OPTIMASI SISTEM KEAMANAN REST API MENGGUNAKAN *ZERO TRUST ARCHITECTURE* DAN *KEYCLOAK TOOLS* PADA INFRASTRUKTUR BERBASIS *MICROSERVICES*

Disusun Oleh:

Denny Afrizal

22206052001

PROGRAM STUDI INFORMATIKA

PROGRAM MAGISTERFAKULTAS SAINS DAN TEKNOLOGI

UIN SUNAN KALIJAGA

YOGYAKARTA

2025

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-109/Un.02/DST/PP.00.9/01/2025

Tugas Akhir dengan judul : Optimasi Sistem Keamanan REST API menggunakan Zero Trust Architecture dan Keycloak Tools pada Infrastruktur berbasis Microservice

yang dipersiapkan dan disusun oleh:

Nama : DENNY AFRIZAL, S.Kom
Nomor Induk Mahasiswa : 22206052001
Telah diujikan pada : Senin, 06 Januari 2025
Nilai ujian Tugas Akhir : A-

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR

Ketua Sidang

Ir. Muhammad Taufiq Nuruzzaman, S.T. M.Eng., Ph.D.
SIGNED
Valid ID: 678a2799c35b1

Penguji I

Dr. Ir. Bambang Sugiantoro, M.T., IPU., ASEAN Eng.
SIGNED
Valid ID: 6789fdf9a7713

Penguji II

Dr. Agung Fatwanto, S.Si., M.Kom.
SIGNED
Valid ID: 678a00bb8a8e1

Yogyakarta, 06 Januari 2025
UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi

Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED
Valid ID: 678dc76026f7c

PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama	: Denny Afrizal
NIM	: 22206052001
Jenjang	: Magister
Program Studi	: Informatika

menyatakan bahwa naskah tesis ini secara keseluruhan adalah hasil penelitian/karya saya sendiri, kecuali pada bagian-bagian yang di rujuk sumbernya.

Yogyakarta, 6 Januari 2025

Saya yang menyatakan

Denny Afrizal

NIM. 22206052001

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama	: Denny Afrizal
NIM	: 22206052001
Jenjang	: Magister
Program Studi	: Informatika

menyatakan bahwa naskah tesis ini secara keseluruhan benar-benar bebas dari plagiasi. Jika di kemudian hari terbukti melakukan plagiasi, maka saya siap ditindak sesuai ketentuan hukum yang berlaku.

Yogyakarta, 6 Januari 2025

Saya yang menyatakan,

Denny Afrizal

NIM. 22206052001

NOTA DINAS PEMBIMBING

PERSETUJUAN TUGAS AKHIR

Kepada Yth.

Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

Assalamu'alaikum wr. wb.

Disampaikan dengan hormat. Setelah melakukan bimbingan, arahan, dan koreksi terhadap penulisan tesis yang berjudul:

OPTIMASI SISTEM KEAMANAN REST API MENGGUNAKAN ZERO TRUST ARCHITECTURE DAN KEYCLOAK TOOLS PADA INFRASTRUKTUR BERBASIS MICROSERVICES

Yang ditulis oleh:

Nama	: Denny Afrizal
NIM	: 22206052001
Jenjang	: Magister
Program Studi	: Informatika

Saya berpendapat bahwa tesis tersebut sudah dapat diajukan kepada Magister Informatika (S2) UIN Sunan Kalijaga untuk diujikan dalam rangka memperoleh gelar Magister Informatika.

Wassalamu'alaikum wr. wb.

Yogyakarta, 15 Januari 2025

Pembimbing

Ir. Muhammad Taufiq Nuruzzaman,

S.T. M.Eng., Ph.D.

