IMPLEMENTASI MULTI FACTOR AUTHENTICATION (MFA) PADA SISTEM PRESENSI MENGGUNAKAN SPRING SECURITY

Diva Kurniawan, Yeremia Alfa Susetyo
Teknik Informatika, Universitas Kristen Satya Wacana
Jl. Dr. O. Notohamidjojo No 1-10, Blotongan, Kec. Sidorejo, Kota Salatiga 50715
divakurniawan2003@gmail.com

ABSTRAK

Sistem presensi digital semakin banyak diadopsi di lingkungan kerja, namun rentan terhadap praktik *buddy punching* akibat lemahnya mekanisme autentikasi satu faktor yang masih umum digunakan. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem presensi berbasis *Multi-Factor Authentication* (MFA) yang menggabungkan NFC *Card* sebagai faktor kepemilikan dan PIN sebagai faktor pengetahuan menggunakan Spring Security 6.x, guna mencegah *buddy punching* dan meningkatkan integritas data kehadiran. Sistem dikembangkan menggunakan metode *Waterfall* dengan arsitektur yang terdiri dari aplikasi *desktop* JavaFX 21.0.1, *backend RESTful API* Spring Boot 3.2.0, dan *database* MongoDB 7.0. Pengujian dilakukan melalui enam skenario autentikasi dan tujuh aspek keamanan untuk mengevaluasi efektivitas sistem. Hasil pengujian menunjukkan seluruh skenario berhasil lulus, di mana sistem berhasil menolak akses pada kasus kartu dipinjam tanpa PIN maupun PIN diketahui tanpa kartu fisik, dengan seluruh *endpoint* kritis merespons di bawah 500 milidetik. Penelitian ini membuktikan bahwa kombinasi NFC *Card* dan PIN dengan Spring Security merupakan solusi yang efektif, praktis, dan aman untuk sistem presensi yang tahan terhadap manipulasi data kehadiran.

Kata kunci : *Multi-Factor Authentication, NFC Card, PIN, Spring Security, Buddy Punching, Sistem Presensi*

1. PENDAHULUAN

Perkembangan teknologi informasi memberikan dampak signifikan di berbagai sektor, termasuk industri dan pendidikan, sehingga perusahaan kini dapat membangun sistem berbasis komputer yang lebih efisien dan terpadu [1].

Adopsi aplikasi absensi modern berbasis metode *Agile* mempercepat siklus pengembangan sesuai kebutuhan pengguna [2], namun sistem presensi digital tetap rentan terhadap serangan *malware*, *phishing*, dan eksploitasi kode palsu yang berpotensi menyebabkan manipulasi data [3], serta meningkatnya pelanggaran data pribadi mendorong urgensi penerapan enkripsi dan audit keamanan berkelanjutan [4].

Oleh karena itu, diperlukan sistem presensi yang tidak hanya efisien tetapi juga aman untuk menjaga integritas data dan kepercayaan *stakeholder*.

Salah satu bentuk penyalahgunaan yang sering terjadi adalah praktik *buddy punching*, yaitu tindakan merekam kehadiran atas rekan yang tidak hadir. Fenomena ini mencerminkan lemahnya autentikasi satu faktor yang masih banyak digunakan, yang tidak hanya menyebabkan kerugian finansial berupa upah fiktif tetapi juga menurunkan produktivitas [5].

Studi komprehensif menegaskan bahwa sistem satu faktor memiliki tingkat keamanan lebih rendah dan memungkinkan manipulasi akses secara signifikan [6].

Permasalahan utama yang diangkat dalam penelitian ini adalah bagaimana merancang sistem presensi yang mampu mencegah praktik *buddy punching* secara efektif melalui mekanisme autentikasi yang tidak dapat dimanipulasi oleh pihak yang tidak berwenang. Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem presensi berbasis *Multi-Factor Authentication* (MFA) yang menggabungkan NFC *Card* dan PIN menggunakan Spring Security 6.x, guna mencegah praktik *buddy punching* dan meningkatkan integritas data kehadiran karyawan.

Sejauh tinjauan pustaka yang dilakukan, belum ditemukan penelitian yang secara spesifik mengimplementasikan MFA berbasis NFC *Card* dan PIN menggunakan arsitektur Spring Security 6.x dalam konteks sistem presensi karyawan untuk mencegah *buddy punching*. Penelitian terdahulu umumnya mengeksplorasi NFC sebagai autentikasi faktor tunggal [7], atau *face recognition* untuk presensi [5][8], namun belum mengintegrasikan keduanya dalam satu arsitektur yang dievaluasi dari sisi efektivitas pencegahan *buddy punching* maupun performa sistem secara bersamaan.

MFA menawarkan keamanan berlapis dengan menggabungkan minimal dua metode verifikasi berbeda, seperti sesuatu yang dimiliki (NFC *Card*) dan sesuatu yang diketahui (PIN), dan terbukti memberikan peningkatan signifikan dalam perlindungan privasi, penurunan kebocoran data, serta pengurangan akses ilegal [9]. Spring Security sebagai *framework* yang fleksibel dan modular memungkinkan implementasi MFA melalui *authentication provider chain*, *custom filter*, dan manajemen sesi yang aman, serta pemisahan logika autentikasi setiap faktor secara terstruktur dan tahan eksploitasi [10][11].

Kombinasi NFC *Card* sebagai faktor kepemilikan fisik dan PIN sebagai faktor pengetahuan menutup celah utama *buddy punching* karena pencatatan kehadiran hanya dapat dilakukan oleh pemilik kartu yang sah sekaligus mengetahui PIN-nya,

sehingga integritas data kehadiran dan akuntabilitas karyawan dapat terjamin secara bersamaan.

2. TINJAUAN PUSTAKA

Pada penelitian yang berjudul ”*Analisis Penggunaan Sistem Face Recognition dalam Mengelola Absensi Karyawan di PT Bintang Inspeksi Indonesia*”, disebutkan bahwa sistem pencatatan kehadiran konvensional memiliki beberapa kelemahan signifikan, baik dalam hal kerentanan terhadap penyalahgunaan absen maupun ketidakcepatan penyusunan laporan kehadiran yang mengurangi produktivitas administrasi [5].

