

Implementasi Sistem Autentikasi Dua Faktor OTP pada Wi-Fi: Perbandingan HOTP dan TOTP

Maya Rahayu¹, Ratu Fayyaza Ayu Humayra², Fajar Gilang Trifebriyanto³

^{1,2,3}Program Studi D4 Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Bandung, Indonesia

¹mayarahayu@polban.ac.id, ²ratufayyazaah11@gmail.com, ³fajar.gilang.tkom422@polban.ac.id

Abstrak

Keamanan akses jaringan Wi-Fi menjadi isu penting seiring meningkatnya penetrasi internet, khususnya pada lingkungan kampus yang rentan terhadap akses tidak sah apabila hanya mengandalkan autentikasi berbasis *password*. Penelitian ini merancang dan mengimplementasikan sistem autentikasi dua faktor (2FA) berbasis *One-Time Password* (OTP) pada jaringan Wi-Fi menggunakan *captive portal* terintegrasi FreeRADIUS, sekaligus membandingkan performa *HMAC-based One-Time Password* (HOTP) dan *Time-based One-Time Password* (TOTP). Sistem dikembangkan secara eksperimental menggunakan dua *virtual machine* berbasis FreeRADIUS, Flask, MySQL, dan Nginx. Pengujian utama meliputi 30 kali percobaan *login* per metode serta *User Acceptance Testing* (UAT) terhadap 50 responden. Pengujian tambahan berupa serangan *brute force* dan beban pengguna simultan (5, 10, 15 pengguna, masing-masing diulang 10 kali) juga dilakukan pada kedua *endpoint* HOTP dan TOTP untuk mengevaluasi ketahanan keamanan dan skalabilitas sistem. Hasil pengujian menunjukkan HOTP mencatatkan *success rate* 96,67% (waktu verifikasi 83,767 ms), sedangkan TOTP 93,33% (93,967 ms), dengan nilai UAT 4,36 dari skala 5. Pengujian *brute force* awal mengungkap ketiadaan *rate limiting* pada kedua *endpoint* OTP, yang ditindaklanjuti dengan mekanisme penguncian setelah lima kali kegagalan berturut-turut dan terbukti efektif memblokir percobaan pada pengujian ulang di kedua *endpoint*. Pengujian pengguna simultan mengungkap keterbatasan Flask *development server* yang hanya dapat diperbaiki sebagian melalui pengaktifan mode *multi-thread*, dengan *success rate* menurun secara konsisten pada kedua metode seiring bertambahnya beban hingga mencapai 0% pada 15 pengguna simultan. TOTP direkomendasikan untuk implementasi jangka panjang karena sifat *time-bound*-nya yang lebih tahan terhadap *replay attack*, sementara temuan keamanan dan skalabilitas menjadi masukan penting untuk penyempurnaan sistem sebelum digunakan pada skala produksi.

Kata kunci: autentikasi dua faktor, *one-time password*, HOTP, TOTP, *captive portal*

1. Pendahuluan

Tingginya penetrasi internet di Indonesia yang mencapai 79,5% atau sekitar 221 juta pengguna pada 2024 (Haryanto, 2024) menjadikan keamanan akses jaringan *wireless* sebagai kebutuhan yang mendasak. Sejumlah kasus penyalahgunaan Wi-Fi publik, seperti pemalsuan *access point* (*evil twin*) dan penyadapan kredensial, menunjukkan bahwa autentikasi satu faktor berbasis *password* saja tidak lagi memadai, terutama pada lingkungan kampus yang menjadi *hotspot* bagi banyak pengguna (Widodo et al., 2022).

Autentikasi dua faktor (2FA) berbasis *One-Time Password* (OTP) banyak diterapkan untuk menambah lapisan verifikasi di luar *username* dan *password*. Dua algoritma OTP yang paling umum digunakan adalah *HMAC-based One-Time Password* (HOTP), yang membangkitkan kode berdasarkan nilai *counter*, dan *Time-based One-Time Password* (TOTP), yang membangkitkan kode berdasarkan interval waktu (Lumburovska et al., 2021). Namun

demikian, penerapan OTP pada sistem *login* Wi-Fi memiliki tantangan tersendiri karena pengguna umumnya belum memperoleh akses internet pada tahap awal autentikasi, sehingga OTP berbasis e-mail atau SMS eksternal tidak selalu dapat diakses secara langsung.

Penelitian terdahulu mengenai keamanan jaringan *wireless*, seperti penerapan 2FA pada sistem inventarisasi (Nuryati et al., 2022) dan autentikasi *captive portal* berbasis Mikrotik (Sukendar et al., 2023), umumnya berfokus pada penerapan satu algoritma OTP tanpa membandingkan performa antar-algoritma secara empiris dalam skenario Wi-Fi yang terkontrol. Penelitian mengenai TOTP dengan algoritma PHOTON (Amry, 2025) dan penerapan OAuth 2.0 dengan OTP pada *Single Sign On* (Arianto et al., 2025) juga menunjukkan tren peningkatan kombinasi OTP dengan mekanisme keamanan lain, namun belum menyentuh keterbatasan konektivitas awal pada *captive portal* Wi-Fi yang menjadi fokus penelitian ini.

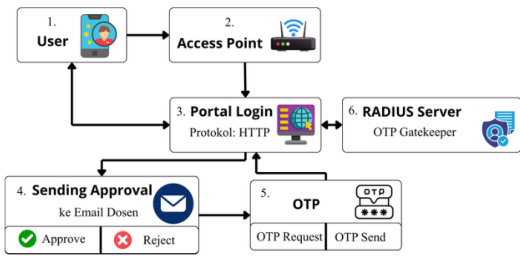
Tabel 1. Perbandingan penelitian terkait

Peneliti	Fokus Penelitian	Celah yang diisi penelitian ini
Nuryati dkk. (2022)	2FA berbasis OTP via SMS pada sistem inventarisasi	Satu metode OTP saja, tidak ada perbandingan algoritma, bukan <i>captive portal</i> Wi-Fi
Lumburovska dkk. (2021)	Analisis komparatif konseptual HOTP vs TOTP	Bersifat teoretis, belum diimplementasikan pada sistem nyata
Amry (2025)	TOTP dengan algoritma PHOTON pengganti SHA-1	Fokus efisiensi komputasi, tidak ada lapisan persetujuan
Sukendar dkk. (2023)	<i>Captive portal</i> Mikrotik pengganti WPA2-PSK	Belum ada verifikasi OTP tambahan atau perbandingan algoritma
Mayorga & Yoo (2025)	Evaluasi praktis solusi OTP untuk 2FA	Tidak diterapkan pada konteks <i>captive portal</i> Wi-Fi tertentu
Rivera-Dourado dkk. (2024)	<i>Captive portal</i> terintegrasi FIDO2/WebAuthn	Menggunakan FIDO2, tidak membandingkan algoritma OTP
Penelitian ini	HOTP vs TOTP + <i>human-in-the-loop</i> pada <i>captive portal</i> FreeRADIUS	–