NIP. 19791118 200501 1 003

ABSTRAK

Di era digital yang semakin canggih, keamanan sistem informasi menjadi tantangan utama, khususnya dalam pengelolaan layanan berbasis *REST API* (*Representational State Transfer*) yang sering digunakan dalam arsitektur *microservices*. Dalam praktiknya, *REST API* menggunakan *JSON* (*JavaScript Object Notation*) untuk pertukaran data antar *API* (*Application Programming Interface*). Tesis ini mengeksplorasi implementasi *Zero Trust Architecture* (ZTA) sebagai strategi keamanan untuk melindungi *REST API* dari ancaman yang berpotensi merugikan. Konsep *Zero Trust* berfokus pada prinsip "jangan percaya, selalu verifikasi", yang mengharuskan setiap akses ke sistem, baik dari dalam maupun luar jaringan, harus divalidasi terlebih dahulu. Penelitian ini dilakukan melalui metode kajian literatur, penyusunan instrumen, pengumpulan data, serta analisis data. Hasilnya, penulis juga membuat proyek keamanan *REST API* menggunakan bahasa *Java* dengan *Spring Boot Framework* serta menggunakan *Keycloak* untuk teknologi pendukung sebagai studi kasus penerapan ZTA. Melalui analisis skenario nyata dan simulasi serangan, hasil dari penelitian ini menunjukkan bahwa penerapan ZTA dapat secara signifikan meningkatkan keamanan *REST API*, mengurangi risiko serangan seperti *injeksi SQL*, dan membantu dalam menjaga integritas data. Selain itu, penelitian ini juga memberikan rekomendasi praktis untuk pengembang dan arsitek sistem dalam menerapkan ZTA untuk meningkatkan ketahanan aplikasi terhadap berbagai ancaman.

Kata Kunci: ***Zero Trust Architecture, REST,* Keamanan, *Microservices,* Serangan Siber, *API, JSON*.**

ABSTRACT

In the increasingly advanced digital era, information system security has become a major challenge, particularly in the management of REST API-based services, which are commonly used in microservices architecture. In practice, REST API uses JSON (JavaScript Object Notation) for data exchange between APIs (Application Programming Interfaces). This thesis explores the implementation of Zero Trust Architecture (ZTA) as a security strategy to protect REST API from potential threats. The Zero Trust concept focuses on the principle of "never trust, always verify", which requires every access to the system, both from within and outside the network, to be validated first. This research was conducted through literature review, instrument development, data collection, and data analysis. The result is that the author also developed a security project for REST API using Java with Spring Boot Framework and Keycloak as a supporting technology as a case study for ZTA implementation. Through real-world scenario analysis and simulation attacks, the results of this research show that the implementation of ZTA can significantly increase the security of REST API, reduce the risk of attacks such as SQL injection, and help maintain data integrity. Furthermore, this research also provides practical recommendations for developers and system architects to implement ZTA to increase application resilience against various threats.

Keywords: ***Zero Trust Architecture, REST, Security, Microservices, Cyber Attack, API, JSON.***

MOTTO

“Kebenaran adalah tujuan utama ilmu pengetahuan”

[Al-Khawarizmi]

“Kehidupan yang paling baik adalah kehidupan yang dihabiskan untuk memahami kebenaran”

[Ibnu Bajjah]

KATA PENGANTAR

Bismillahirahmanirrahim,

Assalamualaikum warahmatullahi wabarakatuhu

Puji syukur penulis panjatkan ke hadirat Allah SWT yangtelah memberikan segala keindahan dan petunjuk-Nya, sehingga penulis dapat menyelesaikan tesis ini dengan judul "Optimasi Sistem Keamanan REST API Menggunakan *Zero Trust Architecture* dan *Keycloak Tools* Pada Infrastruktur Berbasis *Microservices*" sebagai salah satu persyaratan mencapai gelar Magister Informatika.

