Biner : Jurnal Ilmu Komputer , Teknik dan Multimedia
Volume 3, No. 3 Agustus 2025
ISSN 2828-2442 (media online)
Hal 267-274

Implementasi Json Web Token (JWT) Untuk Sistem Autentikasi Dan Otorisasi Pada Aplikasi Android Native Dan Spring Boot Di CV Karya Belia Nusantara

Mohamad Rizki Aditiya, Rendy Dwi Putra Setiadi, Banyu Santoso
[1]Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Kota Tangerang Selatan
rizkidit334@gmail.com[1] ; rendydwiputra43@gmail.com[2] ; banyusantoso01@gmail.com[3]

Abstrak-Keamanan informasi menjadi aspek krusial dalam pengembangan aplikasi modern, terutama pada sistem yang menangani data pengguna. Penelitian ini mengimplementasikan sistem autentikasi dan otorisasi berbasis JSON Web Token (JWT) pada aplikasi Android Native dengan backend Spring Boot di CV Karya Belia Nusantara. Penelitian ini bertujuan untuk meningkatkan sistem keamanan yang sebelumnya rentan terhadap penyalahgunaan akun dan pencurian kredensial. Penelitian dilakukan melalui metode wawancara, studi literatur, perancangan sistem, pengujian black box, dan implementasi teknologi terkini seperti OTP, JWT, dan EncryptedSharedPreferences. Hasil pengujian menunjukkan sistem berjalan efektif dalam menangani sesi pengguna, validasi token, dan pemblokiran akses ilegal. Temuan ini memperkuat relevansi JWT sebagai solusi autentikasi yang aman, efisien, dan scalable.

Kata Kunci: JSON Web Token, autentikasi, otorisasi, Spring Boot, Android Native, OTP.

Abstract-Information security is a crucial aspect of modern application development, particularly in systems handling user data. This study implements a JSON Web Token (JWT)-based authentication and authorization system in a native Android application with a Spring Boot backend at CV Karya Belia Nusantara. The objective is to strengthen a previously vulnerable system that was susceptible to account misuse and credential theft. The research was carried out using interviews, literature review, system design, black-box testing, and the adoption of up-to-date technologies such as OTP, JWT, and EncryptedSharedPreferences. Test results demonstrate that the system effectively manages user sessions, validates tokens, and blocks illicit access. These findings underscore the relevance of JWT as a secure, efficient, and scalable authentication solution.

Keywords: JSON Web Token, authentication, authorization, Spring Boot, native Android, OTP

1. PENDAHULUAN

Di era digital saat ini, keamanan informasi menjadi aspek yang krusial dalam pengembangan sistem, terutama pada aplikasi yang terhubung dengan internet. Sistem autentikasi dan otorisasi berperan penting dalam memastikan integritas, kerahasiaan, serta ketersediaan data agar terhindar dari akses yang tidak sah. Pendekatan autentikasi modern berbasis **JSON Web Token (JWT)** menawarkan solusi yang **stateless**, efisien, dan skalabel karena tidak memerlukan penyimpanan sesi di server. JWT memungkinkan backend untuk memverifikasi identitas pengguna dan memberikan otorisasi hanya berdasarkan token yang dikirim, sehingga proses autentikasi menjadi lebih cepat dan ringan.

Namun, sistem autentikasi berbasis token saja belum sepenuhnya aman dari ancaman seperti pencurian kredensial atau token. Oleh karena itu, penerapan **One-Time Password (OTP)** sebagai faktor kedua autentikasi menjadi langkah strategis untuk memperkuat lapisan keamanan. OTP dapat berupa kode berbasis waktu (TOTP) maupun per permintaan (on-demand) yang hanya berlaku dalam periode singkat, sehingga meskipun kredensial utama terekspos, risiko akses ilegal dapat diminimalkan. Di sisi pengguna Android, keamanan token semakin ditingkatkan dengan memanfaatkan **EncryptedSharedPreferences**, yang menyimpan data secara terenkripsi sehingga sulit diekstrak oleh pihak yang tidak berwenang.