Senada dengan itu, penelitian berjudul "*Online Attendance System Based on Facial Recognition with Face Mask Detection*" mengembangkan sistem presensi berbasis *face recognition* secara *online* yang mampu mendeteksi wajah secara *real-time* melalui antarmuka *browser* tanpa instalasi perangkat lunak khusus [8].

Meskipun sistem ini terbukti efektif dalam otomatisasi pencatatan kehadiran, ketergantungannya pada kondisi visual pengguna seperti penggunaan masker wajah dan kondisi pencahayaan menunjukkan bahwa pendekatan berbasis biometrik masih memiliki keterbatasan adaptabilitas di lingkungan kerja yang beragam [8].

Oleh karena itu, pendekatan alternatif berbasis multi-faktor seperti kombinasi NFC dan PIN menjadi solusi yang lebih stabil dan tidak bergantung pada kondisi lingkungan fisik.

Pada penelitian yang berjudul "An *Improved NFC Device Authentication Protocol*", dijelaskan bahwa pengembangan sistem autentikasi perangkat berbasis *Near Field Communication* (NFC) dengan pendekatan multi-faktor terbukti dapat meningkatkan keamanan komunikasi perangkat terhadap ancaman siber [12].

Penelitian tersebut memperkenalkan protokol otentikasi dua tahap berbasis NFC yang menggabungkan algoritma kriptografi simetris dan asimetris, *hash function*, dan penggunaan nilai acak (*nonce*) dalam dua fase utama: registrasi dan autentikasi. Dalam fase autentikasi, sistem menggunakan pertukaran *nonce* serta verifikasi tanda waktu (*timestamp*) untuk mencegah *replay attack* dan manipulasi data. Hasil eksperimen menunjukkan bahwa protokol ini berhasil melindungi data dari berbagai ancaman seperti *brute-force attack*, *eavesdropping*, dan *replay attack*, tanpa mengorbankan efisiensi waktu proses [12].

Keunggulan teknis ini sejalan dengan penelitian yang sedang dikembangkan, yaitu sistem presensi berbasis MFA yang menggabungkan teknologi NFC *Card* sebagai faktor pertama dan PIN sebagai faktor kedua, dengan dukungan fitur keamanan dari Spring Security. Dengan merujuk pada keberhasilan integrasi NFC dan kriptografi dalam sistem otentikasi dua faktor, penelitian ini mengadaptasi pendekatan serupa dengan menyesuaikan arsitektur protokol menggunakan kombinasi NFC *Card* dan PIN sebagai solusi keamanan presensi yang andal dan efisien.

Pada penelitian yang berjudul "*Lecturer Attendance System Based on Near Field Communication Technology*", dijelaskan bahwa penggunaan teknologi *Near Field Communication* (NFC) terbukti dapat meningkatkan efisiensi dan otomatisasi dalam sistem presensi secara signifikan [7].

Penelitian ini mengembangkan sistem presensi dosen berbasis NFC MIFARE dan NFC *reader/writer*, yang secara otomatis mencatat kehadiran dalam waktu singkat dengan jarak baca hingga 7 cm dan sudut baca hingga 300° [7].

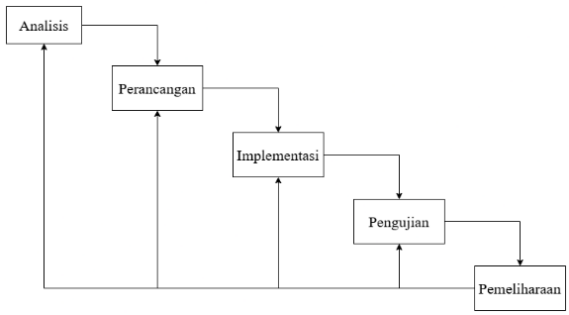
Arsitektur sistem mencakup modul verifikasi ID unik kartu, pencatatan waktu otomatis, serta integrasi dengan modul manajemen data dan pelaporan absensi. Hasil pengujian menunjukkan bahwa penggunaan NFC jauh lebih praktis dibanding metode manual, memangkas waktu presensi dan mengurangi kesalahan input data serta kecurangan seperti penyalahgunaan kartu (*buddy punching*). Dari keberhasilan implementasi ini, penelitian ini mengadaptasi pendekatan NFC otomatis sebagai faktor pertama, yang selanjutnya dilengkapi dengan lapis keamanan berupa PIN dan integrasi protokol Spring Security untuk meningkatkan keandalan autentikasi dan mencegah manipulasi data secara lebih optimal.

Berdasarkan ketiga penelitian terdahulu, dapat disimpulkan bahwa teknologi autentikasi seperti *face recognition* dan *Near Field Communication* (NFC) telah memberikan kontribusi signifikan dalam meningkatkan efisiensi dan keamanan sistem presensi. Sistem *face recognition* terbukti efektif dalam mencegah kecurangan, namun memiliki keterbatasan dalam kondisi pencahayaan yang buruk serta keterhubungan jaringan. Sementara itu, pendekatan berbasis NFC menunjukkan keunggulan dalam hal kecepatan, kepraktisan, dan integrasi dengan teknologi *backend* serta kriptografi modern. Meski demikian, NFC yang diterapkan sebagai autentikasi satu faktor belum sepenuhnya menjawab kebutuhan terhadap sistem presensi yang aman, andal, dan tahan manipulasi karena kartu tetap dapat dipinjamkan kepada pihak lain tanpa hambatan. Oleh karena itu, pendekatan autentikasi berlapis seperti *Multi-Factor Authentication* (MFA) yang menggabungkan NFC *Card* dengan PIN menjadi solusi yang lebih relevan dan adaptif, karena menutup celah utama *buddy punching* yang tidak dapat diselesaikan oleh autentikasi satu faktor.

