# RANCANG BANGUN MODEL KONTROL AKSES DINAMIS BERBASIS KONTEKS PADA ARSITEKTUR ZERO TRUST

**Ruth Amelia Vega S Meliala*[1)], Dedy Kiswanto[2)], Salsa Nabila Harahap[3)], Revidamurty Dly[4)]**

1. Ilmu Komputer, Universitas Negeri Medan
email: ruthameliia.4233250035@gmail.com
2. Ilmu Komputer, Universitas Negeri Medan
email: dedykiswanto@unimed.ac.id
3. Ilmu Komputer, Universitas Negeri Medan
email: salsanhrp.4231250027@mhs.unimed.ac.id
4. Ilmu Komputer, Universitas Negeri Medan
email: revidamurti.4231250007@mhs.unimed.ac.id

***Abstract***

*The rapid development of digital systems and interconnected environments has created new challenges in securing data. Traditional perimeter-based security models are no longer adequate to protect sensitive information from internal and external threats. This study proposes the design and implementation of a Context-Based Dynamic Access Control Model within the Zero Trust Architecture (ZTA) framework. The proposed system integrates contextual authentication, adaptive risk evaluation, and a dynamic policy engine to implement more granular access control in multi-user web applications. The prototype was developed using Node.js, Express.js, and MySQL, featuring multi-factor authentication, contextual verification via OTP, session management, and security notifications.The test results indicate that the system is capable of detecting changes in access context, enforcing re-authentication, and recording all user activities for auditing and anomaly detection purposes. The integration of contextual authentication, adaptive access control, and Zero Trust principles has been proven to enhance data protection and user accountability without reducing system usability..*

***Kata Kunci**: Zero Trust Architecture, Contextual Authentication, Policy Engine, Dynamic Access Control, Information Security*

## A. PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat telah membawa kemudahan dalam pengelolaan data dan layanan digital di berbagai sektor. Namun, di sisi lain, peningkatan konektivitas sistem juga menghadirkan tantangan baru dalam hal keamanan informasi. Ancaman terhadap sistem tidak hanya datang dari luar, tetapi juga dari dalam organisasi itu sendiri. *Insider threat* menjadi salah satu masalah yang sulit diatasi karena pelaku memiliki kredensial resmi dan hak akses sah yang bisa dimanfaatkan untuk tujuan yang tidak semestinya (Laksono & Sari, 2025).

Model keamanan tradisional yang berfokus pada batas jaringan (perimeter *security*) dinilai sudah tidak cukup untuk menghadapi tantangan tersebut. Oleh

karena itu, muncul paradigma baru yang dikenal dengan *Zero Trust Architecture* (ZTA). Prinsip utama dari ZTA adalah "*never trust, always verify*", di mana setiap permintaan akses, baik dari pengguna internal maupun eksternal, harus melewati proses autentikasi dan otorisasi berlapis sebelum diizinkan mengakses sumber daya sistem (Agustina & Achmad, 2019). Pendekatan ini terbukti mampu mengurangi risiko pelanggaran data serta meningkatkan visibilitas terhadap aktivitas pengguna.

Dalam lingkungan pemerintahan, penerapan kontrol akses berbasis *Zero Trust* telah membantu menjaga keamanan layanan informasi publik. Melalui pengelolaan autentikasi, otorisasi, dan audit yang ketat, sistem informasi dapat terlindungi dari akses tidak sah serta menjaga integritas data sensitif yang dikelola (Hose et al., 2025). Penerapan prinsip yang sama juga diterapkan di sektor pendidikan, di mana Zero Trust digunakan untuk melindungi data akademik dan keuangan mahasiswa dari potensi kebocoran akibat kredensial yang disalahgunakan (Zuhriyanto & Sri Rahayu Astari, 2025).

Seiring meningkatnya kebutuhan akan keamanan data digital di lingkungan pendidikan, penggunaan sistem server internal menjadi solusi penting dalam menjaga integritas dan kerahasiaan informasi (Lubis et al., 2022). Menjelaskan bahwa penerapan *mail server* dengan domain institusi, seperti melalui platform Roundcube berbasis Ubuntu, dapat meningkatkan keamanan data pengguna karena hanya akun dengan domain resmi institusi yang dapat mengakses sistem tersebut. Pendekatan ini tidak hanya memperkuat identitas digital organisasi, tetapi juga meminimalkan risiko kebocoran data akibat penggunaan layanan pihak ketiga yang kurang terkontrol. Prinsip pembatasan akses berbasis domain internal tersebut sejalan dengan konsep *Zero Trust Architecture*, di mana setiap entitas jaringan harus diverifikasi sebelum diberikan akses terhadap sumber daya sistem.