Berdasarkan celah tersebut, penelitian ini mengembangkan mekanisme autentikasi dua faktor berbasis OTP pada *captive portal* Wi-Fi, dengan OTP yang diaktifkan setelah permintaan akses disetujui oleh pihak berwenang melalui e-mail, sekaligus membandingkan performa HOTP dan TOTP berdasarkan sinkronisasi kode, *delay* verifikasi, dan tingkat keberhasilan *login*, untuk merekomendasikan algoritma yang paling sesuai diterapkan pada sistem autentikasi Wi-Fi berbasis *captive portal* terintegrasi FreeRADIUS. Kontribusi utama penelitian ini adalah penambahan lapisan persetujuan manusia (*human-in-the-loop*) melalui e-mail dosen sebagai faktor kontrol akses tambahan, yang belum banyak dijumpai pada penelitian autentikasi Wi-Fi berbasis OTP sebelumnya.

Penelitian ini dibatasi pada implementasi sistem di lingkungan Wi-Fi skala lokal menggunakan dua *virtual machine* tanpa mencakup jaringan publik atau skala luas. Metode autentikasi dua faktor yang dibandingkan terbatas pada HOTP dan TOTP dengan mekanisme persetujuan akses melalui e-mail pihak berwenang, sementara pengujian *client* dilakukan hanya menggunakan *smartphone* yang terhubung ke jaringan Wi-Fi. Parameter pengujian difokuskan pada sinkronisasi kode OTP, *delay* verifikasi, tingkat keberhasilan autentikasi, keberhasilan open *access*, durasi proses *login*, ketahanan terhadap percobaan *brute force*, serta kemampuan sistem menangani beban pengguna simultan, yang diuji pada kedua metode HOTP dan TOTP.

Berdasarkan Tabel 1, penelitian sebelumnya umumnya berfokus pada satu aspek autentikasi tanpa menghadirkan perbandingan HOTP dan TOTP se-



Gambar 1. Diagram blok sistem autentikasi dua faktor

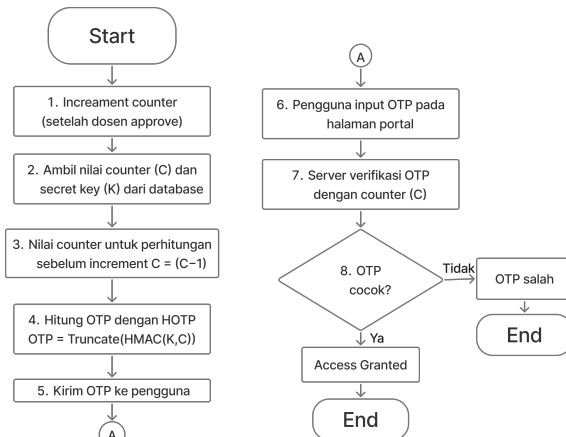
cara empiris pada konteks *captive portal* Wi-Fi yang dilengkapi lapisan persetujuan *human-in-the-loop*, sehingga menjadi celah yang diisi oleh penelitian ini.

2. Metode

Penelitian ini menggunakan metode eksperimental, yaitu merancang, mengimplementasikan, dan menguji sistem autentikasi dua faktor berbasis OTP pada jaringan Wi-Fi dalam lingkungan uji terkontrol berskala laboratorium. Sistem terdiri atas tiga sub-sistem utama yang saling terintegrasi: *Authentication & OTP Engine*, *Captive Portal & User Interface*, serta *Access Point & RADIUS*, yang diimplementasikan menggunakan dua *virtual machine* (VM) berbasis VirtualBox pada jaringan *host-only*, dengan *access point* fisik (USB Wi-Fi adapter) sebagai satu-satunya komponen perangkat keras nyata pada topologi pengujian. VM pertama berfungsi sebagai *server* FreeRADIUS, sedangkan VM kedua berfungsi sebagai *web server* berbasis Flask, MySQL, dan Nginx.

Proses autentikasi dimulai ketika pengguna terhubung ke *access point* dan diarahkan ke halaman *captive portal* untuk *login* menggunakan *username* dan *password* serta memilih dosen dan kelas. Setelah data terverifikasi, sistem mengirimkan notifikasi persetujuan ke e-mail dosen sebagai lapisan *human-in-the-loop*. Apabila dosen menyetujui, sistem membangkitkan kode OTP sesuai metode yang dipilih (HOTP atau TOTP) untuk diinputkan kembali oleh pengguna. Apabila OTP valid, FreeRADIUS mengirimkan *Access-Accept* sehingga *access point* memberikan akses jaringan. Sebaliknya, sistem mengirimkan *Access-Reject*.

Gambar 1 menunjukkan diagram blok sistem. Proses autentikasi dimulai ketika pengguna (*User*) terhubung ke *access point*, lalu diarahkan ke portal *login* melalui protokol HTTP. *Web server* mengirimkan permintaan OTP sekaligus notifikasi *approval* ke e-mail dosen. Apabila dosen memberikan persetujuan, kode OTP dikirimkan ke pengguna dan diinputkan kembali ke portal untuk menyelesaikan proses autentikasi sebelum FreeRADIUS memberikan *Access-Accept* atau *Access-Reject* ke *access point*.