3. METODE PENELITIAN

Pada penelitian ini, digunakan pendekatan rekayasa perangkat lunak dengan metode *Waterfall*. *Waterfall* merupakan metode pengembangan perangkat lunak yang bersifat sekuensial dan terstruktur, di mana setiap tahapan pengembangan diselesaikan secara berurutan sebelum melanjutkan ke tahapan berikutnya, mulai dari analisis kebutuhan,

perancangan, implementasi, hingga pengujian sistem sebagaimana ditunjukkan pada Gambar 1.



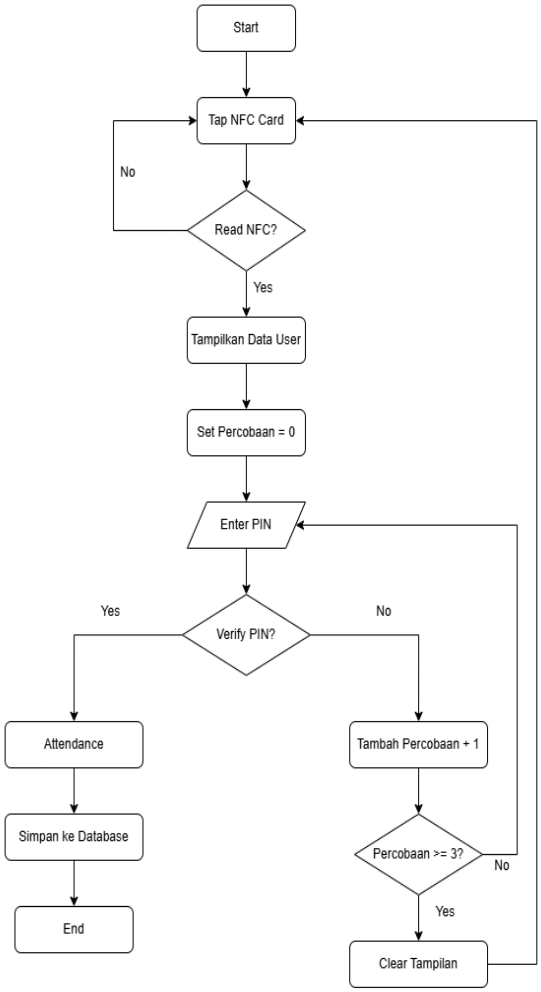
Gambar 1. Tahapan penelitian

Metode *Waterfall* terbukti efektif diterapkan secara sistematis dalam pengembangan sistem informasi manajemen untuk aplikasi bisnis skala perusahaan, khususnya pada proyek dengan kebutuhan yang sudah terdefinisi sejak awal [13]. Dengan karakteristik sistem presensi MFA yang dikembangkan dalam penelitian ini memiliki kebutuhan fungsional dan non-fungsional yang sudah teridentifikasi secara jelas sejak tahap awal, metode *Waterfall* dinilai sesuai untuk memandu proses implementasi kombinasi NFC *Card* dan PIN menggunakan Spring Security secara terstruktur dan sistematis.

a. Tahap Analisis
   Pada tahap ini dilakukan analisis kebutuhan terhadap sistem presensi yang akan dibangun. Peneliti mengidentifikasi masalah utama yang terjadi pada sistem presensi satu faktor (seperti *buddy punching*) dan menentukan kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional meliputi kemampuan sistem untuk membaca kartu NFC, menampilkan data pengguna, memverifikasi PIN, mencatat waktu kehadiran, serta menyimpan data presensi ke *database*. Sedangkan kebutuhan non-fungsional mencakup keamanan, kecepatan proses autentikasi, dan kemudahan penggunaan. Hasil dari tahap ini berupa dokumen spesifikasi kebutuhan sistem.
b. Tahap Perancangan
   Tahap ini fokus pada desain sistem berdasarkan hasil analisis sebelumnya. Perancangan dilakukan dalam bentuk *flowchart* proses presensi, desain arsitektur sistem (komunikasi antara NFC *reader*, *desktop* GUI, dan *backend* Spring Boot), serta perancangan struktur database menggunakan MongoDB. Antarmuka *desktop* dirancang untuk menerima input PIN setelah NFC berhasil dibaca. Selain itu, pada tahap ini dilakukan perancangan logika *Multi-Factor Authentication* dengan menggunakan *filter* dan konfigurasi khusus dari Spring Security untuk memisahkan autentikasi berdasarkan faktor (NFC sebagai faktor pertama dan PIN sebagai faktor kedua).
c. Tahap Implementasi
   Pada tahap ini, sistem mulai dikembangkan berdasarkan desain yang telah dibuat. Aplikasi *desktop* dibuat menggunakan Java (JavaFX/Swing) untuk mendeteksi dan membaca kartu NFC menggunakan *library* yang sesuai (seperti *Java Smart Card API* atau libNFC *wrapper*). Setelah UID terbaca, aplikasi akan menampilkan data pengguna dan meminta input PIN. *Backend* dikembangkan dengan Spring Boot, menggunakan Spring Security untuk mengelola proses autentikasi berlapis. Koneksi ke *database* MongoDB dilakukan menggunakan Spring Data *MongoDB*. Setiap presensi yang berhasil akan disimpan ke dalam *collection* presensi dengan *timestamp* dan ID pengguna yang relevan.
d. Tahap Pengujian
   Pengujian dilakukan dengan pendekatan *black-box* testing untuk memastikan bahwa setiap fitur berjalan sesuai harapan. Pengujian dilakukan terhadap pembacaan UID NFC, validasi input PIN, pembatasan jumlah percobaan PIN, pencatatan kehadiran ke *database*, serta keamanan proses autentikasi. Skenario pengujian mencakup input valid dan tidak valid, percobaan PIN salah lebih dari tiga kali, hingga pengujian koneksi antara perangkat NFC dan aplikasi *desktop*. Tujuannya adalah untuk memastikan sistem berjalan stabil dan akurat.
e. Tahap Pemeliharaan
   Tahap ini dilakukan setelah sistem diuji dan dinyatakan layak pakai. Pemeliharaan yang dilakukan bersifat ringan, seperti perbaikan *minor* terhadap tampilan antarmuka jika ditemukan ketidaksesuaian, pengaturan ulang konfigurasi autentikasi jika dibutuhkan, atau dokumentasi ulang pengguna sistem. Sistem juga didesain agar dapat dikembangkan lebih lanjut, misalnya dengan menambahkan autentikasi biometrik atau integrasi ke sistem kehadiran berbasis *cloud*.