Seiring meningkatnya kebutuhan akan keamanan yang adaptif, berbagai penelitian mulai menggabungkan ZTA dengan teknologi lain seperti Blockchain untuk memperkuat sistem autentikasi dan privasi data. Dengan konsep *Decentralized Identifiers* (DID) dan *Self-Sovereign Identity* (SSI), pengguna memiliki kendali penuh terhadap identitas digital mereka tanpa bergantung pada otoritas tunggal. Pendekatan ini meningkatkan keaslian data dan memperkuat privasi di lingkungan digital yang semakin terbuka (Mukhlisin & Agung Firmansyah, 2025). Selain itu, integrasi Blockchain dan ZTA juga terbukti mampu memperkuat ketahanan sistem *cloud* dengan menyediakan jejak transaksi yang transparan dan sulit dimanipulasi, sehingga mempermudah proses audit keamanan (Pribadi Fitrian et al., 2024).

Upaya peningkatan keamanan tidak hanya berfokus pada arsitektur sistem, tetapi juga pada mekanisme autentikasi pengguna. Penggunaan *Multi-Factor Authentication* (MFA) menjadi langkah penting dalam memperkuat validasi identitas, karena sistem dapat memverifikasi pengguna menggunakan lebih dari satu faktor keamanan (Junga & Sulistyo, 2025). Implementasi MFA terbukti mampu mengurangi risiko pembajakan akun dan serangan *credential theft* yang kerap menargetkan sistem daring (Agustina & Achmad, 2019).

Selain MFA, pendekatan keamanan adaptif juga diterapkan di lapisan jaringan, misalnya melalui *port knocking* dinamis berbasis waktu pada perangkat *router* Mikrotik. Dengan menerapkan mekanisme perubahan *port* secara

berkala, sistem dapat mempersulit upaya penyusupan atau serangan *brute force*. Pendekatan semacam ini menunjukkan bahwa keamanan berbasis konteks dan waktu dapat menjadi lapisan pertahanan tambahan yang efektif (Haeruddin et al., 2025).

Sementara itu, dalam konteks lingkungan *multi-tenant* di *cloud computing*, kontrol akses berbasis kepercayaan (*trust-based access control*) menjadi perhatian utama. Sistem semacam ini menilai setiap aktivitas pengguna berdasarkan perilaku dan konteks untuk menentukan tingkat kepercayaannya secara dinamis. Pendekatan adaptif ini terbukti mampu meningkatkan deteksi anomali serta meminimalkan risiko pelanggaran yang berasal dari dalam sistem (Laksono & Sari, 2025).

Berdasarkan berbagai penelitian tersebut, dapat disimpulkan bahwa tren keamanan digital saat ini bergerak menuju integrasi antara kontrol akses dinamis, autentikasi adaptif, dan *Zero Trust Architecture*. Kombinasi ketiganya menghasilkan sistem yang lebih responsif, kontekstual, dan mampu beradaptasi terhadap perubahan perilaku pengguna maupun kondisi jaringan. Oleh karena itu, penelitian ini mengusulkan "Perancangan Model Kontrol Akses Dinamis Berbasis Konteks dalam Kerangka Keamanan Zero Trust pada Sistem Reservasi Aset Internal" sebagai langkah untuk menghadirkan model keamanan yang tidak hanya tangguh dan fleksibel, tetapi juga relevan dengan kebutuhan sistem organisasi modern yang terus berkembang (Mugianto & Budiarto, 2024).

## B. METODE PENELITIAN


**Gambar 1.** Diagram Alir Penelitian

### Studi Literatur dan Analisis Kebutuhan

*Arsitektur Zero Trust* (ZTA) menjadi pendekatan keamanan modern yang meniadakan kepercayaan implisit antar entitas jaringan. Model ini menekankan verifikasi identitas dan otorisasi dinamis terhadap setiap akses sistem, berbeda dengan model perimeter yang menganggap jaringan internal aman (Mugianto & Budiarto, 2024).