Gambar 2. Alir kerja algoritma HOTP pada sistem

2.1 Algoritma HOTP dan TOTP

HOTP membangkitkan kode OTP berdasarkan kunci rahasia (K) dan nilai *counter* (C) yang diproses menggunakan fungsi HMAC-SHA-1, dirumuskan sebagai:

$$HOTP(K, C) = \text{HMAC-SHA-1}(K, C) \quad (1)$$

Nilai keluaran HMAC-SHA-1 sepanjang 160 bit kemudian melalui proses *dynamic truncation* dan operasi *modulo* agar menghasilkan kode numerik 6 digit. TOTP merupakan pengembangan HOTP yang menggantikan nilai *counter* dengan parameter waktu (T), dirumuskan sebagai:

$$TOTP = HOTP(K, T) \quad (2)$$

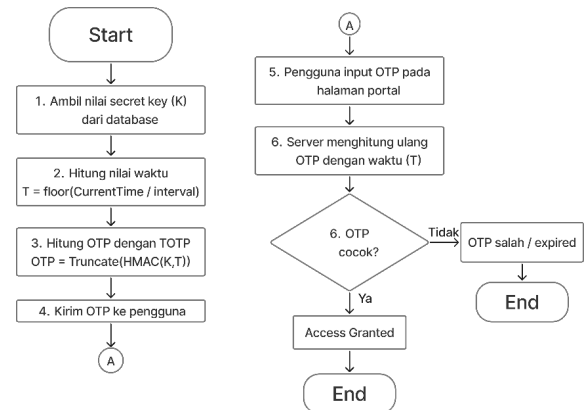
dengan T merepresentasikan langkah waktu (*time-step*) yang dihitung menggunakan persamaan:

$$T = \frac{\text{Current UNIX Time} - T_0}{X} \quad (3)$$

dengan *Current UNIX Time* adalah waktu sistem saat ini, T_0 adalah waktu awal perhitungan (umumnya bernilai 0), dan X adalah interval waktu yang digunakan untuk menghasilkan OTP baru, yang pada penelitian ini ditetapkan selama 30 detik. *Server* verifikasi tidak dapat mengetahui secara pasti kapan OTP dibuat oleh pengguna karena keterlambatan jaringan, sehingga diberikan toleransi waktu yang tidak melebihi satu periode *time-step* agar risiko keamanan tetap terkendali.

Gambar 2 menunjukkan alur kerja HOTP: nilai *counter* bertambah setelah dosen memberikan persetujuan, kemudian *counter* (sebelum *increment*) dan *secret key* diambil dari *database* untuk membangkitkan OTP melalui fungsi HMAC. *Server* melakukan perhitungan ulang dengan nilai *counter* yang sama untuk memverifikasi kecocokan kode yang dimasukkan pengguna, apabila tidak cocok maka autentikasi berhasil, sedangkan apabila tidak cocok maka autentikasi dinyatakan gagal dan proses dihentikan.

Gambar 3 menunjukkan alur kerja TOTP, yang serupa dengan HOTP namun menggantikan nilai *counter* dengan langkah waktu (*time-step*) saat ini. *Server* menghitung ulang nilai T berdasarkan waktu



Gambar 3. Alir kerja algoritma TOTP pada sistem

saat ini dan *secret key* untuk membangkitkan OTP pembandingan, kemudian memverifikasi kecocokannya dengan kode yang dimasukkan pengguna dalam batas toleransi satu *time-step*. Berbeda dengan HOTP, sistem turut memeriksa apakah waktu tampil OTP telah melewati batas kedaluwarsa sebelum verifikasi dilakukan, apabila telah kedaluwarsa, sistem langsung mengirimkan *Access-Reject* tanpa memproses verifikasi kode.

2.2 Keamanan Data Pengguna

Untuk mengurangi risiko kebocoran data apabila *database* disusupi, *password* pengguna diamankan menggunakan *hashing bcrypt*, sedangkan *secret key* OTP diamankan menggunakan enkripsi *symmetric Fernet* pada lapisan aplikasi. *Shared secret* komunikasi antara Flask dan FreeRADIUS dipindahkan dari kode sumber ke berkas konfigurasi lingkungan (.env) menggunakan *python-dotenv* agar tidak tersimpan secara *hardcoded*. Sebagai lapisan tambahan, diterapkan pula mekanisme *session timeout* otomatis untuk membatasi durasi sesi *login* yang tertinggal aktif.

Perlu diperjelas mengenai kanal pengiriman OTP pada sistem ini, kode OTP tidak dikirimkan melalui kanal terpisah seperti SMS atau aplikasi *authenticator* pada perangkat lain, melainkan ditampilkan langsung pada halaman portal yang sama setelah dosen memberikan persetujuan melalui e-mail. Dengan demikian, faktor verifikasi kedua pada sistem ini secara *arsitektural* bertumpu pada langkah persetujuan *human-in-the-loop* oleh dosen (yang berlangsung pada kanal e-mail terpisah), bukan pada pemisahan kanal pengiriman kode OTP itu sendiri. Desain ini berbeda dari definisi 2FA klasik yang umumnya mensyaratkan faktor kedua berupa “sesuatu yang dimiliki” pada perangkat/kanal independen pada sistem ini, kode OTP berfungsi sebagai *token* sekali pakai yang baru aktif setelah otorisasi pihak berwenang, sehingga keamanan tambahan yang diperoleh lebih bersifat kontrol otorisasi berlapis daripada pemisahan kanal fisik. Penguatan lebih lanjut, seperti pengiriman OTP melalui kanal terpisah (SMS atau aplikasi *authenticator*), disarankan seba-

Tabel 2. Spesifikasi lingkungan pengujian

Komponen	Spesifikasi
VM 1 (RADIUS)	FreeRADIUS, IP 192.168.56.10
VM 2 (Web server)	Flask, MySQL, Nginx, IP 192.168.56.20
Access point	USB Wi-Fi adapter (hostapd + dnsmasq)
Client	Smartphone Android/iOS
Notifikasi	SMTP Gmail (port 587)

gai pengembangan lanjutan untuk memenuhi definisi 2FA secara lebih ketat.

Perlu dicatat pula bahwa komunikasi antara *client* dan *captive portal* pada penelitian ini masih berjalan melalui protokol HTTP, bukan HTTPS, sehingga kredensial *login* dan kode OTP berpotensi diteruskan dalam bentuk *cleartext* dan rentan terhadap penyadapan (*sniffing*) pada jaringan Wi-Fi yang sama. Kondisi ini merupakan keterbatasan signifikan yang secara langsung berkontradiksi dengan motivasi keamanan penelitian, dan penerapan HTTPS melalui sertifikat TLS pada *captive portal* menjadi prioritas perbaikan yang sangat disarankan sebelum sistem dipertimbangkan untuk penggunaan produksi.