Penulis ingin mengucapkan terima kasih kepada semua orang yang telah memberikan penghiburan, dukungan, dan pemikiran kreatif sehingga perencanaan proposal ini dapat diselesaikan sedikit demi sedikit. Terima kasih yang luar biasa diberikan kepada:

1. Bapak Prof. Noorhaidi, S.Ag., M.A., M.Phil., Ph.D., selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Ibu Prof. Dr. Dra. Hj. Khurul Wardati, M.Si., selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Bapak Dr. Ir. Sumarsono, S.T., M.Kom., selaku Ketua Program Studi Magister Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Bapak Ir. Muhammad Taufiq Nuruzzaman, S.T. M.Eng., Ph.D. selaku dosen Pembimbing Tesis yang telah memberikan motivasi dan pengarahanselama studi sehingga dapat menyelesaikan penyusunan tesis ini.
5. Bapak dan Ibu dosen Program Studi Magister Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta yang sudah membagi ilmu yang sangat bermanfaat.
6. Istri saya, Fitri Hurul Ain, Orang tua saya, M. Hasan dan Mini, serta anak-anak saya, Muhammad Adnan Al-Fatih dan Muslimah Alifah Almahyra yang senantiasa memberikan doa serta dukungan selama studi di Magister Informatika Universitas Islam Negeri Sunan Kalijaga Yogyakarta.
7. Seluruh Staf Karyawan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga Yogyakarta yang telah membantu sehingga penyusunan tesis ini berjalan lancar.

8. Teman-teman Magister Informatika angkatan 2022 Universitas Islam Negeri Sunan Kalijaga atas kerjasama, saran, dan bantuannya.
9. Semua pihak yang tidak bisa penulis sebutkan satu persatu atas bantuannya dalam penyelesain tesis ini.

Akhir kata, penulis hanya bisa bersyukur kepada Allah SWT, berharap apa yang telah dilakukan selama ini bisa menjadi amal dan persembahan berikut. Penulis menyadari bahwa dalam proses penulisan tesis ini masih terdapat banyak kesalahan dan kekurangan, dan sangat mengharapkan kritik dan saran untuk perbaikan. Semoga skripsi ini dapat bermanfaat bagi penulis khususnya dan pembaca pada umumnya, terima kasih.

Wassalamualaikum warahmatullahi wabarakatuh

Yogyakarta, 6 Januari 2025

Denny Afrizal
22206052001

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN

API	: Application Programming Interface
HTTP	: HyperText Transfer Protocol
IAM	: Identity and Access Management
JPA	: Java Persistence API
JRE	: Java Runtime Environment
JSON	: JavaScript Object Notation
JVM	: Java Virtual Machine
JWT	: JSON Web Token
LDAP	: Lightweight Directory Access Protocol
MFA	: Multi Factor Authentication
MiTM	: Man in The Middle
OOP	: Object-Oriented Programming
RBAC	: Role Based Access Control
REST	: REpresentational State Transfer
SAML	: Security Assertion Markup Language
SQL	: Structure Query Language
SSO	: Single Sign On
URI	: Uniform Resource Identifier
XML	: Extensible Markup Language
XSS	: Cross Site Scripting
ZTA	: Zero Trust Architecture

BAB I

PENDAHULUAN

A. Latar Belakang

Pada era transformasi digital saat ini, banyak organisasi yang beralih dari arsitektur monolitik ke arsitektur *microservices* untuk mendukung inovasi dan efisiensi operasional. *Microservices* adalah gaya arsitektur di mana komponen-komponen dari suatu sistem dirancang sebagai layanan yang dapat dideploy secara independen. *Microservices* dirancang berdasarkan subdomain bisnis yang terdefinisi dengan baik, dan mereka berkomunikasi satu sama lain menggunakan protokol ringan, seperti HTTP (Peralta, 2023). *Microservices* menawarkan fleksibilitas dan skalabilitas dengan memecah aplikasi besar menjadi layanan-layanan kecil yang independen. Layanan-layanan ini berkomunikasi satu sama lain melalui REST API (*REpresentational State Transfer Application Programming Interface*), yang memungkinkan pertukaran data secara cepat dan efisien. REST merupakan jenis *web* service yang menerapkan konsep perpindahan antar *state*. *State* dapat digambarkan seperti jika *browser* meminta suatu halaman web, maka server akan mengirimkan *state* halaman web yang diminta ke *browser*. (Yusril, 2023). Sedangkan API (*Application Programming Interface*) adalah konsep krusial dalam dunia pengembangan perangkat lunak yang memungkinkan berbagai program atau aplikasi berkomunikasi dan berinteraksi satu sama lain. (Wiguna, 2023). Dalam praktiknya, pertukaran data pada REST API umumnya menggunakan data dalam bentuk JSON (*JavaScript Object Notation*) karena ringkas, lebih fleksibel dalam pertukaran format data, mudah