Penelitian oleh Yulianto dkk. menunjukkan bahwa penerapan *Zero Trust Model* pada infrastruktur *private server* dapat memperkuat keamanan data dan mencegah serangan dari luar dengan mengintegrasikan *firewall*, *Cloudflare*, serta autentikasi pengguna berlapis (Pananto & Damayanti, 2023)

Pendekatan lain dikemukakan oleh Haeruddin dkk., yang mengimplementasikan *Zero Trust Network* menggunakan *Ferrumgate* dan metode NDLC. Hasilnya menunjukkan peningkatan keamanan melalui autentikasi multifaktor, segmentasi jaringan, serta pengawasan aktivitas

pengguna secara real-time (Haeruddin et al., 2024).

Dalam konteks autentikasi kontekstual, Mugianto dan Budiarto meneliti penerapan *Zero Trust Network* pada *smart home* untuk menghadapi ancaman data *sniffing*. Sistem ini memanfaatkan validasi perangkat serta autentikasi berbasis identitas untuk mencegah akses tidak sah pada jaringan IoT (Mugianto & Budiarto, 2024).

Sementara itu, penelitian oleh Efendi dkk., menyoroti kelemahan model keamanan berbasis perimeter melalui pengujian *vulnerability assessment* pada aplikasi web. Ditemukan bahwa konfigurasi SSL, autentikasi yang lemah, dan celah injeksi masih menjadi risiko utama, sehingga dibutuhkan pendekatan *Zero Trust* dan kontrol kebijakan yang lebih granular di tingkat aplikasi (Efendi et al., 2024).

Dari kelima penelitian tersebut dapat disimpulkan bahwa konsep *Zero Trust*, autentikasi kontekstual, dan penggunaan *policy engine* saling melengkapi untuk menciptakan sistem keamanan adaptif. Pendekatan ini menutup celah pada model perimeter tradisional dan relevan diterapkan pada aplikasi web multi-pengguna yang mengelola data sensitif (Pananto & Damayanti, 2023).

**Perancangan Sistem dan Arsitektur**

Arsitektur sistem ini dibangun di atas kombinasi teknologi kontemporer seperti Node.js, Express.js, dan database MySQL pada tahap perancangan. Sequelize berfungsi sebagai Maper Objek Relatif (ORM) untuk menghubungkan antara logika aplikasi dan persistensi data. Skema database dirancang untuk mendukung operasi aplikasi dan keamanan. Ini mencakup entitas utama untuk aplikasi studi kasus, seperti Ruangan (aset ruangan) dan Reservasi (jadwal pemesanan), serta entitas penting untuk kerangka keamanan, seperti Pengguna sebagai pusat identitas pengguna, *LoginHistories* untuk mencatat konteks akses, dan *AuditLogs* sebagai rekaman audit untuk setiap tindakan administratif. Fokus penelitian ini terletak pada desain Model Kontrol Akses Dinamis, yang merupakan integrasi dari berbagai lapisan verifikasi yang bekerja sama, seperti verifikasi identitas (kredensial), evaluasi konteks (autentikasi adaptif), dan penegakan kebijakan (*policy engine* dan RBAC).

**Implementasi dan Pengembangan Prototipe**

Seluruh rancangan sistem diwujudkan menjadi sebuah prototipe fungsional pada tahap implementasi. Proses ini dimulai dengan pembuatan komponen keamanan inti yang menjadi dasar aplikasi. Komponen ini termasuk penggunaan Autentikasi Kontekstual (Adaptif) untuk mengevaluasi risiko login, Pengendalian Akses Berbasis Rol (RBAC) untuk pemisahan hak akses, *Policy Engine* dinamis yang memungkinkan pencabutan akses secara *real-time* (fitur suspend), dan mekanisme pencabutan sesi lintas perangkat melalui pengembangan token. Antarmuka pengguna (UI) dibangun di atas fondasi keamanan ini yang kokoh. Antarmuka pengguna ini terdiri dari tiga bagian utama: Dasbor Admin yang digunakan untuk mengelola dan memantau, Dasbor Pengguna, yang berfungsi sebagai aplikasi studi kasus (Sistem Reservasi Ruangan), dan Pusat Keamanan Pengguna, yang digunakan untuk mengatur sesi pengguna akhir.