Untuk menggambarkan lebih lanjut bagaimana proses presensi dijalankan pada sistem yang dikembangkan, berikut disajikan *flowchart* sistem *Multi-Factor Authentication* (MFA) menggunakan kombinasi NFC dan PIN.

Sebagaimana ditunjukkan pada Gambar 2, prosedur presensi diawali dengan penempelan kartu NFC ke *reader* untuk membaca UID sebagai faktor pertama. Jika UID valid, sistem menampilkan data karyawan dan meminta input PIN 6 digit sebagai faktor kedua. PIN di-*hash* dan dibandingkan dengan *hash* di *database*, dengan maksimal 3 kali percobaan sebelum sistem mengunci akses. Jika PIN benar, presensi dicatat ke *database* dan sistem kembali ke mode siaga untuk menerima kartu berikutnya.



Gambar 2. Flowchart proses presensi MFA

4. HASIL DAN PEMBAHASAN

Penelitian ini berhasil mengimplementasikan sistem presensi berbasis MFA menggunakan kombinasi NFC *Card* dan PIN melalui *framework* Spring Security untuk mencegah *buddy punching*. Sistem terdiri dari tiga komponen terintegrasi sebagaimana ditunjukkan pada Tabel 1, yaitu aplikasi *desktop* JavaFX 21.0.1 yang terhubung dengan NFC *reader* ACR122U, *backend* RESTful API berbasis Spring Boot 3.2.0 dengan Spring Security, dan *database* MongoDB 7.0 yang di-deploy pada *cloud* MongoDB Atlas.

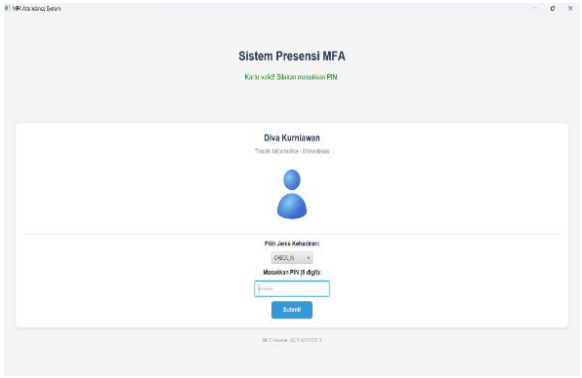
Tabel 1. Arsitektur Sistem

Komponen	Teknologi
Backend API	Spring Boot 3.2.0 + Spring Security
Aplikasi Desktop	JavaFX 21.0.1
Database	MongoDB 7.0 (Atlas)
Perangkat Keras	ACR122U NFC Reader + Kartu NFC

Dari Gambar 3 dapat dilihat kondisi awal antarmuka saat sistem siap menerima autentikasi, menampilkan instruksi tap kartu NFC dan status koneksi NFC reader ACR122U yang telah terdeteksi.

Gambar 3. Tampilan awal aplikasi sistem presensi MFA dalam mode siaga

Validasi dilakukan di backend melalui dua endpoint terpisah: /api/auth/nfc untuk verifikasi UID dan /api/auth/pin untuk verifikasi PIN, dengan data MongoDB mencakup UID kartu, PIN yang di-hash menggunakan BCrypt, serta riwayat presensi.

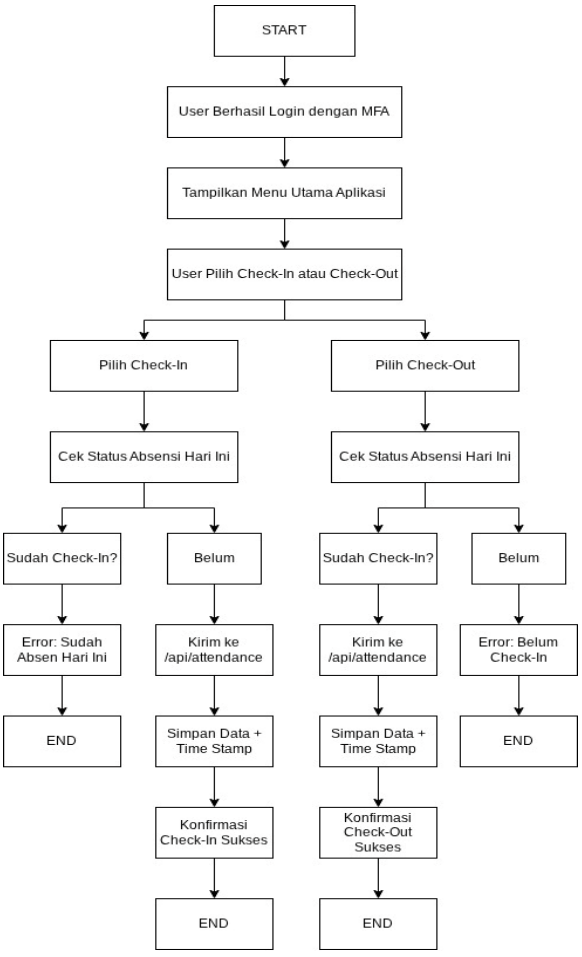


Gambar 4. Tampilan antarmuka setelah validasi NFC berhasil dan input PIN faktor kedua

Mekanisme MFA yang diimplementasikan menempatkan kartu NFC sebagai faktor kepemilikan dan PIN 6 digit sebagai faktor pengetahuan yang divalidasi secara berurutan. Proses dimulai ketika pengguna menempelkan kartu ke reader, sistem membaca UID dan mengirimkannya ke endpoint */api/auth/nfc* untuk divalidasi pada database MongoDB. Jika UID valid, antarmuka menampilkan notifikasi "Kartu valid! Silakan masukkan PIN." serta informasi karyawan meliputi nama, departemen, dan foto sebagai konfirmasi visual, sebagaimana ditunjukkan pada Gambar 4, sekaligus menyediakan dropdown jenis kehadiran dan input PIN sebagai autentikasi tahap kedua. PIN yang dimasukkan kemudian dikirim ke endpoint */api/auth/pin*, di-hash menggunakan BCrypt dengan *cost factor* 10, dan dibandingkan dengan data di database sebelum presensi dicatat. Apabila PIN valid, presensi dicatat secara otomatis dan konfirmasi ditampilkan sebagaimana terlihat pada Gambar 5.