CV Karya Belia Nusantara (KBN) sebagai entitas yang mengelola data sensitif menghadapi tantangan nyata berupa potensi pencurian kredensial, pembuatan sesi palsu, dan serangan **replay attack**.

Biner : Jurnal Ilmu Komputer , Teknik dan Multimedia
Volume 3, No. 3 Agustus 2025
ISSN 2828-2442 (media online)
Hal 267-274

Oleh karena itu, diperlukan sistem autentikasi yang tidak hanya mengikuti standar industri, tetapi juga mampu mengantisipasi ancaman-ancaman tersebut. Penelitian ini merancang sistem yang menggabungkan JWT dengan OTP, menggunakan mekanisme **access token** dan **refresh token** yang memiliki masa berlaku jelas, serta dilengkapi fitur **token revocation** secara real-time untuk mengurangi dampak jika token jatuh ke tangan pihak yang tidak sah.

Perancangan sistem ini juga mempertimbangkan aspek **usability**, sehingga proses login tetap mudah tanpa mengorbankan keamanan. Integrasi OTP dirancang agar mendukung **one-click verification** atau **autofill**, sehingga pengguna dapat melakukan autentikasi dengan cepat tanpa harus mengetik ulang kode. Pendekatan ini diharapkan mampu menjaga keseimbangan antara **user experience** dan **security compliance**, yang sering kali menjadi tantangan dalam implementasi keamanan pada aplikasi mobile.

Melalui penelitian ini, diharapkan tercipta sebuah panduan teknis yang dapat diadopsi oleh pengembang lain dalam membangun sistem autentikasi berbasis JWT + OTP pada ekosistem **Android–Spring Boot**. Tidak hanya memberikan kontribusi praktis bagi pengembang dan perusahaan, penelitian ini juga berpotensi menambah khazanah literatur akademis di bidang keamanan aplikasi mobile, khususnya terkait sinkronisasi antara teknologi backend dan frontend dalam penerapan mekanisme autentikasi multi-faktor yang aman dan efisien.

2. METODOLOGI PENELITIAN

2.1 Metode Pengumpulan Data

Penelitian ini menggunakan pendekatan multi-metode untuk memperoleh data yang lengkap dan mendalam. Proses pengumpulan data dilakukan melalui wawancara mendalam dengan pemilik dan tim teknis CV Karya Belia Nusantara guna mengidentifikasi kebutuhan sistem, permasalahan autentikasi sebelumnya, serta ekspektasi keamanan yang diharapkan. Selain itu, dilakukan studi literatur untuk memahami prinsip, standar, dan praktik terbaik terkait penggunaan JSON Web Token (JWT), One-Time Password (OTP), arsitektur Model–View–ViewModel (MVVM), serta teknologi keamanan pada perangkat mobile. Metode berikutnya adalah eksperimen langsung melalui proses pengembangan dan pengujian sistem, dengan fokus pada integrasi antara backend dan aplikasi Android serta pengujian end-to-end untuk memastikan sistem berjalan sesuai harapan.

2.2 Arsitektur Sistem

Sistem yang dikembangkan memiliki tiga komponen utama yang saling terintegrasi, yaitu client berbasis Android Native, backend berbasis Spring Boot dengan Spring Security, serta basis data MySQL. Pada sisi client, aplikasi menyediakan antarmuka pengguna untuk proses login, verifikasi OTP, dan akses ke konten, sekaligus menyimpan token dengan aman menggunakan mekanisme enkripsi perangkat. Token tersebut akan disisipkan secara otomatis pada header HTTP setiap kali melakukan permintaan ke API. Di sisi backend, Spring Boot berperan menyediakan endpoint autentikasi seperti login, OTP, dan refresh token, serta API layanan yang dilindungi. Sistem ini menerapkan filter stateless untuk memvalidasi JWT dan mengekstraksi klaim, serta mengelola sesi dan data refresh token secara terpusat di dalam basis data. Sementara itu, MySQL digunakan untuk menyimpan data pengguna, kode OTP, refresh token, dan informasi sesi, serta berfungsi sebagai pusat revocation control ketika pengguna melakukan logout atau token kadaluarsa. Seluruh komunikasi antar komponen dilakukan melalui protokol RESTful dengan payload berbasis JSON dan header Authorization.