**Pengujian Validasi dan Skenario**

Pada tahap pengujian dan validasi, serangkaian pengujian berbasis skenario dilakukan untuk memverifikasi setiap elemen keamanan, baik secara individual maupun saat terintegrasi. Dilakukan analisis menyeluruh terhadap data log

yang dihasilkan dari setiap skenario, termasuk data dari *LoginHistory* dan *AuditLog*, untuk membuktikan secara empiris bahwa model keamanan yang dirancang telah berhasil mematuhi prinsip-prinsip dasar *Zero Trust*.

**Lingkungan Pengembangan Perangkat Lunak**

Serangkaian teknologi modern yang dipilih berdasarkan skalabilitas dan fleksibilitasnya untuk aplikasi web adalah dasar pengembangan prototipe sistem ini. Sistem di sisi server (*backend*) berjalan di lingkungan *runtime* Node.js, dan kerangka kerja Express.js digunakan untuk menangani logika aplikasi dan perutean. Sistem menggunakan *database* relasional MySQL untuk persistensi data; *Sequelize* sebagai *Object-Relational Mapper* (ORM) membantu menghubungkan logika aplikasi dan skema *database* diabstraksi. Antarmuka pengguna dirender secara dinamis di lapisan presentasi (*frontend*) server menggunakan mesin templat EJS (*Embedded JavaScript*). Lapisan keamanan dasar digunakan dengan bcrypt.js untuk *hashing* kredensial pengguna dan jsonwebtoken (JWT) untuk manajemen.

**Tabel 1.** Lingkungan Pengembangan

| Kategori | Teknologi | Deskripsi |
|---|---|---|
| **Bahasa Pemrograman** | JavaScript (Node.js) | Lingkungan runtime untuk eksekusi kode di sisi server. |
| **Kerangka Kerja Backend** | Express.js | Kerangka kerja minimalis untuk membangun server web dan API. |
| **Database** | MySQL | Sistem manajemen database relasional untuk persistensi data. |
| **ORM** | Sequelize | *Object-Relational Mapper* untuk mempermudah interaksi antara aplikasi dan database. |
| **Mesin Template Frontend** | EJS (Embedded Javascript) | Mesin templat untuk merender halaman HTML dinamis di sisi server. |
| **Manajemen Sesi & Token** | Jsonwebtoken(JWT) | Standar industri untuk membuat token akses yang aman. |
| **Keamanan Kredensial** | Bcrypt.js | Library untuk melakukan *hashing* pada password pengguna. |

## C. HASIL DAN PEMBAHASAN

### Implementasi Sistem

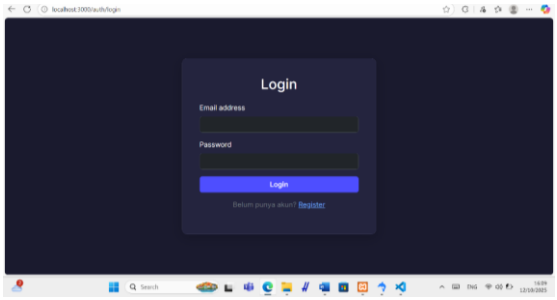
Sistem kontrol akses dinamis berbasis konteks ini dibangun dengan arsitektur modular yang menggunakan Node.js sebagai *backend* utama.

Arsitektur sistem terdiri dari beberapa komponen, seperti modul autentikasi, *middleware* kebijakan keamanan, manajemen data pengguna, dan antarmuka web. *Middleware* auth.js digunakan untuk autentikasi berbasis token JWT, sedangkan policy.js dan stepupauth.js digunakan untuk mengatur kebijakan akses dan memberikan verifikasi tambahan berdasarkan token JWT. Model *Sequelize* yang terhubung ke basis data mencatat aktivitas pengguna, reservasi ruangan, dan histori *login*.

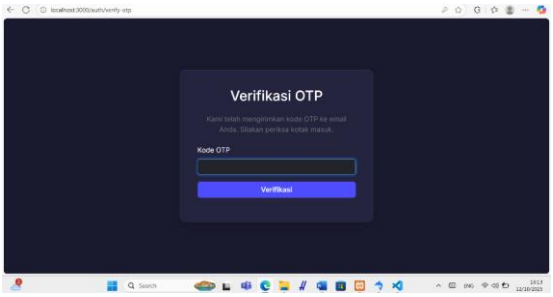
Dari sisi tampilan, sistem dilengkapi dengan antarmuka berbasis web menggunakan *template engine* EJS yang responsif dan mudah dipahami. Pengguna dapat mengakses sistem melalui halaman login sebagai gerbang utama, dilanjutkan dengan proses autentikasi OTP via *e-mail* sebelum diarahkan ke *dashboard* sesuai peran pengguna (admin atau user biasa). Seluruh proses akses dilakukan secara terpusat dan terkontrol sesuai prinsip *Zero Trust Architecture*.