dibaca, proses serialisasi dan deserialisasi yang cepat, serta didukung oleh berbagai macam bahasa pemrograman.

Namun, di balik keunggulan yang ditawarkan oleh arsitektur *microservices*, muncul tantangan keamanan yang semakin kompleks. Arsitektur ini memperbanyak titik-titik komunikasi yang rentan terhadap serangan siber, seperti MiTM (*man-in-the-middle*), *injection attacks*, atau pencurian identitas. Dengan semakin tingginya volume data yang ditransfer antar-layanan dan melalui jaringan, keamanan REST API menjadi aspek krusial untuk menjamin integritas dan kerahasiaan data, serta untuk memastikan tidak ada layanan yang disusupi oleh pihak tidak bertanggung jawab. Karena konsep dari REST API adalah bahwa server menyediakan sumber daya yang dapat diakses dan dimanipulasi oleh klien melalui permintaan HTTP (*HyperText Transfer Protocol*). Setiap sumber daya direpresentasikan dalam bentuk URI (*Uniform Resource Identifier*) yang unik. Melalui metode HTTP seperti *GET, POST, PUT,* dan *DELETE,* klien dapat berinteraksi dengan sumber daya tersebut untuk melakukan operasi seperti membaca, menambah, memperbarui, atau menghapus data. (Febriansyah, 2023).

Pendekatan keamanan tradisional yang mengandalkan keamanan perimeter tidak lagi memadai dalam menghadapi ancaman ini. Model lama yang menganggap semua perangkat atau pengguna dalam jaringan internal dapat dipercaya sudah tidak relevan, terutama dengan semakin tingginya penggunaan teknologi berbasis *cloud*, jaringan yang terdistribusi, dan kerja jarak jauh (*Remote Working*). Oleh karena itu, muncul pendekatan baru yang dikenal sebagai ZTA yang menekankan bahwa tidak ada entitas yang dapat dipercaya secara otomatis, baik yang berada di dalam maupun di luar jaringan.

Pendekatan ini didasarkan pada prinsip "*never trust, always verify*" di mana setiap permintaan akses harus selalu diverifikasi secara menyeluruh tanpa asumsi kepercayaan. ZTA umumnya berkaitan dengan arsitektur keamanan siber yang didasarkan pada prinsip-prinsip zero trust. ZTA dirancang khusus untuk membatasi pergerakan lateral internal dan mencegah kebocoran data. Arsitektur ini menerapkan otorisasi dan autentikasi identitas yang ketat dan menghilangkan kepercayaan implisit. ZTA didasarkan pada tujuh prinsip, yaitu data, perangkat, pengguna, otomatisasi dan orkestrasi, jaringan dan lingkungan, visibilitas dan analitik, serta aplikasi dan beban kerja. (dez, 2024).

Penelitian ini dilakukan untuk menganalisis dan menguji efektivitas ZTA dalam mengamankan REST API pada infrastruktur *microservices*. Dengan mengintegrasikan ZTA ke dalam arsitektur keamanan API, diharapkan akan terbentuk lapisan keamanan yang lebih kuat dan mampu menghadapi ancaman yang semakin berkembang di dunia siber.