2.3 Alat dan Teknologi

Pengembangan dan evaluasi sistem ini memanfaatkan sejumlah alat dan teknologi yang mendukung proses kerja di sisi client maupun backend. Pada sisi client, Android Studio dengan bahasa pemrograman Kotlin digunakan untuk membangun antarmuka pengguna dan logika

Biner : Jurnal Ilmu Komputer , Teknik dan Multimedia
Volume 3, No. 3 Agustus 2025
ISSN 2828-2442 (media online)
Hal 267-274

aplikasi. Untuk komunikasi antara aplikasi dan server, digunakan Retrofit bersama OkHttp yang dilengkapi interceptor token. Di sisi backend, Spring Boot dan Spring Security menjadi pondasi pengelolaan autentikasi dan keamanan API, sementara JWT Library seperti jjwt digunakan untuk pembuatan dan validasi token. MySQL digunakan sebagai basis data relasional, sedangkan EncryptedSharedPreferences memastikan token yang disimpan di perangkat tetap aman. Dalam tahap pengujian, Postman dimanfaatkan untuk uji coba API, JMeter digunakan untuk mengukur kinerja, dan Appium dipakai untuk mengotomatisasi pengujian UI mobile.

2.4 Tahapan Implementasi

Tahap implementasi dimulai dengan perancangan antarmuka pengguna (UI) dan pengalaman pengguna (UX) untuk proses login dan OTP, yang dibangun berdasarkan prinsip arsitektur MVVM agar pemisahan antara UI, logika, dan data tetap terjaga. Pada tahap ini, juga disusun diagram alur autentikasi yang mencakup skenario kesalahan, mekanisme retry, dan proses refresh token. Selanjutnya, dilakukan pembuatan API backend yang mencakup endpoint seperti /auth/login, /request-otp, /verify-otp, /refresh-token, serta endpoint layanan yang memerlukan JWT. Implementasi mencakup mekanisme OTP berbasis HOTP atau TOTP, pembuatan access dan refresh token, serta penyimpanan refresh token di basis data. Integrasi OTP dan JWT dilakukan dengan cara backend menghasilkan OTP yang dikirimkan melalui SMS atau email, kemudian memvalidasinya sebelum menerbitkan token. Aplikasi Android akan menyimpan token tersebut secara terenkripsi dan menggunakan interceptor untuk menyisipkan Authorization header secara otomatis. Mekanisme refresh token akan berjalan otomatis ketika access token telah kadaluarsa. Setelah itu dilakukan pengujian black box secara end-to-end tanpa akses ke kode sumber, yang mencakup skenario login dengan kredensial dan OTP yang valid maupun tidak valid, pengujian akses API dengan token yang valid, kadaluarsa, atau tanpa token, serta pengujian proses refresh token, logout, dan revocation. Teknik pengujian yang digunakan meliputi boundary value testing, state transition testing, dan decision table untuk memastikan cakupan skenario yang optimal.

2.5 Evaluasi dan Validasi

Evaluasi dan validasi sistem dilakukan dengan beberapa fokus utama. Dari sisi fungsionalitas, pengujian bertujuan memastikan seluruh alur mulai dari login, verifikasi OTP, proses refresh token, hingga akses API berjalan sesuai dengan spesifikasi yang telah ditentukan. Dari sisi keamanan, dilakukan simulasi serangan umum seperti brute-force terhadap OTP, reuse token, serta pencurian token untuk memastikan ketahanan sistem terhadap ancaman tersebut. Dari sisi kinerja, dilakukan pengukuran terhadap latensi pengiriman dan verifikasi OTP, waktu verifikasi JWT, serta analisis dampaknya terhadap pengalaman pengguna. Sementara itu, reliabilitas sistem diuji dengan menjalankan skenario pengujian otomatis secara berulang dan dalam kondisi multi-user untuk mengevaluasi kestabilan sistem pada beban yang berbeda.