**Halaman Login dan Proses OTP**

Proses autentikasi dimulai pada halaman login, di mana pengguna memasukkan *username* dan *password* mereka. Sistem memeriksa konteks perangkat dan IP setelah kredensial valid. Jika perangkat atau IP tidak terdaftar, pengguna diarahkan ke halaman OTP (*One Time Password*) untuk verifikasi tambahan. OTP dikirim melalui email setelah berhasil. Perangkat dan IP akan dicatat sebagai entitas yang dapat diandalkan. Sebaliknya, jika perangkat/IP sudah terdaftar, proses OTP dilewati, memungkinkan pengguna masuk ke *dashboard* secara instan. Sistem ini membuat *login* rutin lebih cepat tanpa mengurangi keamanan. Selain itu, jika ada akses dari konteks mencurigakan, sistem akan memberikan peringatan. Proses login yang dapat disesuaikan ini sangat penting untuk menerapkan *Zero Trust*.



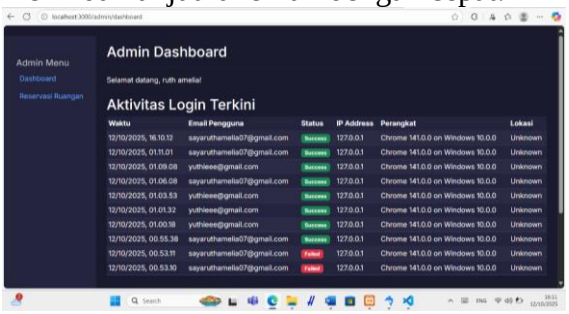

**Gambar 2.** Tampilan Halaman Login




**Gambar 3.** Proses Verifikasi OTP

**Dashboard Admin dan Manajemen Akses**

Setelah proses autentikasi selesai, pengguna yang memiliki otorisasi manajemen diarahkan ke *dashboard* khusus yang berfungsi sebagai pusat kontrol sistem. Tabel aktivitas login yang tampil secara *real time* di *dashboard* ini menunjukkan status *login*, email, IP address, waktu akses, dan perangkat yang terlihat pada gambar 4. Data ini digunakan oleh admin untuk memantau aktivitas mencurigakan dan menindaklanjuti anomali dengan cepat.




**Gambar 4.** Dasboard Admin dan Monitoring Aktivitas

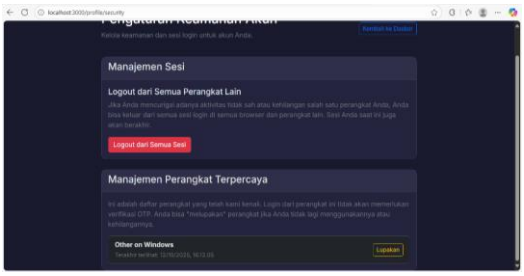
Selain itu, admin memiliki kemampuan untuk mengontrol status akun pengguna melalui fitur *Suspend* dan *Reactivate*. Pada gambar 6 terlihat bahwa Sistem harus meminta Admin untuk membuktikan ulang identitasnya saat ingin menggunakan fitur *suspend* atau *reactive*, dengan memasukkan kembali password mereka. Ini disebut juga *Step-Up Authentication*. Selanjutnya, Akun yang disuspend secara otomatis akan kehilangan hak akses, sehingga jika mencoba login, sistem menolak permintaan tersebut. Meskipun akun yang disuspend memiliki kredensial dan OTP yang valid, fitur ini tidak dapat masuk, menurut pengujian. Ini mendukung prinsip akses *least privilege* dalam arsitektur *Zero Trust*.

**Dashboard pengguna**

Setelah proses login dan OTP selesai, pengguna umum akan melihat *Dashboard* Reservasi Ruangan di sistem. Mereka dapat melakukan pemesanan ruangan dengan mengisi form kegiatan, memilih ruangan, dan menentukan waktu mulai dan selesai. Pengguna memiliki pandangan penuh atas aktivitas mereka dan dapat melihat riwayat reservasi langsung di *dashboard*.