Penggunaan REST API dalam konteks *microservices* menawarkan banyak manfaat, seperti skala independent berdasarkan kebutuhan sumber daya dan permintaan pengguna untuk setiap layanan, pengembangan secara bersama-sama sehingga lebih cepat dan efisien dalam prosesnya, serta dapat dijalankan di berbagai jenis *platform.*

Namun, sifat REST API yang memungkinkan komunikasi lintas-jaringan juga meningkatkan risiko keamanan. Setiap titik akses API dapat menjadi target bagi penyerang yang mencari celah keamanan untuk mengeksploitasi sistem. Ini menempatkan keamanan API sebagai salah satu tantangan utama dalam infrastruktur *microservices* yang semakin kompleks.

Meningkatnya penggunaan REST API dalam arsitektur *microservices* memperbanyak jumlah titik akses yang harus diamankan. Beberapa tantangan keamanan utama yang dihadapi seperti banyaknya titik akses, autentikasi dan otorisasi yang lemah dan kemungkinan tidak tepat, serangan injeksi, dan serangan.MiTM.

ZTA hadir sebagai jawaban terhadap tantangan-tantangan keamanan yang dihadapi REST API dalam *microservices*. ZTA adalah paradigma yang dipromosikan oleh *National Institute of Standards and Technology* (NIST) dalam sebuah publikasi khusus yang didedikasikan pada konsep inti " *never trust, always verify* ". Konsep ini dijabarkan dalam beberapa prinsip kunci, seperti autentikasi terus-menerus, kontrol akses yang ketat, prinsip hak akses minimum, pemantauan, dan segmentasi. Standar ini juga menawarkan beberapa arsitektur referensi tingkat tinggi untuk mengimplementasikan prinsip ZTA di lingkungan yang sudah ada. Ada sejumlah tantangan dalam adopsi dan migrasi ke arsitektur ZTA, salah satunya adalah kompleksitas manajemen keamanan, terutama seiring pertumbuhan infrastruktur (evaluasi risiko, konfigurasi, dan manajemen perubahan siklus hidup) (Federici, 2023).

Pendekatan ini mematahkan asumsi bahwa jaringan internal dapat dipercaya secara otomatis. Dengan menerapkan ZTA, setiap permintaan akses baik dari dalam maupun luar jaringan harus melalui proses verifikasi identitas dan validasi hak akses secara menyeluruh. Dalam proses implementasi ZTA, tentunya membutuhkan *Tools* pndukung. Dalam hal ini, penulis menggunakan *Keycloak* sebagai alat bantu demi tercapainya tujuan penelitian. *Keycloak* adalah alat Manajemen Identitas dan Akses sumber terbuka yang berfokus pada aplikasi modern seperti aplikasi satu halaman, aplikasi seluler, dan

API REST. Proyek ini dimulai pada tahun 2014 dengan fokus yang kuat untuk memudahkan pengembang mengamankan aplikasi mereka. Sejak saat itu, *Keycloak* telah berkembang menjadi proyek sumber terbuka yang mapan dengan komunitas dan basis pengguna yang kuat. Keycloak digunakan dalam produksi untuk skenario yang berkisar dari situs web kecil dengan hanya beberapa pengguna hingga perusahaan besar dengan jutaan pengguna (Thorgersen, 2021).

Prinsip-prinsip utama ZTA yang relevan untuk REST API dalam *microservices* seperti akses berbasis identitas, segmentasi mikro, enkripsi *end-to-end,* serta deteksi berkelanjutan.

Terdapat beberapa alasan mengapa penelitian ini penting dan relevan untuk dilakukan seperti meningkatnya serangan siber, kebutuhan keamanan yang lebih kuat pada skema *microservices*, ketidak efektifan model keamanan tradisional, serta peningkatan pemahaman terkait ZTA itu sendiri.