2.3 Skenario Pengujian

Pengujian dilakukan pada topologi Wi-Fi skala lokal dengan *smartphone* sebagai *client*. Pengujian utama dilakukan sebanyak 30 kali percobaan *login* pada masing-masing metode (HOTP dan TOTP), mengukur tingkat keberhasilan autentikasi, waktu verifikasi OTP, dan total waktu autentikasi. Pengujian tambahan berupa skenario kegagalan (*input* OTP salah dan penolakan oleh dosen) juga dilakukan masing-masing 30 kali percobaan per metode untuk menguji konsistensi deteksi kegagalan sistem. Penerimaan pengguna diukur melalui kuesioner *User Acceptance Testing* (UAT) terhadap 50 responden pada enam aspek: keamanan, keandalan sistem, kemudahan penggunaan, efisiensi waktu, tampilan antarmuka, dan kepuasan pengguna, menggunakan skala *Likert* 1-5. Dua skenario pengujian tambahan turut dilakukan pada kedua metode untuk memperdalam evaluasi keamanan dan performa sistem dengan melakukan pengujian *brute force*, yaitu 50 kali percobaan penebak kode OTP secara otomatis terhadap *endpoint* verifikasi untuk mengukur ketahanan sistem terhadap serangan penebak, dan pengujian pengguna simultan, yaitu simulasi *login* konkuren pada tiga tingkat beban (5, 10, dan 15 pengguna sekaligus) menggunakan *multi-thread* untuk mengukur *success rate* dan waktu respons sistem di bawah beban akses bersamaan.

Tabel 3. Ringkasan hasil pengujian fungsionalitas

Fitur	Hasil
<i>Signup & login</i>	Berhasil (✓)
Pemilihan dosen & kelas	Berhasil (✓)
Notifikasi & <i>approve/reject</i> dosen	Berhasil (✓)
Tampilan & verifikasi OTP (HOTP/TOTP)	Berhasil (✓)
<i>Access granted & redirect portal</i>	Berhasil (✓)
<i>Polling</i> status persetujuan	Berhasil (✓)

Tabel 4. Hasil pengukuran tingkat keberhasilan autentikasi

Parameter	HOTP	TOTP
Jumlah percobaan	30	30
<i>Login</i> sukses	30	30
<i>Access Granted</i>	29	28
OTP <i>Expired</i>	–	1
Ditolak RADIUS	1	1
<i>Success rate</i> (%)	96,67	93,33

3. Hasil dan Pembahasan

3.1 Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan untuk memastikan seluruh fitur sistem berjalan sesuai rancangan. Hasil pengujian pada 12 fitur utama, dikelompokkan menjadi 6 kategori pada Tabel 3, mulai dari *signup*, *login*, pemilihan dosen dan kelas, pengiriman notifikasi e-mail, halaman *approve/reject* dosen, tampilan OTP, verifikasi OTP HOTP dan TOTP, pemberian akses, hingga *redirect captive portal* dan *polling* status persetujuan.

Berdasarkan Tabel 3, seluruh fungsi berhasil dijalankan tanpa ditemukan kegagalan fungsional, yang mengonfirmasi bahwa integrasi antara *access point*, FreeRADIUS, *backend* Flask, dan *captive portal* telah berjalan dengan baik sebagai satu kesatuan sistem yang saling mendukung.

3.2 Tingkat Keberhasilan Autentikasi

Pada pengujian utama, dari 30 kali percobaan pada masing-masing metode, seluruh proses *login* berhasil (30/30) tanpa kegagalan pada tahap awal autentikasi. HOTP mencatatkan *success rate* verifikasi OTP sebesar 96,67%, sedangkan TOTP sebesar 93,33%, dengan selisih disebabkan oleh satu kasus OTP *expired* pada TOTP yang tidak dialami HOTP karena HOTP tidak terikat batas waktu. Rincian hasil ditampilkan pada Tabel 4.

Untuk menguji apakah selisih *success rate* 96,67% (29/30) berbanding 93,33% (28/30) tersebut signifikan secara statistik, dilakukan uji *Fisher's exact test* pada tabel kontingensi 2x2 (berhasil/gagal x HOTP/TOTP). Hasil uji menunjukkan *p-value* = 1,00 (jauh di atas ambang signifikansi $\alpha = 0,05$), dengan interval kepercayaan 95% yang saling tumpang

Tabel 5. Hasil rata-rata waktu proses autentikasi

Parameter waktu	HOTP	TOTP
Delay e-mail sent (ms)	5.133	5.900
OTP tampil (ms)	0,084	0,039
Response time OTP (ms)	83,767	93,967
Total waktu autentikasi (ms)	5.216,851	5.994,006

tindih lebar antara HOTP (83,3%-99,4%) dan TOTP (78,7%-98,2%) berdasarkan estimasi Wilson. Dengan demikian, selisih satu kegagalan dari 30 percobaan pada masing-masing metode tidak dapat disimpulkan sebagai perbedaan performa yang signifikan antara HOTP dan TOTP, melainkan variasi yang wajar pada ukuran sampel sebesar ini.

Pada pengujian skenario kegagalan spesifik (OTP salah dan penolakan dosen), sistem berhasil mendeteksi dan menangani seluruh 30 kali percobaan pada masing-masing skenario dan metode, dengan *detection rate* 100%, yang membuktikan keandalan mekanisme validasi dan alur persetujuan sistem.

3.3 Waktu Proses Autentikasi

Rata-rata waktu verifikasi OTP tercatat 83,767 ms untuk HOTP dan 93,967 ms untuk TOTP, sementara total waktu autentikasi rata-rata tercatat 5.216,851 ms berbanding 5.994,006 ms. Rincian ditampilkan pada Tabel 5.

Selisih waktu verifikasi OTP antara HOTP dan TOTP pada Tabel 5 hanya sekitar 10 ms. Sebagai pembanding, pengujian *brute force* pada Bagian 3.5 mengukur *response time* dengan metode berbeda (*round-trip* HTTP eksternal, lihat catatan metodologis pada Bagian 3.5) namun dengan sampel yang jauh lebih besar (50 percobaan per metode), pada pengujian tersebut, selisih HOTP dan TOTP juga tetap kecil dan tidak konsisten arahnya (13-26 ms, kadang HOTP lebih lambat, kadang lebih cepat). Konsistensi temuan “selisih kecil dan tidak konsisten arah” pada dua metode pengukuran yang berbeda ini memperkuat interpretasi bahwa selisih 10 ms pada Tabel 5 lebih tepat dipandang sebagai variasi pengukuran (*noise*) yang wajar terjadi pada sampel kecil, bukan sebagai bukti bahwa salah satu algoritma secara konsisten lebih cepat secara komputasional. Klaim keunggulan kecepatan salah satu metode sebaiknya tidak ditarik dari data ini.