Hasil implementasi menunjukkan bahwa model kontrol akses dinamis berbasis konteks memperkuat autentikasi dan memberi pengguna lebih banyak kontrol atas keamanan akun. Pengguna dapat mengakses pengaturan keamanan akun tambahan di halaman dashboard, yang terdiri dari dua fitur utama:

1. **Manajemen Sesi**, yang memungkinkan pengguna *logout* dari semua perangkat sekaligus, fitur penting untuk mencegah akses tidak sah apabila perangkat masih tersambung setelah sesi aktif.
2. **Manajemen Perangkat Terpercaya**, yang menampilkan daftar perangkat dan IP yang pernah digunakan untuk login. Pengguna dapat menghapus perangkat tertentu dari daftar ini, sehingga pada login berikutnya sistem akan kembali meminta OTP.




**Gambar 5.** Pengaturan Keamanan Akun

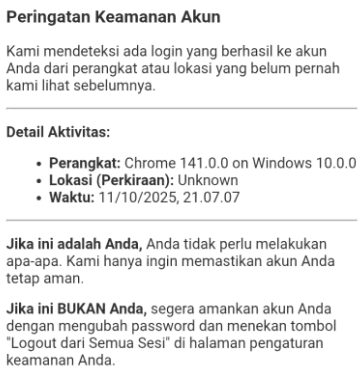
Dengan fitur ini, pengguna memiliki akses langsung ke semua aktivitas yang terjadi di akunnya. Metode ini sejalan dengan prinsip *Zero Trust* "*never trust, always verify*" di mana keamanan bergantung pada pengawasan dan pengelolaan konteks yang berkelanjutan selain pada autentikasi awal. Fitur ini meningkatkan keamanan akun dan memberikan pengalaman pengguna yang lebih aman dan terkontrol, serta memperkuat pertahanan terhadap akses tidak sah.

**Validasi Konteks Akses dan Step-Up Authentication**

Dalam arsitektur *Zero Trust*, verifikasi tidak hanya dilakukan saat login pertama, tetapi juga saat terjadi perubahan konteks akses. Sistem ini menerapkan mekanisme deteksi perangkat dan lokasi menggunakan *user-agent* dan GeoIP. Jika terjadi perbedaan lokasi atau perangkat dari sesi login sebelumnya, *middleware* stepupauth.js secara otomatis meminta verifikasi ulang melalui OTP.

Selain itu, sistem secara otomatis mengirimkan peringatan keamanan melalui *email* ketika ada upaya login dari

perangkat atau lokasi yang belum pernah tercatat sebelumnya. Pemberitahuan ini mencantumkan detail aktivitas seperti perangkat, lokasi, dan waktu login, serta memberikan opsi tindakan cepat seperti mengganti kata sandi atau keluar dari semua sesi.

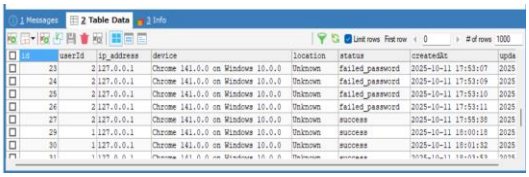



**Gambar 6.** Peringatan Keamanan

Hasil pengujian menunjukkan bahwa ketika pengguna mencoba login dari perangkat yang berbeda atau IP address yang tidak sesuai dengan riwayat login sebelumnya, sistem akan memberi peringatan atau meminta autentikasi tambahan. Hal ini membuktikan bahwa mekanisme kontrol akses dinamis telah berjalan sesuai desain.

**Audit Log dan Monitoring Aktivitas**

Setiap aktivitas pengguna, seperti *login*, *logout*, reservasi, dan upaya akses, dicatat dalam audit log dan login history pada database MySQL. Alamat IP, waktu, status akses, dan perangkat yang digunakan dicatat. Sebagai bagian dari prinsip visibilitas dan analisis *Zero Trust*, audit log ini memungkinkan admin melakukan pelacakan insiden dan analisis keamanan sistem.




**Gambar 7.** Tabel Login History

Hasil uji coba menunjukkan bahwa fitur pencatatan ini membantu proses evaluasi keamanan jangka panjang. Sistem dapat merekam semua aktivitas tanpa mengganggu operasi aplikasi.