Berdasarkan pemaparan di atas, penerapan ZTA dalam mengamankan REST API di infrastruktur *microservices* merupakan pendekatan yang relevan dan penting di tengah meningkatnya ancaman siber. Dengan prinsip "*never trust, always verify*", ZTA memberikan fondasi yang lebih kuat dalam melindungi data, layanan, dan pengguna di lingkungan arsitektur yang semakin kompleks dan terdistribusi. Penelitian ini bertujuan untuk mengkaji efektivitas ZTA dalam memperkuat keamanan REST API, serta memberikan kontribusi dalam pengembangan model keamanan modern yang lebih adaptif dan tangguh di masa depan.

B. Rumusan Masalah

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

1. Bagaimana implementasi ZTA pada REST API pada infrastruktur *microservice* dapat meningkatkan keamanan sistem.
2. Komponen-komponen apa saja yang diperlukan dalam implementasi ZTA pada REST API.
3. Bagaimana efektivitas implementasi ZTA pada REST API dalam menghadapi ancaman keamanan.

C. Batasan Masalah

Batasan masalah merupakan ruang lingkup dari permasalahan yang akan dibatasi dalam penelitian mengenai implementasi ZTA pada REST API pada Mahasiswa Informatika maupun masyarakat umum. Adapun Batasan masalah yang sudah ditetapkan dalam penelitian ini, antara lain:

1. Penelitian ini tertuju kepada mahasiswa maupun masyarakat umum yang sedang mengembangkan aplikasi.
2. Penelitian ini menguji kerentanan pada saat proses pertukaran data pada aplikasi terutama data yang bersifat sensitif.
3. Parameter yang diuji adalah tingkat keamanan dari praktikum yang diberikan dengan beberapa faktor seperti autentikasi, otorisasi, kemudahan pengamanan, serta deteksi ancaman siber.
4. Penelitian ini termasuk dalam penelitian kuantitatif dan teknik pengumpulan datanya menggunakan pendekatan *experimental*.

5. Penelitian ini juga membuat produk keamanan menggunakan bahasa pemrograman *Java* dengan *Spring Boot* sebagai *framework* dan *Keycloak* sebagai alat autentikasi dan otorisasi.

D. Tujuan Penelitian

Berdasarkan rumusan masalah dan Batasan masalah di atas, maka tujuan dari penelitian ini adalah melakukan optimasi keamanan pada REST API menggunakan konsep ZTA pada proses pengembangan aplikasi khususnya pengambangan API yang sangat mungkin berisi data sensitif dengan metode kuantitatif dan teknik pengumpulan datanya menggunakan pendekatan *experimental.*

E. Manfaat Penelitian

Dalam hasil penelitian yang telah dilakukan, diharapkan akan memberikan manfaat sebagai berikut:

1. Memberikan informasi terhadap hasil implementasi ZTA pada pengembangan *backend* aplikasi berbasis REST API.
2. Meningkatkan keamanan dalam proses pertukaran data karena melakukan pendekatan yang lebih ketat dalam proses autentikasi dan otorisasi.
3. Memberikan deteksi keamanan melalui pemantauan ancaman secara berkelanjutan pada aplikasi.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa Penerapan ZTA dalam konteks *microservices* terbukti memberikan peningkatan signifikan terhadap keamanan REST API. Dengan menggunakan *Keycloak* sebagai penyedia identitas terpusat dan *Redis* untuk penyimpanan sementara *code verifier* dalam skema PKCE (*Proof Key for Code Exchange*), pendekatan ini memperkuat keamanan sekaligus mendukung fleksibilitas layanan berbasis *microservices*.

1. *Keycloak* memungkinkan pengelolaan otorisasi dan autentikasi di seluruh *microservices* secara terpusat, memastikan setiap layanan hanya menerima permintaan yang telah tervalidasi sementara *Redis*, melalui penyimpanan sementara *code verifier*, memperkuat mekanisme PKCE untuk mencegah serangan *authorization code interception* pada proses authentikasi.
2. Dengan pendekatan ZTA, setiap *microservice* dianggap sebagai entitas terpisah tanpa kepercayaan antar-layanan. Validasi dilakukan melalui token yang diverifikasi di *Keycloak.* Penggunaan *service-to-service authentication* memastikan bahwa komunikasi antar-*microservices* dienkripsi dan diautentikasi, mencegah serangan MiTM atau penyalahgunaan akses internal.
3. *Redis* sebagai *cache* memberikan latensi rendah untuk validasi data autentikasi sementara dan ZTA mendukung arsitektur *microservices* yang elastis karena aturan kebijakan akses dapat diterapkan pada skala besar tanpa penurunan performa signifikan.