Temuan yang lebih substansial dari pengujian waktu adalah bahwa *bottleneck* utama sistem bukan berada pada proses kriptografis OTP, melainkan pada tahap pengiriman e-mail notifikasi persetujuan ke dosen, yang menyumbang lebih dari 98% dari keseluruhan waktu autentikasi pada kedua metode. Hal ini sejalan dengan penelitian mengenai integrasi FreeRADIUS pada jaringan *hotspot* yang juga menyoroti pentingnya optimasi komponen komunikasi eksternal terhadap performa autentikasi secara keselu-

Tabel 6. Rekapitulasi hasil UAT (6 aspek, 50 responden)

Aspek	SS/S (%)	Mean
Keamanan sistem	88	4,50
Keandalan sistem	90	4,44
Kemudahan penggunaan	86	4,42
Efisiensi waktu	76	4,12
Tampilan antarmuka	86	4,34
Kepuasan pengguna	84	4,40

ruhan (Pratama et al., 2023).

3.4 Penerimaan Pengguna (UAT)

Hasil UAT terhadap 50 responden menunjukkan nilai rata-rata keseluruhan 4,36 dari skala 5. Aspek keamanan sistem memperoleh penilaian tertinggi (4,50), diikuti keandalan sistem (4,44) dan kemudahan penggunaan (4,42), yang mencerminkan tingkat kepercayaan tinggi terhadap stabilitas dan keamanan sistem. Aspek efisiensi waktu memperoleh nilai terendah (4,12), sejalan dengan temuan bahwa durasi autentikasi sangat dipengaruhi oleh kecepatan respons dosen dalam memberikan persetujuan.

Kehadiran tiga lapisan verifikasi, yakni kredensial, persetujuan dosen, dan OTP secara bersamaan, menjadikan sistem lebih sulit ditembus dibandingkan autentikasi Wi-Fi konvensional yang hanya mengandalkan password, konsisten dengan temuan bahwa 2FA secara signifikan mempersempit peluang penyerang dibandingkan autentikasi satu faktor (Azkiya et al., 2026). Mekanisme persetujuan dosen menjadi keunggulan unik yang membedakan sistem ini dari penelitian autentikasi *captive portal* sebelumnya (Rivera-Dourado et al., 2024), karena unsur *human-in-the-loop* dihadirkan sebagai lapisan kontrol yang tidak dapat dilewati secara otomatis.

Secara kualitatif, HOTP dan TOTP menawarkan mitigasi terhadap sejumlah skenario serangan yang tidak dimiliki autentikasi Wi-Fi konvensional berbasis *password* saja, seperti *unauthorized access* dan *password sharing*, karena akses tetap memerlukan kode OTP dinamis di samping kredensial. Karakteristik *time-bound* pada TOTP secara konseptual memberi keunggulan tambahan pada skenario yang bergantung pada jendela waktu penyerangan, seperti *OTP interception* dan *replay attack*, karena kode yang disadap akan kedaluwarsa dalam hitungan detik. Namun demikian, penilaian keamanan ini perlu dibaca berdampingan dengan hasil pengujian empiris pada Bagian 3.5: sebelum mekanisme *rate limiting* diimplementasikan, kedua metode terbukti sama-sama rentan terhadap percobaan *brute force* tanpa batas, sehingga klaim ketahanan terhadap serangan penebakan hanya berlaku setelah perbaikan tersebut diverifikasi, bukan merupakan sifat bawaan algoritma sejak awal.

Password sharing dan *credential stuffing* turut dapat dicegah pada kedua algoritma karena akses

tetap memerlukan persetujuan dosen dan kode OTP yang dinamis, sehingga kredensial saja tidak cukup untuk memperoleh akses (Nurandani et al., 2025). Namun demikian, perlindungan 2FA pada penelitian ini terbatas pada proses autentikasi saat *login* dan belum menjangkau keamanan jalur data setelah pengguna terautentikasi maupun selama proses *login* itu sendiri, sebagaimana dibahas pada Bagian 2.2, transmisi kredensial dan OTP melalui HTTP tanpa enkripsi TLS membuat sistem tetap rentan terhadap penyadapan pada jaringan yang sama, terlepas dari algoritma OTP yang digunakan. Serangan *man-in-the-middle* dan *session hijacking* karena itu hanya dapat dimitigasi sebagian pada kedua metode, karena kanal komunikasi belum dilindungi enkripsi *end-to-end* baik sebelum maupun sesudah autentikasi berhasil.

3.5 Pengujian Brute Force

Pengujian tambahan dilakukan berupa serangan *brute force* terhadap *endpoint* verifikasi OTP untuk mengukur ketahanan sistem terhadap percobaan penembakan kode secara berulang. Perlu dicatat bahwa waktu respons yang dilaporkan pada bagian ini dan Bagian 3.6 diukur secara eksternal menggunakan skrip Python yang menghitung total waktu *round-trip* HTTP dari saat permintaan dikirim hingga respons lengkap diterima (mencakup latensi jaringan, *over-head* Nginx, dan pemrosesan Flask), berbeda dari waktu respons OTP pada Tabel 5 yang diukur secara internal di dalam kode aplikasi (hanya mencakup waktu komputasi verifikasi itu sendiri). Perbedaan metode pengukuran ini menyebabkan magnitud waktu yang dilaporkan pada kedua bagian tidak dapat dibandingkan secara langsung, meskipun keduanya sama-sama valid untuk konteks pengujian masing-masing. Pengujian awal dilakukan dengan mengirimkan 50 tebakan OTP acak enam digit secara berurutan terhadap sesi pengguna berstatus *approved* pada kedua *endpoint*, HOTP dan TOTP, menggunakan skrip otomatis. Hasil pengujian awal menunjukkan tidak satu pun dari 50 tebakan berhasil menembus verifikasi pada kedua metode, namun sistem juga tidak menunjukkan mekanisme pembatasan percobaan (*rate limiting*) maupun penguncian akun (*account lockout*), dengan rata-rata waktu respons *round-trip* yang relatif setara antara HOTP (1.329,10 ms) dan TOTP (1.315,72 ms) per percobaan.

Sebagai tindak lanjut atas temuan tersebut, diimplementasikan mekanisme *rate limiting* pada *endpoint* verifikasi OTP: sistem akan mengunci sementara (selama 30 detik) sesi pengguna setelah lima kali percobaan OTP gagal berturut-turut, diterapkan pada kedua *endpoint* HOTP dan TOTP. Pengujian *brute force* diulang pada kedua *endpoint* setelah implementasi *rate limiting* untuk memverifikasi efektivitas perbaikan. Hasil pengujian sebelum dan sesudah perbaikan ditampilkan pada Tabel 7.