Biner : Jurnal Ilmu Komputer , Teknik dan Multimedia
Volume 3, No. 3 Agustus 2025
ISSN 2828-2442 (media online)
Hal 267-274

3. HASIL DAN PEMBAHASAN

3.1 Perancangan Sistem

Pada tahap perancangan sistem, dibuat serangkaian diagram dan dokumentasi yang menjelaskan secara rinci alur, struktur, dan hubungan antar komponen. Diagram aktivitas digunakan untuk menggambarkan proses pengguna mulai dari memasukkan kredensial, mengajukan permintaan OTP, menerima OTP, melakukan verifikasi, hingga mendapatkan akses ke konten aplikasi.

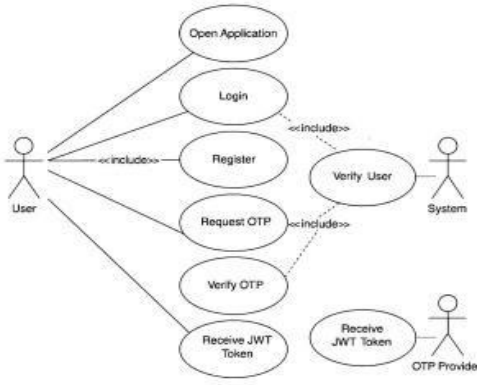
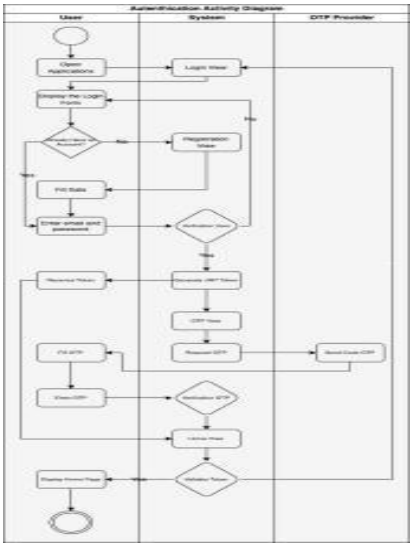
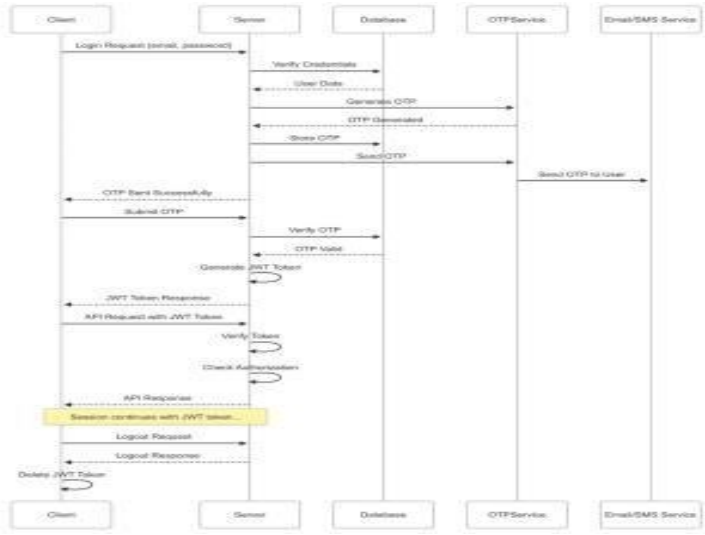


Diagram ini juga memuat skenario alternatif, seperti kesalahan input OTP, OTP yang sudah kedaluwarsa, serta proses permintaan ulang OTP. Selanjutnya, dibuat use case diagram yang menunjukkan aktor-aktor yang terlibat, yaitu pengguna, sistem OTP, dan backend, serta skenario penggunaan seperti registrasi dan permintaan OTP, verifikasi OTP serta penerbitan JWT, proses refresh token, dan logout yang disertai revocation token.