B. Saran

Untuk mengoptimalkan penerapan ZTA pada REST API dalam lingkungan microservices, berikut adalah saran dan rekomendasi untuk penelitian selanjutnya:

1. Perlu dilakukan implementasi terhadap fitur-fitur *Keycloak* yang lainnya untuk mengoptimalkan keamanan REST API.
2. Perlu diaktifkan sistem enkripsi pada konfigurasi koneksi yang ada pada aplikasi dan *Redis* agar *value-value* konfigurasi tidak tersebar luas.
3. Perlu dilakukan pengembangan sistem yang dibuat menggunakan teknologi terbaru agar bisa tetap relevan digunakan di masa mendatang.
4. Perlu dilakukan audit terhadap kebijakan akses di *Keycloak* untuk memastikan tidak ada konfigurasi yang rentan atau tidak relevan.

DAFTAR PUSTAKA

Ahmadi, Sina. 2024. *Zero Trust Architecture in Cloud Networks: Application, Challenges and Future Opportunities*. Journal of Engineering Research and Reports, 26(2), 215–228. https://doi.org/10.9734/jerr/2024/v26i21083

Akhunzada, A., et al. 2020. *Security and Privacy in Internet of Things: Challenges and Solutions*. IEEE Internet of Things Journal.

Alfardan, N., & Paterson, K. G. 2021. *A Security Analysis of JSON Web Token and JSON Object Signing and Encryption Standards*. *Journal of Network Security*, 10(2), 90–101.

Almeida, J. P., Lu, X., Silva, A. R., & van Sinderen, M. 2020. *OAuth 2.0 and Beyond: Exploring Security and Interoperability Challenges*. *Journal of Network and Computer Applications*, 162, 102663.

Bertino, E., Sandhu, R., & Paci, F. 2021. *Handbook of Identity and Access Management: Methods, Tools, and Techniques*. Springer International Publishing.

Bi, S., Huang, L., & Zhang, Y. 2020. *Joint Optimization of Service Caching Placement and Computation Offloading in Mobile Edge Computing Systems.* IEEE Transactions on Wireless Communications, 19(7), 4947-4963.

Bourhis, Pierre, Reutter, Juan L., Vrgoč , Domagoj. 2020. *JSON: Data model and query languages*, Information Systems, 89, 101478, https://doi.org/10.1016/j.is.2019.101478.

Chen, W., Zhang, Z., & Liang, C. 2019. *Secure and Scalable Authentication and Authorization for Microservices using JWT in Cloud*. *IEEE Access*, 7, 137844–137855.

Divyabharathi, D. N., and Nagaraj G. Cholli. 2020. "*A review on identity and access management server (keycloak)*." International Journal of Security and Privacy in Pervasive Computing (IJSPPC) 12.3, 46-53.

Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. 2019. *Microservices: Yesterday, Today, and Tomorrow*. Journal of Systems and Software, 144, 43-68.

Febriansyah, Fadil, Awangga, Rolly Maulana. 2023. *Membangun RESTful API dengan Go*. Penerbit Buku Pedia.

Federici, F.; Martintoni, D. Senni, V. *A Zero-Trust Architecture for Remote Access in Industrial IoT Infrastructures*. Electronics 2023, 12, 566. https://doi.org/10.3390/electronics12030566

Fernandez B, Brazhuk A. 2024. *A Critical Analysis of Zero Trust Architecture (ZTA), Computer Standards & Interfaces*. 2024;103832.