Tabel 7. Hasil pengujian *brute force* sebelum dan sesudah implementasi *rate limiting*

Kondisi	Jumlah percobaan	Rata-rata waktu/percobaan	Rate limiting terdeteksi
HOTP - sebelum <i>fix</i>	50	1.329,10 ms	Tidak
HOTP - sesudah <i>fix</i>	6	1.338,19 ms	Ya, pada percobaan ke-6
TOTP - sebelum <i>fix</i>	50	1.315,72 ms	Tidak
TOTP - sesudah <i>fix</i>	6	1.312,87 ms	Ya, pada percobaan ke-6

Pada seluruh pengujian, baik sebelum maupun sesudah perbaikan, tidak satu pun tebakan OTP berhasil menembus verifikasi. Setelah *rate limiting* diimplementasikan, kedua *endpoint* konsisten memblokir sesi pada percobaan ke-6 (setelah lima kali kegagalan berturut-turut) dengan respons HTTP 429, membuktikan perbaikan berhasil menutup celah *brute force* yang teridentifikasi pada pengujian awal. Menariknya, pada kedua kondisi (sebelum maupun sesudah *fix*) dengan status pengujian yang telah dipastikan benar (*approved*), waktu respons HOTP dan TOTP tercatat relatif setara (berkisar 1.310-1.340 ms), berbeda dari dugaan awal pada pengujian pendahuluan bahwa TOTP jauh lebih cepat, perbedaan besar yang sempat teramati sebelumnya ternyata disebabkan oleh perbedaan status sesi pengujian yang tidak konsisten, bukan oleh perbedaan komputasi verifikasi HOTP dan TOTP itu sendiri. Dengan waktu respons yang setara, ketahanan terhadap *brute force* pada kedua metode kini lebih ditentukan oleh karakteristik desainnya: TOTP tetap dinilai relatif lebih unggul secara konseptual karena kode yang valid berubah setiap 30 detik sehingga membatasi jendela peluang serangan, sedangkan kode HOTP tetap valid tanpa batas waktu hingga berhasil ditebak atau *di-resync*.

3.6 Pengujian Pengguna Simultan

Pengujian kedua yang dilakukan adalah pengujian jumlah pengguna simultan, untuk mengetahui perilaku sistem ketika diakses oleh banyak pengguna pada waktu bersamaan, mensimulasikan kondisi ketika banyak mahasiswa terhubung ke *captive portal* secara serentak. Pengujian dilakukan pada *endpoint login* HOTP dan TOTP, dengan mengirimkan permintaan *login* secara konkuren menggunakan *multi-thread* pada tiga tingkat beban: 5, 10, dan 15 pengguna simultan.

Pengujian awal (satu kali percobaan per tingkat beban) menunjukkan seluruh permintaan gagal total (*success rate* 0% pada ketiga tingkat beban) dengan galat HTTP 500/502. Investigasi log Nginx mengonfirmasi penyebabnya web server Flask dijalankan menggunakan *development server* bawaan (`app.run()`)

Tabel 8. Hasil pengujian pengguna simultan pada *endpoint* HOTP (rata-rata dari 10 pengulangan, setelah *threaded=True*)

Jumlah pengguna	Success rate (mean $\pm$ 95% CI)	Waktu respons (mean $\pm$ 95% CI)
5	90,00% $\pm$ 19,60%	6.889,03 $\pm$ 1.779,48 ms
10	40,00% $\pm$ 32,01%	9.134,76 $\pm$ 2.089,54 ms
15	0,00% $\pm$ 0,00%	9.694,63 $\pm$ 1.712,77 ms

Tabel 9. Hasil pengujian pengguna simultan pada *endpoint* TOTP (rata-rata dari 10 pengulangan, setelah *threaded=True*)

Jumlah pengguna	Success rate (mean $\pm$ 95% CI)	Waktu respons (mean $\pm$ 95% CI)
5	86,00% $\pm$ 19,42%	9.108,61 $\pm$ 1.833,16 ms
10	80,00% $\pm$ 26,13%	9.835,84 $\pm$ 1.477,83 ms
15	0,00% $\pm$ 0,00%	7.916,58 $\pm$ 926,44 ms

yang secara default bersifat *single-threaded*, sehingga hanya mampu memproses satu permintaan pada satu waktu. Ketika beberapa permintaan *login* tiba secara bersamaan, koneksi ditolak atau direset (*connection refused/reset by peer*), menyebabkan Nginx mengembalikan galat 502 Bad Gateway ke seluruh pengguna. Sebagai perbaikan awal, parameter *threaded=True* ditambahkan pada konfigurasi Flask untuk mengaktifkan pemrosesan *multi-thread*.

Mengingat satu kali percobaan per tingkat beban tidak cukup reliabel untuk menarik kesimpulan, pengujian pengguna simultan kemudian diulang sebanyak 10 kali secara independen untuk setiap tingkat beban pada masing-masing metode, dengan status seluruh akun pengguna *direset* ke idle sebelum setiap pengulangan agar setiap percobaan bersifat independen. Hasil dilaporkan sebagai rata-rata beserta 95% *confidence interval* (CI) dari 10 pengulangan tersebut, ditampilkan pada Tabel 8 (HOTP) dan Tabel 9 (TOTP).

Berdasarkan 10 pengulangan pada Tabel 8 dan Tabel 9, kedua metode menunjukkan pola degradasi yang konsisten dan monoton seiring bertambahnya jumlah pengguna simultan, *success rate* menurun dari kisaran 86-90% pada 5 pengguna, menjadi 40-80% pada 10 pengguna, hingga 0% pada 15 pengguna untuk kedua metode. Waktu respons yang dilaporkan mencakup seluruh percobaan (baik yang berhasil maupun gagal/*timeout*), sehingga pada 15 pengguna simultan, angka yang ditampilkan merepresentasikan rata-rata waktu hingga kegagalan tercatat (*connection timeout*), bukan waktu penyelesaian permintaan yang berhasil. *Confidence interval* yang relatif lebar pada beberapa titik data (misalnya $\pm 32,01\%$ pada HOTP $n=10$) mengindikasikan variabilitas antar-percobaan yang cukup tinggi, kemungkinan dipengaruhi oleh kondisi jaringan Wi-Fi lokal

dan waktu respons pengiriman e-mail notifikasi ke dosen yang bervariasi antar-pengulangan, di samping keterbatasan kapasitas *development server* itu sendiri. Konsistensi tren penurunan pada kedua metode sepanjang 10 pengulangan independen memperkuat validitas temuan bahwa mode *threaded=True* pada *development server* Flask hanya memberikan perbaikan parsial dan tidak cukup andal untuk menangani beban 15 pengguna simultan ke atas. Temuan ini menegaskan bahwa *deployment production* sebaiknya menggunakan WSGI *server* yang dirancang untuk menangani banyak koneksi konkuren (misalnya Gunicorn atau uWSGI dengan beberapa *worker*), bukan *development server* Flask, sejalan dengan peringatan resmi Flask bahwa *app.run()* tidak ditujukan untuk lingkungan produksi.