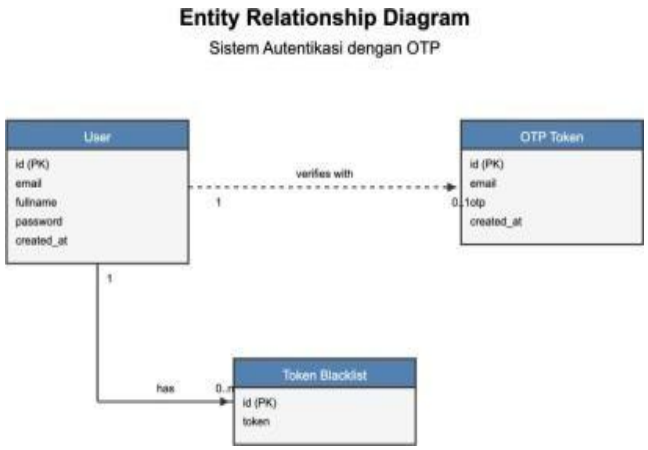


Biner : Jurnal Ilmu Komputer , Teknik dan Multimedia
Volume 3, No. 3 Agustus 2025
ISSN 2828-2442 (media online)
Hal 267-274

Sequence diagram kemudian disusun untuk menjelaskan urutan proses komunikasi antar komponen, dimulai dari pengiriman permintaan login oleh aplikasi Android, pembuatan OTP oleh backend yang disimpan dan dikirimkan melalui SMS atau email, input OTP oleh pengguna, pengiriman OTP ke backend untuk validasi, penerbitan access token dan refresh token jika OTP valid, hingga penyimpanan token terenkripsi di sisi Android dan penggunaannya melalui interceptor. Urutan ini juga mencakup mekanisme refresh token otomatis dan skenario retry OTP.



Selain itu, dibuat Entity Relationship Diagram (ERD) untuk memodelkan struktur data utama, yang mencakup entitas pengguna, OTP, dan refresh token, dengan relasi one-to-many antara pengguna dengan OTP maupun refresh token. Relasi ini memungkinkan pengelolaan OTP per sesi serta revocation token per perangkat atau per pengguna.



3.2 Implementasi Sistem

Biner : Jurnal Ilmu Komputer , Teknik dan Multimedia
Volume 3, No. 3 Agustus 2025
ISSN 2828-2442 (media online)
Hal 267-274

Pada tahap implementasi, backend dibangun menggunakan arsitektur layered yang terdiri dari lapisan controller, service, dan repository, sedangkan aplikasi Android menerapkan pola MVVM untuk memisahkan lapisan tampilan, logika, dan data. Proses login mengadopsi metode payment-based authentication, di mana setelah verifikasi OTP berhasil, backend menghasilkan access token dengan masa berlaku sekitar 15 menit dan refresh token dengan masa berlaku sekitar 7 hari. Kedua token tersebut disimpan secara aman di EncryptedSharedPreferences. Aplikasi Android menggunakan interceptor HTTP untuk menyisipkan access token pada setiap permintaan ke server dan secara otomatis mendeteksi respon 401 jika token kadaluarsa. Apabila hal tersebut terjadi, aplikasi akan memanggil endpoint refresh-token, menerima access token yang baru, dan mengulangi permintaan sebelumnya tanpa interaksi tambahan dari pengguna. Mekanisme revocation token juga diterapkan, di mana saat pengguna logout, refresh token yang tersimpan di backend akan ditandai sebagai revoked di basis data. Dengan begitu, access token yang masih aktif tidak dapat digunakan untuk memperoleh token baru.

3.3 Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode black box dengan beragam skenario penggunaan nyata. Pada tahap registrasi dan permintaan OTP, sistem diuji untuk memastikan OTP dapat diterima dalam waktu singkat dan memberikan pesan kesalahan yang jelas apabila kode yang dimasukkan salah atau sudah kadaluarsa. Pada tahap verifikasi OTP dan penerbitan token, pengujian memastikan bahwa OTP yang valid menghasilkan token dalam format JSON yang lengkap, sementara OTP yang salah atau kadaluarsa menghasilkan respon 401. Pengujian akses endpoint dengan token valid memastikan respon sukses 200 OK dan data yang dihasilkan sesuai hak akses pengguna. Ketika token kadaluarsa digunakan, sistem secara otomatis melakukan refresh token dan mengulangi permintaan awal. Jika token telah di-revoke, sistem memberikan respon 401 dan mengarahkan pengguna kembali ke halaman login. Proses logout juga diuji untuk memastikan bahwa revocation token berjalan sesuai fungsinya dan akses baru hanya dapat dilakukan setelah login ulang.