Fett, D., Küsters, R., & Schmitz, G. 2019. *The Web Security Model: OAuth 2.0, OpenID Connect, and Beyond*. *Journal of Computer Security*, 27(5), 673–706.

Fielding, Roy T. 2000. *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine.

Fowler, M., & Lewis, J. 2020. *Microservices a Definition of This New Architectural Term.* martinfowler.com.

Jones, M., Bradley, J., & Sakimura, N. 2020. *JSON Web Token (JWT): RFC 7519 Implementation and Security Considerations*. *Internet Society*.

Kuppusamy, V., Manoharan, S., & Velusamy, K. 2019. *An Analysis on OAuth 2.0 Security Architecture for Web Services*. *International Journal of Web and Services Computing*, 10(2), 85–94.

Kurniawan, F., & Kusuma, S. 2019. *The Implementation of IAM in Cloud and Multi-Organizational Systems*. *International Journal of Cloud Computing and Applications*, 4(1), 45-59.

Ma S-P, Hsu M-J, Chen H-J, Lin C-J. 2023. *RESTful API Analysis, Recommendation, and Client Code Retrieval*. Electronics. 12(5):1252. https://doi.org/10.3390/electronics12051252

Misra, P., & Gupta, S. 2020. *Challenges in IAM for Cloud and Hybrid Environments*. *Journal of Information Security and Applications*, 10(3), 112-125.

Ometov, A., Bezzateev, S., et al. 2021. *Multi-Factor Authentication: A Survey and Challenges in V2X Applications*. MDPI Cryptography.

Oracle. 2020. *Java: The Language of Enterprise Application Development*. Oracle Corporation.

Patel, M., & Aggarwal, N. 2021. *Exploring the Security of JSON Web Tokens (JWT) in Modern Web Applications*. *International Journal of Advanced Computer Science and Applications*, 12(6), 35–42.

Peralta, J. H. 2023. *Microservice APIs: Using Python, Flask, FastAPI, OpenAPI and More*. Amerika Serikat: Manning.

Petrillo, F., & et al. 2020. *Which RESTful API Design Rules Are Important and How Do They Improve Software Quality? A Delphi Study with Industry Experts*. arXiv

Rafique, M., & Humayun, M. 2020. *Comprehensive Survey on Identity Management and Access Control Techniques*. *IEEE Access*, 8, 23910–23930.

Richer, J., & Sanso, A. 2020. *OAuth 2 in Action*. Manning Publications.

RightScale 2020. *State of the Cloud Report*.

Schwartz, J., & Hogan, M. 2021. *Zero Trust and Identity Management in Modern IT Security Strategies*. *ACM Computing Surveys*, 53(4), 1-26.

Setyawan, M. Yusril Helmi, Syuhada, Esadhira Giovany. 2023. *Pengembangan Dashboard Laporan Bulanan Untuk Monitoring Kinerja Perusahaan*. Penerbit Buku Pedia.

Suzhi, B., et al. 2020. *Caching Strategies for Mobile Edge Computing.* Proceedings of IEEE.

Thönes, J. 2019. *Microservices: Grundlagen flexibler Softwarearchitekturen.* dpunkt.verlag.

Thorgersen, Stian, Silva, Pedro Igor. 2021. *Keycloak – Identity and Access Management for Modern Applications.* Packt Publishing.

Wang, R., & Chen, M. 2020. *Optimizing Memory Caching in Distributed Systems.* Journal of Cloud Computing.

Wiguna, I Komang Arya Ganda, Desnanjaya, I Gusti Made Ngurah. 2023. *Konsep API dan Implementasinya dalam Membangun Sistem Informasi menggunakan Laravel*. PT. Sonpedia Publishing Indonesia.

Yellafula, Naren. 2017. *Building RESTful Web Service with Go.* Packt Publishing Ltd.

Yuniati, R., Sidiq, A. 2020. *Penggunaan Tanda Tangan Digital di Sektor Pelayanan Publik*. Jurnal Teknologi Informasi.