**Analisis Penerapan Zero Trust dan Evaluasi Sistem**

Penerapan prinsip *Zero Trust Architecture* dalam sistem ini mencakup tiga pilar utama:

1. **Verifikasi eksplisit**, setiap akses diverifikasi berdasarkan identitas, perangkat, dan konteks lingkungan.
2. **Least privilege access**, akses diberikan sesuai peran dan status pengguna (aktif/nonaktif).
3. **Assume breach**, sistem tidak memberikan kepercayaan implisit bahkan setelah autentikasi berhasil.

Dibandingkan dengan kontrol akses konvensional, sistem ini lebih tahan terhadap perubahan situasi dan memiliki lapisan keamanan tambahan melalui penerapan undang-undang dan OTP. Namun, keterbatasan sistem masih terletak pada ketergantungan pada IP publik yang rentan dilewati VPN dan mekanisme OTP yang masih sederhana. Integrasi MFA yang lebih kuat dan deteksi anomali berbasis ML dapat menjadi fokus pengembangan berikutnya.

**Ringkasan Hasil Pengujian**

**Tabel 2.** Hasil Pengujian

| No. | Komponen Fitur | Hasil Uji | Status |
|---|---|---|---|
| **1** | Login & OTP Adaptif | Autentikasi dua langkah berjalan; OTP hanya diminta saat login dari perangkat/IP baru. | Berhasil |
| **2** | Halaman Login & | Halaman login dan | Berhasil |

| | | | |
|---|---|---|---|
| | Dashboard | navigasi dashboard responsif, proses masuk cepat dan stabil. | |
| **3** | Autentikasi Kontekstual | Sistem mengenali perangkat/IP terpercaya dan menyesuaikan proses autentikasi. | Berhasil |
| **5** | Manajemen Sesi | Pengguna dapat logout dari semua perangkat melalui dashboard keamanan. | Berhasil |
| **6** | Notifikasi Keamanan (Email) | Email peringatan dikirim otomatis saat login dari perangkat/lokasi baru | Berhasil |
| **7** | Manajemen Perangkat Terpercaya | Daftar perangkat/IP tercatat, dapat dihapus untuk memicu OTP ulang pada login berikutnya. | Berhasil |
| **8** | Step-up Authentication | OTP diminta ulang jika konteks berubah (mis. device berbeda). | Berhasil |
| **9** | Audit Log | Seluruh aktivitas login, perangkat, dan perubahan konteks tercatat dalam database. | Berhasil |
| **10** | Suspend & Reactivate Account | Akun tidak dapat login saat status nonaktif; akses pulih setelah diaktifkan Kembali. | Berhasil |
| **11** | GeoIP Detection | Sistem mendeteksi lokasi login, meski akurasi terbatas saat VPN digunakan. | Terbatas |
| **12** | Keamanan Data Login | Data kredensial dan sesi terlindungi, token dihapus otomatis saat logout. | Berhasil |

Pengujian menunjukkan bahwa sistem kontrol akses dinamis yang Anda buat sesuai dengan prinsip *Zero Trust* dan beroperasi dengan baik. Kombinasi OTP adaptif, autentikasi kontekstual, manajemen perangkat, dan notifikasi keamanan memberikan lapisan perlindungan berlapis tanpa mengurangi kenyamanan pengguna. Akurasi GeoIP *Detection* mengalami kesalahan kecil, yang dapat memengaruhi penggunaan VPN, tetapi tidak mengganggu proses autentikasi utama.

## D. KESIMPULAN DAN SARAN

Salah satu cara untuk meningkatkan keamanan akun pengguna adalah dengan menerapkan sistem notifikasi keamanan login melalui email saat terjadi aktivitas dari perangkat atau lokasi yang tidak dikenal. Pengguna dapat menerima peringatan secara *real-time* dengan fitur ini sehingga mereka dapat melakukan hal-hal seperti mengganti kata sandi, menonaktifkan sesi aktif, atau melaporkan kepada administrator jika aktivitas mencurigakan terjadi. Hasil

pengujian menunjukkan bahwa semua fitur utama sistem termasuk verifikasi OTP, autentikasi dua langkah, *suspend* dan reaktivasi akun, pencatatan audit log, dan deteksi lokasi berfungsi dengan baik dalam skenario yang direncanakan. Namun, deteksi GeoIP yang akurat saat pengguna menggunakan VPN memiliki keterbatasan, yang dapat menjadi masalah untuk pengembangan selanjutnya. Oleh karena itu, agar sistem dapat memberikan perlindungan yang lebih baik dan responsif terhadap ancaman keamanan siber, disarankan untuk memperkuat sistem keamanan dengan menambahkan metode autentikasi berlapis, meningkatkan akurasi deteksi lokasi, dan meningkatkan kecepatan dan detail notifikasi keamanan.