Gambar 5. Tampilan konfirmasi keberhasilan proses check-in setelah autentikasi MFA selesai

Dari Gambar 5 dapat dilihat notifikasi "✓ Check-in berhasil!" sebagai konfirmasi bahwa autentikasi MFA telah terpenuhi dan data kehadiran berhasil dicatat, setelah itu antarmuka otomatis kembali ke mode siaga. Manajemen sesi menggunakan JWT yang bersifat stateless memastikan hanya token valid yang dapat mengakses *endpoint* [14], sehingga kedua faktor wajib terpenuhi bersamaan sebelum presensi dapat dicatat.



Gambar 6. Flowchart Alur Autentikasi MFA

Spring Security berperan sentral dalam implementasi MFA dengan menyediakan infrastruktur keamanan yang memungkinkan pemisahan logika autentikasi melalui *custom authentication filter* dan *provider*, sebagaimana digambarkan pada Gambar 6. *Framework* ini mengelola validasi NFC di tahap pertama menggunakan custom filter yang mengekstrak UID dari *request* dan memverifikasi keberadaannya di MongoDB, namun belum menggenerate JWT token karena autentikasi belum lengkap. Validasi PIN di tahap kedua hanya diproses jika tahap pertama berhasil, menggunakan BCrypt *password encoder* untuk hashing dan membandingkannya dengan database sebelum menghasilkan JWT token. Spring Security juga mengimplementasikan mekanisme *account locking* yang menghitung percob aan PIN gagal per UID dan mengunci akses selama 5 menit setelah 3 kali percobaan gagal, serta menerapkan *role-based access control* dengan dua role yaitu USER untuk akses terbatas dan ADMIN untuk akses penuh.

Setelah implementasi sistem selesai, dilakukan pengujian untuk mengevaluasi efektivitas mekanisme autentikasi yang telah diterapkan. Pengujian autentikasi MFA dilakukan untuk memvalidasi efektivitas sistem dalam mencegah *buddy punching* melalui enam skenario yang mencakup autentikasi normal, kartu tidak terdaftar, PIN salah, dan simulasi serangan.

Tabel 2. Hasil Pengujian Autentikasi MFA

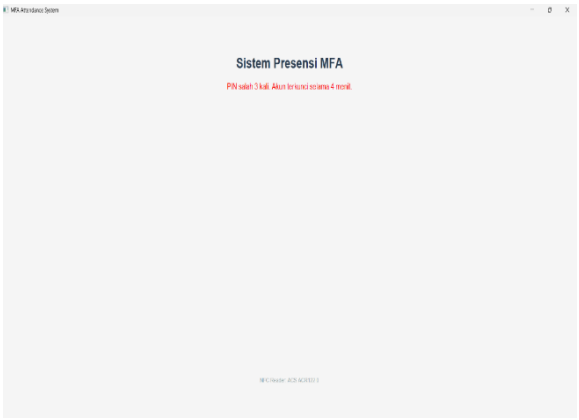
Skenario Pengujian	**Faktor 1 (NFC)**	**Faktor 2 (PIN)**	**Hasil**	**Status**
Autentikasi Normal	UID Valid	PIN Benar	Autentikasi Berhasil	Lulus
Kartu Tidak Terdaftar	UID Tidak Valid	-	Ditolak di Tahap 1	Lulus
Pin Salah (1x)	UID Valid	Pin Salah	Error, Minta Input Ulang	Lulus
Pin Salah (3x)	UID Valid	Pin Salah 3x	Akun Terkunci 5 Menit	Lulus
Kartu Dipinjam Tanpa PIN	UID Valid	Pin Salah	Ditolak di Tahap 2	Lulus
Tahu PIN Tanpa Kartu	UID Tidak Valid	Pin Benar	Ditolak di Tahap 1	Lulus

Berdasarkan hasil pengujian autentikasi MFA pada Tabel 2, seluruh skenario berhasil lulus dengan hasil sesuai harapan. Pencapaian ini dapat diperbandingkan langsung dengan hasil penelitian yang mengembangkan protokol autentikasi NFC dua fase berbasis pertukaran *nonce*, kriptografi simetris-asimetris, dan verifikasi *timestamp* [12].

Protokol tersebut terbukti berhasil mencegah *brute force attack*, *replay attack*, dan *man-in-the-middle attack* dalam lingkungan komunikasi antar perangkat NFC [12].

Namun, protokol tersebut dirancang khusus untuk autentikasi perangkat-ke-perangkat (*device-to-devicedar*), bukan untuk skenario autentikasi pengguna manusia pada sistem presensi. Penelitian ini mengadaptasi prinsip serupa yakni validasi berlapis

dengan penggunaan nilai acak dan *hash* namun mengimplementasikannya dalam konteks yang lebih aplikatif: autentikasi karyawan menggunakan kombinasi NFC *Card* (faktor kepemilikan) dan PIN (faktor pengetahuan) melalui *framework* Spring Security.



Gambar 7. Tampilan antarmuka ketika akun terkunci setelah tiga kali percobaan PIN gagal

Dari sisi mekanisme perlindungan terhadap serangan *brute force* yang mengandalkan kompleksitas kriptografis melalui kombinasi enkripsi simetris, asimetris, dan fungsi *hash* sebagai penghambat utama [12].

Berbeda dengan pendekatan tersebut, penelitian ini menerapkan metode yang lebih pragmatis melalui mekanisme *account locking* berbasis jumlah percobaan gagal, di mana akun dikunci selama 5 menit setelah tiga kali kesalahan PIN, sebagaimana ditunjukkan pada Gambar 7. Pendekatan ini dinilai lebih sesuai untuk sistem presensi karyawan yang menuntut kemudahan operasional tanpa mengorbankan keamanan. Selain itu, waktu komputasi BCrypt rata-rata 342 milidetik turut memberikan resistensi tambahan terhadap serangan *brute force* otomatis dengan memperlambat laju percobaan PIN sejalan dengan tujuan yang ingin dicapai melalui kompleksitas kriptografis [12].

