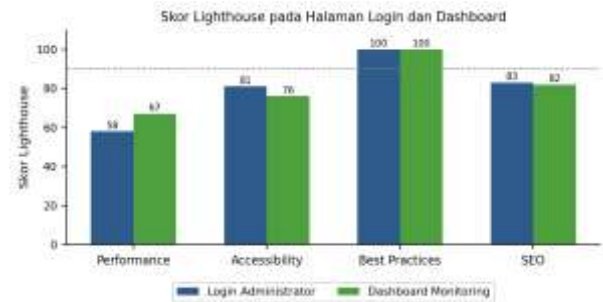
Hasil pengujian menunjukkan bahwa sistem mampu menolak akses tanpa token atau dengan token tidak valid pada kategori *Broken Access Control* (A01), menahan permintaan yang terindikasi SQL Injection pada kategori *Injection* (A05) menggunakan SQLMap [4], membatasi percobaan login berulang melalui CAPTCHA dan *rate limiting* pada kategori *Authentication Failures* (A07), serta menangani kesalahan tanpa menampilkan informasi teknis sensitif pada kategori *Mishandling of Exceptional Conditions* (A10). Audit dependensi menggunakan npm audit dan pip-audit pada kategori *Software Supply Chain Failures* (A03) juga tidak menemukan kerentanan yang diketahui, sedangkan pengujian pada kategori *Insecure Design* (A06) menunjukkan bahwa alur akses ke API maupun halaman *dashboard* tanpa melalui proses login tetap ditolak oleh *backend*.

Meskipun demikian, kategori *Cryptographic Failures* (A04) dan *Security Logging and Alerting Failures* (A09) baru memperoleh status Memenuhi Sebagian karena penggunaan HTTPS/TLS dan mekanisme *alert* otomatis belum dibuktikan dalam pengujian. Kategori *Software or Data Integrity Failures* (A08) memperoleh status Memenuhi Terbatas karena pengujian baru mencakup penolakan metode penghapusan data pada *endpoint* log, dan belum mencakup seluruh kemungkinan perubahan data. Kategori *Security Misconfiguration* (A02) belum memenuhi seluruh kriteria karena informasi versi perangkat lunak masih terlihat pada header respons dan sejumlah *security header* belum tersedia. Dengan demikian,

mekanisme keamanan yang diterapkan telah memberikan perlindungan dasar, tetapi belum seluruh kategori OWASP Top Ten dapat dinyatakan terpenuhi secara penuh.

E. Hasil Pengujian Kualitas Website Menggunakan Lighthouse

Pengujian kualitas halaman dilakukan pada dua halaman utama, yaitu Login Administrator dan Dashboard Monitoring, menggunakan Lighthouse pada aspek *performance*, *accessibility*, *best practices*, dan SEO. Tabel III dan Gambar 4 menunjukkan hasil pengujian pada kedua halaman.



Gambar 4. Grafik Skor Lighthouse pada Halaman Login dan Dashboard

Tabel III. Rekapitulasi Rata-Rata Skor Lighthouse

Aspek	Login	Dashboard	Rata-rata
Performance	58	67	62,50
Accessibility	81	76	78,50
Best Practices	100	100	100,00
SEO	83	82	82,50

Halaman Login memperoleh rata-rata skor 80,50, sedangkan halaman Dashboard memperoleh rata-rata skor 81,25, sehingga keduanya berada pada kategori Perlu Peningkatan (skor 50–89). Aspek *Best Practices* memperoleh skor tertinggi, yaitu 100 pada kedua halaman, sedangkan aspek *Performance* menjadi bagian yang paling memerlukan perhatian dengan skor 58 pada halaman Login dan 67 pada halaman Dashboard. Aspek *Accessibility* memperoleh skor 81 pada halaman Login dan 76 pada halaman Dashboard, sedangkan aspek SEO memperoleh skor 83 pada halaman Login dan 82 pada halaman Dashboard.

F. Evaluasi Sistem

Secara keseluruhan, sistem telah memenuhi kebutuhan utama dari sisi fungsionalitas karena seluruh *test case* memperoleh status *Pass*. Dari sisi keamanan, mekanisme perlindungan dasar seperti JWT, API Key, CAPTCHA, *rate limiting*, validasi akses pada *backend*, audit dependensi, dan *error handling* telah berjalan, tetapi konfigurasi *security header*, penyembunyian informasi versi server, penerapan HTTPS/TLS, mekanisme *alert* otomatis, dan pengujian integritas data masih memerlukan penyempurnaan sebelum seluruh kategori OWASP Top Ten dapat dinyatakan terpenuhi. Dari sisi kualitas halaman, hasil Lighthouse menunjukkan kualitas yang cukup baik pada aspek *Best Practices*, tetapi *performance*, *accessibility*, dan SEO masih memerlukan optimasi, seperti pengurangan ukuran aset,

optimasi gambar dan berkas JavaScript/CSS, penambahan label pada elemen formulir, peningkatan kontras warna, serta penyempurnaan struktur judul dan deskripsi halaman.