3.7 Perbandingan dengan Penelitian Terdahulu

Success rate HOTP dan TOTP pada penelitian ini (96,67% dan 93,33%) sejalan dengan temuan Lumburovska et al. (2021) yang menyimpulkan kedua algoritma sama-sama andal untuk autentikasi. Lumburovska et al. (2021) mencatat TOTP unggul dari sisi keamanan waktu-nyata karena sifat *time-bound*-nya, sementara HOTP dinilai lebih sederhana secara struktural karena tidak memerlukan sinkronisasi waktu antara *client* dan *server*, perlu ditegaskan bahwa penilaian “kesederhanaan struktural” ini berbeda dari klaim kecepatan komputasi, dan penelitian ini sendiri tidak menemukan bukti bahwa salah satu algoritma secara konsisten lebih cepat (lihat Bagian 3.3 dan 3.5). Nilai UAT 4,36 pada penelitian ini juga sebanding dengan penerimaan pengguna yang dilaporkan pada penelitian penerapan OTP untuk keamanan tambahan pada aplikasi finansial (Nurandani et al., 2025), yang turut melaporkan penerimaan pengguna tinggi terhadap lapisan verifikasi berbasis OTP meski disertai catatan pada aspek kecepatan proses. Perbedaan utama penelitian ini terletak pada penambahan lapisan persetujuan dosen sebagai kontrol *human-in-the-loop*, yang belum ditemukan pada studi *captive portal* maupun 2FA sebelumnya, sehingga memberikan lapisan keamanan tambahan yang unik meskipun berdampak pada durasi autentikasi secara keseluruhan.

4. Kesimpulan

Sistem autentikasi dua faktor berbasis OTP pada jaringan Wi-Fi berhasil dirancang dan diimplementasikan menggunakan *captive portal* yang terintegrasi dengan FreeRADIUS, Flask, MySQL, dan Nginx, dengan seluruh alur autentikasi berjalan penuh hingga pemberian akses jaringan. HOTP mencatatkan *success rate* 96,67% dengan rata-rata waktu verifikasi OTP 83,767 ms, sedangkan TOTP mencatatkan 93,33% dengan 93,967 ms, uji Fisher’s exact test ($p = 1,00$) menunjukkan selisih tersebut

tidak signifikan secara statistik, dan pengujian *brute force* dengan sampel lebih besar turut mengonfirmasi waktu respons kedua algoritma pada dasarnya setara. Berdasarkan hasil perbandingan keseluruhan, TOTP lebih direkomendasikan untuk implementasi jangka panjang karena sifatnya yang *time-bound* dan *self-recovering* sehingga lebih tahan terhadap *replay attack*, sementara hasil UAT terhadap 50 responden (rata-rata 4,36 dari skala 5) mengonfirmasi penerimaan pengguna yang positif terhadap sistem. Pengujian tambahan *brute force* awal mengungkap ketiadaan *rate limiting* pada *endpoint* verifikasi OTP, sebagai tindak lanjut, mekanisme penguncian sementara setelah lima kali kegagalan berturut-turut berhasil diimplementasikan dan diverifikasi efektif pada kedua metode, sedangkan pengujian pengguna simultan mengungkap keterbatasan *development server* Flask dalam menangani permintaan konkuren pada kedua metode, yang berhasil diperbaiki sebagian melalui pengaktifan mode *multi-thread* namun tetap menunjukkan penurunan keandalan pada beban pengguna yang lebih tinggi.

Password dan *secret key* pengguna juga telah diamankan melalui hashing *bcrypt* dan enkripsi *Fernet* pada lapisan aplikasi sehingga mengurangi risiko kebocoran data kredensial maupun kunci OTP apabila *database* berhasil disusupi. Tujuan penelitian untuk merancang, mengimplementasikan, dan membandingkan performa keamanan HOTP dan TOTP pada *captive portal* Wi-Fi berhasil dicapai, dengan kontribusi nyata berupa implementasi serta perbandingan empiris HOTP dan TOTP pada sistem autentikasi *captive portal* berbasis FreeRADIUS yang dilengkapi lapisan persetujuan *human-in-the-loop*.

Berdasarkan temuan pengujian *brute force* dan pengguna simultan, dua langkah perbaikan diajukan untuk penyempurnaan sistem sebelum digunakan pada skala produksi. Pertama, sebagai tindak lanjut atas celah *brute force* yang teridentifikasi, mekanisme *rate limiting* dan *account lockout* telah diimplementasikan pada *endpoint* verifikasi OTP (maksimal 5 kali percobaan gagal per sesi sebelum sesi dikunci sementara selama 30 detik) dan terbukti efektif memblokir percobaan *brute force* pada pengujian ulang di kedua metode (Tabel 7), parameter durasi kunci dan ambang batas percobaan ini dapat disesuaikan lebih lanjut sesuai kebutuhan operasional. Kedua, migrasi dari *development server* Flask (`app.run()`) ke *WSGI server production* seperti *Gunicorn* atau *uWSGI* dengan konfigurasi *multi-worker* sangat disarankan, mengingat pengujian pengguna simultan membuktikan *development server* tidak mampu menangani permintaan konkuren tanpa mode *multi-thread*, dan bahkan dengan mode tersebut performa masih menurun pada beban pengguna yang lebih tinggi.