Sistem juga menampilkan pesan peringatan berwarna merah, “PIN salah 3 kali. Akun terkunci selama 4 menit.”, yang menunjukkan penonaktifan akses sementara beserta sisa waktu penguncian secara dinamis dari total 5 menit. Selama periode ini, antarmuka kembali ke mode minimalis tanpa menampilkan data karyawan untuk mencegah paparan informasi sensitif. Mekanisme ini sesuai dengan skenario pengujian keempat pada Tabel 2 dan terbukti efektif menghambat serangan *brute force*, karena percobaan PIN tidak dapat dilanjutkan meskipun kartu NFC tetap tersedia.

Tabel 3. Hasil Pengujian Keamanan

Aspek Keamanan	**Metode Validasi**	**Hasil**
Hashing BCrypt	Inspeksi database	Tidak ada plaintext tersimpan
Autentikasi JWT	Percobaan akses tanpa otorisasi	403 Forbidden
Mekanisme Penguncian	3 kali percobaan PIN gagal	Penguncian 5 menit diterapkan
Proteksi Brute Force	Percobaan PIN otomatis	Terlindungi
Validasi Input	Injeksi karakter khusu	Input berbahaya ditolak
Session Management	Penggunaan token expired	Akses ditolak, perlu login ulang
Role-based Access	User biasa akses endpoint admin	403 Forbidden dikembalikan

Berdasarkan hasil pengujian pada Tabel 3, implementasi fitur keamanan Spring Security terbukti efektif dalam memproteksi sistem melalui tujuh parameter pengujian yang komprehensif. Inspeksi *database* membuktikan tidak ada PIN tersimpan dalam *plaintext* karena semua sudah di-*hash* dengan BCrypt dengan *prefix* 2a$10 yang mengindikasikan *cost factor* 10. BCrypt dipilih sebagai algoritma hashing PIN karena kemampuannya dalam mempersulit serangan *brute force* melalui mekanisme *adaptive cost factor* dan penggunaan *salt* yang mencegah *rainbow table attack* [15].

Pengujian autentikasi JWT menunjukkan *server* mengembalikan *response* 403 *Forbidden* ketika mengakses *endpoint* tanpa token, membuktikan *filter* Spring Security memvalidasi token dengan benar.

Jika dibandingkan dengan sistem presensi dosen berbasis NFC MIFARE sebagai faktor autentikasi tunggal, terdapat perbedaan mendasar dalam aspek keamanan [7].

Sistem tersebut dilaporkan mampu membaca kartu pada jarak hingga 7 cm dengan sudut baca 0°–300°, dan berhasil memangkas waktu presensi secara signifikan dibanding metode manual [7].

Namun, penelitian tersebut tidak membahas mekanisme pengamanan terhadap skenario peminjaman kartu (*card lending*) yang merupakan inti dari praktik *buddy punching*. Dengan kata lain, sistem NFC tunggal yang dikembangkan secara inheren rentan terhadap penyalahgunaan kartu oleh pihak tidak berwenang karena tidak ada lapisan verifikasi identitas di luar kepemilikan fisik kartu [7].

Sistem yang dikembangkan dalam penelitian ini mengatasi celah tersebut secara langsung: meskipun kartu NFC berhasil dibaca, proses autentikasi tetap akan gagal apabila PIN yang dimasukkan tidak valid, sebagaimana divalidasi dalam skenario pengujian kelima pada Tabel 2 (*"Kartu Dipinjam Tanpa PIN"* Ditolak di Tahap 2).

Dari sisi keamanan penyimpanan data, sistem NFC tunggal pada penelitian terdahulu tidak menguraikan mekanisme proteksi data sensitif di *database* [7].

Sebaliknya, penelitian ini mengimplementasikan *hashing* BCrypt dengan *cost factor* 10 untuk seluruh PIN, sehingga tidak ada *plaintext* PIN yang dapat dieksfiltrasi meskipun terjadi kebocoran *database*, sebagaimana dikonfirmasi pada Tabel 3. Perbedaan ini menunjukkan bahwa penambahan faktor PIN berbasis Spring Security tidak hanya meningkatkan pencegahan *buddy punching*, tetapi juga memperkuat postur keamanan sistem secara keseluruhan dibandingkan sistem NFC tunggal.

Tabel 4. Waktu respons API (n=50 permintaan)

Endpoint	Rata-rata (ms)	Std Dev (ms)
POST /api/auth/nfc	125	23,5
POST /api/auth/pin	342	38,2
POST /api/attendance	87	18,7

Dari sisi performa, seluruh *endpoint* kritis berhasil merespons di bawah ambang batas 500 milidetik (Tabel 4). *Endpoint* /api/auth/nfc mencatat rata-rata 125ms karena hanya melakukan *query* sederhana ke *database*, sementara /api/attendance hanya membutuhkan 87ms untuk operasi *insert*. *Endpoint* /api/auth/pin menjadi yang paling lambat dengan rata-rata 342ms, akibat proses *hashing* BCrypt yang *computationally expensive* namun hal ini justru merupakan fitur keamanan yang disengaja, bukan kelemahan. Perlu dicatat bahwa sistem NFC berbasis *single factor* pada penelitian terdahulu tidak menyertakan data waktu respons *backend* secara eksplisit; mereka hanya melaporkan parameter fisik pembacaan kartu NFC (jarak dan sudut baca) [7]. Dengan demikian, sistem dalam penelitian ini memberikan transparansi performa yang lebih terukur dan dapat dibandingkan secara kuantitatif.