Hasil pengujian menunjukkan bahwa sistem memiliki keandalan tinggi dengan semua skenario berjalan sesuai alur tanpa error kritis. Dari segi efisiensi, latensi penerimaan OTP dan waktu verifikasi rata-rata kurang dari satu detik pada jaringan normal, sehingga memberikan pengalaman pengguna yang responsif. Dari sisi keamanan, mekanisme refresh token, revocation, dan penyimpanan token secara terenkripsi terbukti mampu mencegah akses ilegal. Selain itu, arsitektur stateless untuk access token yang dipadukan dengan refresh token yang stateful membuat sistem lebih mudah diskalakan untuk kebutuhan yang lebih besar.

3.4 Rangkuman Pembahasan

Berdasarkan hasil pengembangan dan pengujian, dapat disimpulkan bahwa perancangan sistem dengan diagram yang mendetail mempermudah proses implementasi dan validasi. Integrasi OTP, interceptor, serta manajemen siklus hidup token (token lifecycle) berhasil diterapkan dengan baik. Seluruh skenario kritis telah diuji dan berjalan tanpa menimbulkan risiko fungsional. Dari sisi keamanan, penerapan revocation token dan penyimpanan token secara terenkripsi memberikan lapisan perlindungan tambahan terhadap potensi serangan. Dari sisi kinerja, sistem terbukti responsif dan andal untuk digunakan pada skala kecil hingga menengah, sementara penggunaan OTP semakin memperkuat keamanan melalui verifikasi ganda yang berlaku satu kali dan memiliki masa berlaku singkat.

Biner : Jurnal Ilmu Komputer , Teknik dan Multimedia
Volume 3, No. 3 Agustus 2025
ISSN 2828-2442 (media online)
Hal 267-274

3.5 Kelebihan dan Kekurangan Sistem

Sistem autentikasi yang dikembangkan memiliki beberapa kelebihan utama. Keamanan ditingkatkan melalui kombinasi JWT dan OTP, sementara skalabilitas meningkat karena sistem tidak menyimpan sesi secara langsung di server. Dari sisi pengalaman pengguna, proses login menjadi lebih cepat dan sederhana tanpa mengorbankan keamanan. Meski demikian, terdapat beberapa kekurangan yang perlu diperhatikan. Tidak semua pengguna familiar dengan mekanisme OTP, sehingga diperlukan edukasi atau panduan yang jelas. Selain itu, jika token JWT bocor, token tersebut tetap dapat digunakan hingga masa berlakunya habis jika tidak ada mekanisme blacklist yang diterapkan. Sistem OTP juga membutuhkan integrasi dengan layanan pihak ketiga seperti SMS gateway atau Firebase jika ingin mendukung pengiriman kode secara real-time, yang dapat menambah kompleksitas dan biaya operasional.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil penelitian dan implementasi yang telah dilakukan, dapat disimpulkan bahwa penerapan JSON Web Token (JWT) dan One-Time Password (OTP) pada aplikasi Android yang terintegrasi dengan backend Spring Boot di CV Karya Belia Nusantara mampu meningkatkan keamanan autentikasi serta manajemen akses pengguna. Sistem yang dikembangkan berjalan secara stateless untuk access token dan memanfaatkan refresh token secara terkelola, sehingga proses autentikasi menjadi lebih efisien dan tidak membebani server dengan penyimpanan sesi. Mekanisme OTP berfungsi sebagai lapisan keamanan kedua yang efektif dalam mencegah akses tidak sah, bahkan jika kredensial utama terekspos. Selain itu, penyimpanan token secara terenkripsi di sisi client menggunakan EncryptedSharedPreferences berhasil menjaga kerahasiaan data sensitif. Hasil pengujian menunjukkan bahwa sistem memiliki kinerja responsif, keandalan tinggi, serta mampu menangani berbagai skenario autentikasi, termasuk penanganan token kadaluarsa, token yang direvoke, dan validasi OTP dalam batas waktu yang ditentukan. Dengan demikian, sistem ini telah memenuhi tujuan penelitian untuk menciptakan autentikasi yang aman, efisien, dan tetap memberikan pengalaman pengguna yang baik.