Penelitian lanjutan disarankan menguji sistem pada skala jaringan penuh dengan banyak *access point* dan pengguna secara bersamaan baik pada lingkungan kampus maupun instansi dengan topologi

dan kebijakan keamanan yang berbeda, guna menguji skalabilitas dan generalisasi sistem pada kondisi yang lebih beragam. Mengingat *success rate* telah mencapai 0% pada 15 pengguna simultan menggunakan *development server* dengan mode *multi-thread*, penelitian lanjutan disarankan memetakan titik jenuh kapasitas secara lebih presisi pada rentang 10-15 pengguna, serta mengulang pengujian yang sama setelah migrasi ke *WSGI server production* untuk mengukur peningkatan kapasitas yang diperoleh. Optimasi pengiriman notifikasi e-mail secara *asynchronous* menggunakan *message queue* berbasis Celery dan Redis juga disarankan untuk mengurangi *bottleneck* utama pada waktu autentikasi yang saat ini menyumbang lebih dari 98% dari total durasi *login*.

Penambahan mekanisme *resync counter* otomatis pada HOTP sesuai rekomendasi RFC 4226 disarankan agar sistem dapat mentoleransi ketidaksinkronan *counter* tanpa reset manual. Penggunaan AES-256 atau *Hardware Security Module* (HSM) juga dapat dipertimbangkan untuk tingkat keamanan yang lebih tinggi pada implementasi skala produksi, disertai sinkronisasi status sesi pengguna dengan aturan *iptables* pada *access point* secara otomatis agar pemutusan akses jaringan terjadi tanpa jeda antara status di *database* dan status akses aktual. Pada sisi antarmuka, penambahan *countdown timer* visual, penjelasan singkat perbedaan HOTP dan TOTP, tombol salin kode OTP, serta fitur *trusted device* disarankan untuk meningkatkan efisiensi waktu penggunaan sesuai temuan UAT.

Daftar Pustaka

- Amry, Y., Ari, K., & Primantara, H. T. (2025): *Implementasi Time-Based One-Time Password menggunakan algoritma PHOTON untuk autentikasi dua faktor*, Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), 12(4), 799-808. <https://doi.org/10.25126/jtik.124>
- Arianto, I. G., Witanti, W., & Ashaury, H. (2025): *Sistem keamanan otentikasi pengguna pada modul Single Sign On menggunakan OAuth 2.0 dan One Time Password*, Jurnal Ilmu Komputer dan Teknologi (IKOMTI), 6(1), 25-31. <https://doi.org/10.35960/ikomti.v6i1.1768>
- Azkiya, A., Ichsan, M. B. P., Hasibuan, N. A. P. F., Andrianti, R., & Maulana, I. (2026): *Analisis sistem autentikasi dua faktor (2FA) dan efektivitasnya dalam meningkatkan keamanan akses*, JIKUM: Jurnal Ilmu Komputer, 2(2), 196-202. <https://doi.org/10.62671/jikum.v2i2.175>
- Haryanto, A. T. (2024): *APJII: Jumlah pengguna internet Indonesia tembus 221 juta orang*, detikInet. <https://apjii.or.id/berita/d/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang>
- Lumburovska, L., Dobrev, J., Andonov, S., Trpcheska, H. M., & Dimitrova, V. (2021): *A com-*

- parative analysis of HOTP and TOTP authentication algorithms: Which one to choose?*, Security & Future, 5(4), 131-136. <https://stumejournals.com/journals/confsec/2021/4/131>
- Maharani, A., Tahir, M., Buana, D. S., Fauziah, S. M., Putra, W. R., & Hasan, M. A. (2025): *Penerapan web proxy pada Mikrotik untuk optimalisasi keamanan jaringan wireless LAN di SMKN 1 Bangkalan*, JATI (Jurnal Mahasiswa Teknik Informatika), 9(4), 6029-6035. <https://doi.org/10.36040/jati.v9i4.13950>
- Mayorga, O. E. A., & Yoo, S. G. (2025): *One time password (OTP) solution for two factor authentication: A practical case study*, Journal of Computer Science, 21(5), 1099-1112. <https://doi.org/10.3844/jcssp.2025.1099.1112>
- M'Raihi, D., Bellare, M., Hoornaert, F., Naccache, D., & Ranen, O. (2005): *HOTP: An HMAC-based one-time password algorithm* (RFC 4226). <https://doi.org/10.17487/rfc4226>
- M'Raihi, D., Machani, S., Pei, M., & Rydell, J. (2011): *TOTP: Time-based one-time password algorithm* (RFC 6238), <https://doi.org/10.17487/rfc6238>
- Nurandani, D., Sitorus, S. P., Manoppo, A. R., Adrian, D., & Ayuda, M. I. (2025): *Penggunaan OTP (one time password) sebagai lapisan keamanan tambahan dalam aplikasi AY Pulsa*, Jurnal Minfo Polgan, 14(2), 3304-3314. <https://doi.org/10.33395/jmp.v14i2.15724>
- Nuryati, N., Lasambouw, C. M., Djabatika, D., Meilinda, L. L., Agoes, F., Sholahuddin, M. R., & Harika, M. (2022): *Two factor authentication, sistem inventarisasi barang dan manajemen dana bantuan operasional sekolah dinas pendidikan nasional*. Building of Informatics, Technology and Science (BITS), 4(2), 1129-1136. <https://doi.org/10.47065/bits.v4i2.2297>
- Pratama, A. P., Sugiyanta, L., & Idrus, A. (2023): *Design and implementation of FreeRADIUS as a user manager on the Mikrotik hotspot network at PT Indotruck Utama using the NDLC method*, International Journal of Information System & Technology, 7(2), 136-143.
- Rivera-Dourado, M., Gestal, M., Pazos, A., & Vázquez-Naya, J. (2024): *A novel protocol using captive portals for FIDO2 network authentication*, Applied Sciences, 14(9). <https://doi.org/10.3390/app14093610>
- Sukendar, T., Saputro, M. I., Ishaq, A., & Sumbaryadi, A. (2023): *Authentication wireless area network menggunakan captive portal berbasis Mikrotik pada Madrasah Ibtidaiyah Misbahul Athfal Bogor*, Jurnal Teknologi Informatika dan Komputer, 9(1), 243-254. <https://doi.org/10.37012/jtik.v9i1.1465>
- Widodo, S., Sutrisman, A., Amin, M. M., Harefa, M. F., Farhan, M. A., & Reinaldo, M. (2022): *Keamanan data user pada jaringan wireless menggunakan two factor, password dan MAC address filtering di Jurusan Teknik Komputer*, JUPITER: Jurnal Penelitian Ilmu dan Teknologi Komputer, 14(1), 37-44.

Halaman ini sengaja dikosongkan