Dibandingkan dengan protokol autentikasi NFC yang mengutamakan pengurangan jumlah transmisi pesan guna meningkatkan efisiensi [12], penelitian ini mengambil pendekatan yang berbeda: efisiensi diraih melalui pemisahan logika validasi per-*endpoint* pada Spring Security, sehingga setiap tahap autentikasi dapat diproses secara independen dan transparan. Standar deviasi yang rendah pada ketiga *endpoint* (23,5ms, 38,2ms, dan 18,7ms) mengindikasikan konsistensi waktu respons tanpa *bottleneck* signifikan, yang mengkonfirmasi stabilitas arsitektur sistem dalam kondisi operasional normal.

Tabel 5. Perbandingan dengan Penelitian Sebelumnya

Aspek	Penelitian Ini (MFA NFC+PIN)	Dewi & Fatimah 2022 (Face Recognition)	Putra & Labasariyani 2021 (NFC Tunggal)
Metode Autentikasi	2 Faktor	1 Faktor (Biometrik)	1 Faktor (Kartu)
Pencegahan Buddy Punching	Tinggi	Sedang	Rendah
Ketergantungan Lingkungan	Rendah	Tinggi	Rendah
Keamanan Brute Force	Ada (Locking 5 menit)	Tidak Ada	Tidak ada
Proteksi Data (Hashing)	BCrypt cost factor 10	Tidak dibahas	Tidak dibahas
Waktu Respons Backend	87–342ms (terukur)	2-5 detik (per pengguna)	Tidak dilaporkan

Berdasarkan hasil implementasi dan pengujian secara menyeluruh, sistem MFA yang dikembangkan dalam penelitian ini menunjukkan keunggulan komparatif yang jelas terhadap pendekatan-pendekatan yang digunakan dalam penelitian terdahulu sebagaimana yang di rangkum pada Tabel 5. Dibandingkan dengan penelitian terdahulu yang menganalisis penerapan sistem *face recognition* di PT Bintang Inspeksi Indonesia menggunakan metode deskriptif kualitatif terdapat perbedaan fundamental dalam aspek ketahanan terhadap variabel lingkungan [5].

Sistem tersebut dilaporkan bahwa sistem *face recognition* berhasil meningkatkan efisiensi pencatatan absensi dan mempercepat proses rekapitulasi dengan waktu pengenalan hanya 2–5 detik per individu. Namun, penelitian tersebut juga mengakui adanya ketergantungan pada kualitas jaringan dan perangkat keras sebagai tantangan implementasi [5].

Lebih dari itu, secara teknis sistem *face recognition* memiliki keterbatasan inheren terhadap kondisi pencahayaan, penggunaan masker wajah, dan variasi sudut wajah faktor-faktor yang dapat menurunkan akurasi pengenalan dan membuka potensi *false rejection* (penolakan pengguna sah) maupun *false acceptance* (penerimaan pihak tidak sah) [5].

Sistem MFA berbasis NFC Card dan PIN yang dikembangkan dalam penelitian ini tidak menghadapi keterbatasan tersebut. Autentikasi NFC bersifat deterministik kartu dibaca atau tidak dibaca tanpa ambiguitas akurasi pengenalan yang bergantung pada kondisi fisik pengguna. Sementara itu, validasi PIN melalui BCrypt memastikan bahwa bahkan jika kartu berhasil dipinjamkan, proses autentikasi tetap akan gagal tanpa pengetahuan PIN yang benar. Dengan demikian, sistem ini mencapai tingkat pencegahan *buddy punching* yang lebih tinggi dibanding *face recognition*, sekaligus mempertahankan ketergantungan lingkungan yang rendah sebuah keseimbangan yang tidak berhasil dicapai secara optimal oleh ketiga penelitian terdahulu secara individual.

Selain dari sisi efektivitas autentikasi, sistem yang dikembangkan juga dirancang dengan mempertimbangkan aspek keamanan secara

menyeluruh melalui pendekatan berlapis. Arsitektur keamanan sistem mengikuti strategi *defense-in-depth* dengan empat lapisan independen yang saling melindungi. Lapisan aplikasi melakukan validasi input untuk mencegah *injection attack* termasuk NoSQL *injection* meskipun menggunakan MongoDB, lapisan autentikasi mengimplementasikan MFA untuk verifikasi identitas kuat, lapisan otorisasi menerapkan JWT dan *role-based access control* untuk kontrol akses *resource*, dan lapisan data menggunakan BCrypt *hashing* untuk melindungi PIN saat tersimpan. Implementasi ini selaras dengan panduan OWASP Top 10 2021 untuk mitigasi *Broken Access Control* dan *Cryptographic Failures*, serta rekomendasi pembatasan percobaan *login* pada rentang 5–30 menit untuk menghambat serangan otomatis tanpa mengganggu pengguna sah [16].

Keterbatasan penelitian mencakup pengujian dalam lingkungan terkontrol dengan maksimal 10 pengguna konkuren sehingga belum divalidasi untuk *deployment* produksi skala *enterprise* dengan ratusan pengguna bersamaan. Sistem baru diuji dalam periode singkat sehingga keandalan jangka panjang seperti potensi *memory leak* atau degradasi performa *database* dengan akumulasi data besar belum diketahui. Belum dilakukan audit keamanan eksternal oleh pihak ketiga independen atau *penetration testing* komprehensif yang mungkin mengungkap celah keamanan tidak terdeteksi dalam pengujian internal. Arah pengembangan selanjutnya meliputi aplikasi *mobile* untuk mendukung kerja jarak jauh dengan NFC *smartphone* atau *geolocation*, implementasi mode *offline* dengan sinkronisasi lokal, pengujian skalabilitas ratusan hingga ribuan pengguna, dan integrasi *blockchain* untuk jejak audit *immutable*.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan *Multi-Factor Authentication* (MFA) pada sistem presensi menggunakan Spring Security, dengan menggabungkan NFC *Card* sebagai faktor kepemilikan fisik dan PIN sebagai faktor pengetahuan yang divalidasi secara berurutan. Implementasi MFA terbukti efektif mencegah *buddy punching* karena sistem mensyaratkan kedua faktor terpenuhi bersamaan, di mana autentikasi gagal jika hanya meminjamkan kartu tanpa PIN maupun sebaliknya. Spring Security terbukti menjadi *framework* yang tepat melalui mekanisme keamanan berlapis mencakup BCrypt *hashing*, *account locking* setelah 3 kali PIN salah, autentikasi JWT, dan *role-based access control* sesuai prinsip *least privilege*. Dari sisi performa, seluruh *endpoint* kritis merespons di bawah 500ms dengan konsistensi tinggi, membuktikan sistem layak digunakan dalam kondisi operasional nyata.