## E. REFERENSI


Agustina, E. R., & Achmad, F. (2019). Perancangan spesifikasi keamanan kontrol akses pada aplikasi layanan informasi di lingkungan instansi pemerintah. Jurnal Teknologi Informasi dan Ilmu Komputer, 6(2), 195–200. https://doi.org/10.25126/jtiik.2019621304

Efendi, R., Wahyono, T., & Widiasari, I. R. (2024). Uji kerentanan keamanan pada aplikasi berbasis web menggunakan metode vulnerability assessment. AITI: Jurnal Teknologi Informasi, 21(Maret), 44–57.

Haeruddin, H., Favian, F., & Prasetyo, S. E. (2024). Implementasi zero trust network untuk meningkatkan keamanan jaringan menggunakan Ferrumgate dengan metode NDLC. Infotech: Journal of Technology Information, 10(2), 307–318. https://doi.org/10.37365/jti.v10i2.324

Haeruddin, H., Prasetyo, S. E., & Mindy, A. (2025). Implementasi multi-factor authentication untuk optimalisasi keamanan akses data di PT. ABC. Jurnal Manajemen Informatika (JAMIKA), 15(1), 85096. https://doi.org/10.34010/5rdjmw37

Hose, F., Censaka, F., & Wijaya, R. A. (2025). Analisis peran blockchain dalam zero-trust architecture untuk penguatan identitas digital dan privasi data. JIMU: Jurnal Ilmiah Multidisipliner, 4(1), 568–578.

Junga, D., & Sulistyo, W. (2025). Implementasi port knocking dinamis berbasis waktu pada router untuk pengamanan akses SSH. IT-Explore: Jurnal Penerapan Teknologi Informasi dan Komunikasi, 4(1), 106–115. https://doi.org/10.24246/itexplore.v4i1.2025.pp106-115

Laksono, A. C., & Sari, B. W. (2025). Pengembangan framework tata kelola akses multi-tenant untuk mitigasi ancaman insider di cloud publik. Jurnal Informatika Teknologi dan Sains (JINTEKS), 7(3), 1520–1527. https://doi.org/10.51401/jinteks.v7i3.6574

Lubis, A. A., Pinem, J., Lubis, M. A. S., & Kiswanto, D. (2022). Implementasi Roundcube pada mail server untuk lingkungan Program Studi Ilmu Komputer

UNIMED. Blend Sains Jurnal Teknik, 1(3), 194–201. https://doi.org/10.56211/blendsains.v1i3.163

Mugianto, D. R., & Budiarto, R. (2024). Evaluasi pengujian keamanan arsitektur zero trust network pada jaringan smart home untuk mengatasi serangan data sniffing. Syntax Literate: Jurnal Ilmiah Indonesia, 9(11), 6703–6718. https://doi.org/10.36418/syntax-literate.v9i11.16898

Mukhlisin, M., & Agung Firmansyah, R. (2025). Zero trust architecture: Solusi keamanan dan privasi untuk institusi pendidikan, systematic literature review. JATI (Jurnal Mahasiswa Teknik Informatika), 9(4), 6926–6935. https://doi.org/10.36040/jati.v9i4.14344

Pananto, R. C., & Damayanti, S. (2023). Jurnal ilmu komputer. Biomaterials, 7(12), 85–90.

Pribadi Fitrian, H., Nur Aziz Bisri, F., Willy Al Fathir, M., & Jaisy Hizbulloh, M. (2024). Analisis keamanan cloud dengan zero trust dan blockchain yang tangguh. Journal Global Tecnology Computer, 4(1), 36–43. https://doi.org/10.47065/jogtc.v4i1.6432

Zuhriyanto, I., & Astari, S. R. (2025). Penerapan zero trust architecture untuk mitigasi ancaman pembajakan akun WhatsApp. JITU: Journal Informatic Technology and Communication, 9(1), 50–58. https://doi.org/10.36596/jitu.v9i1.1815