IV. KESIMPULAN DAN SARAN

Sistem monitoring real-time dan manajemen log akses pintu berbasis *web service* berhasil dirancang dan dibangun menggunakan Flask sebagai *backend*, React JS sebagai *frontend*, Supabase PostgreSQL sebagai basis data, dan Virtual Private Server sebagai lingkungan *deployment* dengan konsep *decoupled architecture*. Sistem mampu menjalankan fungsi login Administrator, validasi CAPTCHA, *dashboard monitoring*, daftar dan registrasi wajah pengguna, kelola pengguna, log akses pintu, pencarian log, proteksi halaman, dan logout.

Hasil pengujian menunjukkan bahwa seluruh 11 skenario pengujian fungsional memperoleh status *Pass* dengan tingkat keberhasilan 100%. Pengujian keamanan berdasarkan OWASP Top Ten menghasilkan enam kategori berstatus Memenuhi, dua kategori Memenuhi Sebagian, satu kategori Memenuhi Terbatas, dan satu kategori Belum Memenuhi, yang menunjukkan bahwa sistem telah memiliki perlindungan dasar namun masih memerlukan penyempurnaan pada beberapa aspek keamanan. Pengujian kualitas halaman menggunakan Lighthouse menghasilkan rata-rata skor 80,50 pada halaman Login dan 81,25 pada halaman Dashboard, keduanya berkategori Perlu Peningkatan, dengan skor *Best Practices* tertinggi sebesar 100.

Pengembangan selanjutnya disarankan menerapkan domain resmi dan protokol HTTPS agar komunikasi antara *frontend*, *backend*, dan basis data lebih aman, mengingat sistem memproses data autentikasi, token sesi, dan log akses pintu yang perlu dilindungi selama transmisi data. Selain itu, disarankan menambahkan fitur notifikasi keamanan otomatis melalui email atau layanan pesan lain apabila terjadi aktivitas mencurigakan, seperti percobaan login berulang atau permintaan tanpa token, agar Administrator dapat mengetahui potensi ancaman secara lebih cepat.

REFERENSI

- [1] Andreas, Aldawira, C.R., Putra, H.W., Hanafiah, N., Surjarwo, S. and Wibisurya, A. Door Security System For Home Monitoring Based On ESP32. *Procedia Computer Science*. 2019; 157: 673-682.
- [2] Maulana, I., Azriadi, E. and Musridho, R.J. Rancang Bangun Sistem Smart Door Lock Menggunakan Mikrokontroler ESP32 Berbasis Internet of Things (IoT) dan Smartphone Android. *Jurnal Teknik Industri Terintegrasi (JUTIN)*. 2023; 6(1): 195-208.
- [3] OWASP Foundation. OWASP Top 10:2025. 2025.
- [4] Praptomo, S., Suryanto and Kurniawan, J. Database Vulnerability Testing Memanfaatkan SQLMap Pada Ubuntu 22.04 (Studi Kasus: Antaratech.net). *Jurnal Informatika Medis (J-INFORMED)*. 2023; 1(2): 68-72.
- [5] Situmorang, R., Kiswanto, D., Khoiriah, N. and Harahap, F.A. Implementasi Sistem Keamanan Jaringan Berbasis Web: Logging, Deteksi Anomali dan Pemblokiran Otomatis. *Jurnal Informatika dan Teknik Elektro Terapan*. 2026; 14(1): 1230-1236.
- [6] Putri, S.J., Putri, D.G.P. and Putra, W.H.N. Analisis Komparasi Pada Teknik Black Box Testing (Studi Kasus: Website Lars). *Journal of Internet and Software Engineering (JISE)*. 2024; 5(1): 23-28.
- [7] Putra, D.M.J., Anantra, I.N.N.Y., Kusuma, P.A., Pratama, P.D.J., Saskara, G.A.J. and Listartha, I.M.E. Analisis Perbandingan Serangan Hydra, Medusa dan Ncrack pada Password Attack. *JINTEKS (Jurnal Informatika Teknologi dan Sains)*. 2022; 4(4): 461-466.
- [8] Jones, M., Bradley, J. and Sakimura, N. JSON Web Token (JWT), RFC 7519. Internet Engineering Task Force. 2015.
- [9] Gulo, S.A., Kiswanto, D., Arifin, M.H. and Aulia, W. Implementasi Sistem Login Web Berbasis ZTA dengan Integrasi OTP Brevo dan CAPTCHA. *Jurnal Informatika Progres*. 2026; 18(1): 23-29.