Dibanding penelitian terdahulu, sistem MFA memiliki keunggulan tingkat pencegahan *buddy punching* yang tinggi dengan ketergantungan lingkungan yang rendah — tidak terpengaruh kondisi pencahayaan seperti *face recognition* dan tidak rentan peminjaman kartu seperti NFC tunggal — sehingga memberikan keseimbangan optimal antara keamanan dan kemudahan penggunaan. Meskipun demikian, pengujian masih terbatas pada lingkungan terkontrol dengan maksimal 10 pengguna konkuren dan belum mencakup audit keamanan eksternal oleh pihak ketiga independen. Untuk penelitian selanjutnya, disarankan pengujian skalabilitas dengan beban pengguna lebih besar, integrasi autentikasi biometrik sebagai faktor ketiga, serta implementasi mode *offline* dengan mekanisme sinkronisasi data lokal.

.

DAFTAR PUSTAKA

[1] T. Saputra *et al.*, “Peningkatan disiplin kerja karyawan dengan menggunakan sistem absensi self potrait dan geolocation pada pt sucofindo palembang,” *KLIK - J. Ilmu Komput.*, vol. 4, no. 2, 2023.

[2] F. Agustiawan dan P. Rosyani, “IMPLEMENTASI APLIKASI PRESENSI DAN PENGGAJIAN MENGGUNAKAN METODE AGILE (STUDI KASUS : PT . MAJU BERJAYA TEKNOLOGI),” *J. UNGGUL Teknol. DAN Inform.*, vol. 1, no. 1, hal. 21–25, 2024.

[3] S. Agustin, “Dampak Kemajuan Teknologi Informasi Era Digital Terhadap Keamanan Data Pribadi Tantangan Dan Penanggulangan Terhadap Kejahatan Cyber,” *J. Penelit. MULTIDISIPLIN BANGSA*, vol. 1, no. 6, hal. 500–504, 2024.

[4] J. K. Wijaya, “PERLINDUNGAN DATA PRIBADI DALAM ERA DIGITAL: TANTANGAN DAN SOLUSI,” *Meraja J.*, vol. 7, no. 2, hal. 138–155, 2024.

[5] N. Maret, A. Putra, U. Mansyuri, G. Dwiki, dan P. Aryono, “Analisis Penggunaan Sistem Face Recognition dalam Mengelola Absensi Karyawan di PT Bintang Inspeksi Indonesia Universitas Bina Bangsa , Indonesia transformasi digital perusahaan . Dengan dukungan teknologi ini , perusahaan berharap,” *J. Ilm. Sist. Inf. dan Ilmu Komput.*, vol. 5, no. 1, hal. 01–08, 2025.

[6] N. A. Karim, H. Kanaker, W. K. Abdulraheem, M. A. Ghaith, A. Mousa, dan F. Alali, “Choosing the right MFA method for online systems: A comparative analysis,” *Int. J. Data Netw. Sci.*, vol. 8, hal. 201–212, 2024, doi: 10.5267/j.ijdns.2023.10.003.

[7] I. G. Sujana, E. Putra, N. Luh, dan P. Labasariyani, “Lecturer Attendance System Based on Near Field Communication Technology,” *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 9, no. 1, hal. 25–31, 2021.

[8] M. H. M. Kamil, N. Zaini, L. Mazalan, dan A. H. Ahamad, “Online attendance system based on facial recognition with face mask detection,” *Online Attend. Syst. based facial Recognit. with face mask Detect.*, vol. 82, hal. 34437–34457, 2023, [Daring]. Tersedia pada:

https://doi.org/10.1007/s11042-023-14842-y

[9] I. Cahyanto *et al.*, "EFFECTIVENESS OF MULTIFACTOR AUTHENTICATION TECHNOLOGY FOR PROTECTING STUDENT PRIVACY : A SYSTEMATIC LITERATURE REVIEW," *Edum J.*, vol. 7, no. 2, hal. 253–269, 2024.

[10] S. Shukla, G. Varshney, S. Singh, dan S. Goel, "A Passwordless MFA Utlizing Biometrics , Proximity and Contactless Communication," *Inf. Secur. J. A Glob. Perspect.*, vol. 34, no. 6, hal. 633–654, 2025.

[11] A. C. Study dan A. Prinz, "Applying Spring Security Framework with KeyCloak-Based OAuth2 to Protect Microservice Architecture APIs :," *MDPI*, vol. 22, no. 5, hal. 1703, 2022.

[12] H. L. Id dan D. Liu, "An improved NFC device authentication protocol," *PLoS One*, hal. 1–8, 2021, doi: 10.1371/journal.pone.0256367.

[13] N. D. Arizona, R. Adwiya, A. Info, A. Algorithm, dan W. Based, "Implementation of the Waterfall Method in Designing and Building an Income and Cost Management Information System (Case study : Limited Liability Company Adau Kapuas)," *Bull. Comput. Sci. Electr. Eng.*, vol. 4, no. 2, hal. 65–76, 2023, doi: 10.25008/bcsee.

[14] A. Bucko, K. Vishi, dan B. Krasniqi, "Enhancing JWT Authentication and Authorization in Web Applications Based on User Behavior History," *MDPI*, hal. 1–18, 2023, doi: https://www.mdpi.com/2073-431X/12/4/78.

[15] C. Skanda, B. Srivasta, dan P. B.S., "Secure Hashing using BCrypt for Cryptographic Applications," in *2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*, 2022. doi: https://ieeexplore.ieee.org/document/10126956.

[16] OWASP (Open Worldwide Application Security Project), "A01:2021 – Broken Access Control," OWASP Top 10. [Daring]. Tersedia pada: https://owasp.org/Top10/2021/A01_2021-Broken_Access_Control/