4.2 Saran

Untuk pengembangan di masa mendatang, sistem ini dapat diperluas dengan penerapan **Role-Based Access Control (RBAC)** agar pengaturan hak akses pengguna lebih terstruktur dan sesuai kebutuhan organisasi. Mekanisme refresh token yang ada dapat diperkuat dengan sistem blacklist atau token invalidation berbasis real-time untuk menutup celah keamanan jika token dicuri. Integrasi dengan layanan pembayaran digital juga dapat menjadi fitur tambahan yang meningkatkan nilai fungsional aplikasi, khususnya untuk transaksi yang memerlukan autentikasi ganda. Selain itu, disarankan untuk menambahkan **audit log** yang mencatat setiap aktivitas login, percobaan OTP, dan perubahan data penting, guna memudahkan proses pemantauan dan investigasi keamanan. Pembuatan **dashboard admin** yang menampilkan statistik penggunaan, riwayat login, dan status token juga akan memberikan manfaat signifikan dalam pengelolaan dan pengawasan sistem secara menyeluruh. Dengan pengembangan berkelanjutan dan penerapan praktik keamanan terbaru, sistem ini berpotensi menjadi solusi autentikasi yang tidak hanya aman tetapi juga fleksibel untuk berbagai kebutuhan bisnis dan teknologi di masa depan.

DAFTAR PUSTAKA

Jones, M., Bradley, J., & Sakimura, N. (2015). *Json web token (jwt)* (No. rfc7519).

Bucko, A., Vishi, K., Krasniqi, B., & Rexha, B. (2023). Enhancing jwt authentication and authorization in web applications based on user behavior history. Computers, 12(4), 78.

Jones, M., Campbell, B., & Mortimore, C. (2015). *Json web token (jwt) profile for oauth 2.0 client authentication and authorization grants* (No. rfc7523).

Walls, C. (2015). *Spring Boot in action*. Simon and Schuster.

Webb, P., Syer, D., Long, J., Nicoll, S., Winch, R., Wilkinson, A., ... & Deleuze, S. (2013). Spring boot reference guide. *Part IV. Spring Boot features*, *24*.

Suryotrisongko, H., Jayanto, D. P., & Tjahyanto, A. (2017). Design and development of backend application for public complaint systems using microservice spring boot. *Procedia Computer Science*, *124*, 736-743.

Pattinama, Y. L., & Susanti, I. (2023). Implementasi Rest API Web Service Dengan Otentifikasi JSON Web Token Untuk Aplikasi Properti. *Informatik: Jurnal Ilmu Komputer*, *19*(1), 81-89.

Sina, W. I. (2023). *IMPLEMENTASI KEAMANAN AUTHENTICATION RESTFUL WEB SERVICE MENGGUNAKAN JWT PADA APLIKASI FOOD DELIVERY* (Doctoral dissertation, UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA).

Nashikhuddin, A. Y., Karaman, J., & Litanianda, Y. (2023). Implementasi Api Restful Dengan Json Web Token (JWT) Pada Aplikasi E-Commerce Thrifty Shop Untuk Otentikasi Dan Otorisasi Pengguna. *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, *7*(2), 239- 246.

Pudoli, A., Yulianawati, Y., & Suwandi, I. (2023). Implementasi Web Service Restful Dengan Autentikasi JSon Web Token Berbasis Web dan Android. *Acad. J. Comput. Sci. Res*, *5*(2), 95-103.

Dimitrijević, N., Zdravković, N., Bogdanović, M., & Mesterović, A. (2024). Advanced Security

Mechanisms in the Spring Framework: JWT, OAuth, LDAP and Keycloak-based solutions.

Barabanov, A., & Makrushin, D. (2020). Authentication and authorization in microservice-based systems: survey of architecture patterns.

Gonzalez, D., Rath, M., & Mirakhorli, M. (2020). Did You Remember to Test Your Tokens?

Nemade, B. (2024). Mastering JWT Authentication & Authorization in Spring Boot 3.1.

Crudu, V., & MoldStud (2024). Securing Your Spring Boot Application with JWT Authentication Step-by-Step Guide.