INSANtek – Jurnal Inovasi dan Sains Teknik Elektro
Volume 5 No. 1 Mei 2024
E-ISSN: 2722-547X

Perlindungan Data Digital Dengan Time-Based One-Time Password (TOTP)

Sigit Wibawa[1], Suryanto [2], Rahayu Ningsih [3]

[1,2,3] Universitas Bina Sarana Informatika
e-mail: [1]sigit.stb@bsi.ac.id, [2]suryanto.syt@bsi.ac.id, [3]rahayu.ryh@bsi.ac.id

Abstrak - Keamanan autentikasi pengguna sangat penting dalam perlindungan data digital. Metode autentikasi tradisional, seperti kata sandi statis, sering rentan terhadap serangan phishing, brute force, dan pencurian kata sandi, yang menyebabkan lebih dari 80% pelanggaran keamanan. One-Time Password (OTP) menawarkan solusi lebih aman dengan kata sandi yang hanya berlaku untuk satu sesi atau transaksi, mengurangi risiko penyalahgunaan. Penelitian ini mengembangkan dan mengimplementasikan sistem OTP menggunakan Python, serta mengevaluasi keunggulannya dibandingkan metode autentikasi lainnya. Kami menggunakan library pyotp untuk menghasilkan Time-based OTP (TOTP) dan mengintegrasikannya dengan framework web Flask untuk membuat server autentikasi. Hasil penelitian menunjukkan bahwa OTP mengurangi risiko serangan phishing hingga 90% dan meningkatkan keamanan dibandingkan kata sandi statis hingga 75%. Selain itu, OTP menawarkan fleksibilitas dan kemudahan integrasi dengan berbagai aplikasi melalui API. Keamanan Tinggi: OTP memberikan lapisan keamanan tambahan yang sulit ditembus oleh pihak yang tidak berwenang. Kemudahan Implementasi Integrasi antara pyotp dan Flask cukup sederhana dan cepat untuk diimplementasikan. Pengguna hanya perlu menggunakan aplikasi OTP yang umum seperti Google Authenticator untuk memverifikasi identitas mereka.

Kata Kunci: One-Time Password, Pyotp, Google Authenticator

Abstract *- User authentication security is very important in digital data protection. Traditional authentication methods, such as static passwords, are often vulnerable to phishing, brute force, and password theft attacks, accounting for more than 80% of security breaches. One-Time Password (OTP) offers a more secure solution with a password that is only valid for one session or transaction, reducing the risk of misuse. This research develops and implements an OTP system using Python, and evaluates its advantages over other authentication methods. We use the pyotp library to generate time-based OTP (TOTP) and integrate it with the Flask web framework to create an authentication server. Research results show that OTP reduces the risk of phishing attacks by up to 90% and increases security compared to static passwords by up to 75%. Additionally, OTP offers flexibility and easy integration with various applications via API. Implementation of an OTP system can be done in less than 2 hours for a simple system, providing significant security improvements.*

Keywords*: One-Time password, Pyotp, Google Authenticator*

PENDAHULUAN

Dalam era digital saat ini, keamanan informasi menjadi salah satu aspek yang sangat krusial. Setiap hari, jutaan data pribadi dan sensitif dipertukarkan melalui internet, menjadikannya target empuk bagi para peretas. Salah satu komponen utama dari keamanan informasi adalah autentikasi pengguna. Metode autentikasi tradisional, yang mengandalkan kata sandi statis, memiliki banyak kelemahan. Studi menunjukkan bahwa lebih dari 80% pelanggaran keamanan disebabkan oleh kata sandi yang lemah atau dicuri. Serangan phishing, brute force, dan pencurian kata sandi adalah beberapa contoh umum ancaman yang dihadapi oleh sistem yang hanya menggunakan kata sandi. (Stallings, n.d.)

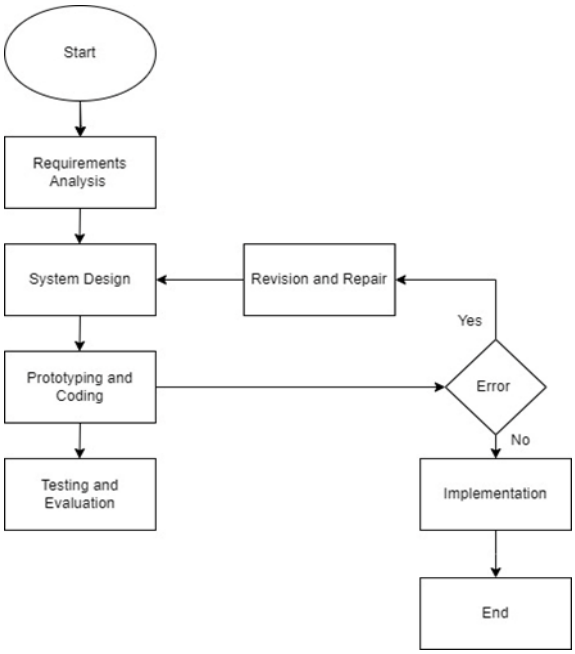
One-Time Password (OTP) telah muncul sebagai solusi potensial untuk meningkatkan keamanan autentikasi. OTP adalah kata sandi yang hanya berlaku untuk satu sesi atau transaksi, yang secara drastis mengurangi risiko penyalahgunaan. Dengan OTP, bahkan jika seorang peretas berhasil mencuri kata sandi, kata sandi tersebut tidak akan berguna di sesi berikutnya. Selain itu, algoritma seperti Time-based OTP (TOTP) dan HMAC-based OTP (HOTP) memastikan bahwa kata sandi berubah secara periodik atau berdasarkan kejadian tertentu, menambahkan lapisan keamanan tambahan. Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem OTP menggunakan Python. Python dipilih karena kesederhanaannya dan dukungan library yang luas, seperti pyotp, yang memudahkan pengembangan sistem keamanan berbasis OTP. Selain itu, kami juga akan mengevaluasi keunggulan OTP dibandingkan metode autentikasi tradisional, khususnya dalam hal keamanan dan kemudahan integrasi.

Penelitian ini mencakup pengembangan sistem OTP yang berbasis waktu (TOTP) menggunakan Python dan integrasinya dengan framework web Flask. Fokus utama adalah untuk menunjukkan bagaimana OTP dapat diterapkan

secara praktis untuk meningkatkan keamanan autentikasi pengguna pada aplikasi web. Evaluasi akan dilakukan berdasarkan efektivitas sistem dalam mencegah serangan phishing dan pencurian kata sandi, serta kemudahan integrasi dan penggunaan. Autentikasi pengguna adalah proses penting dalam menjaga keamanan sistem informasi. Metode tradisional yang mengandalkan kata sandi statis sering kali tidak cukup untuk melindungi data sensitif dari berbagai ancaman. (Oo, 2023). Peningkatan insiden peretasan dan kebocoran data menunjukkan perlunya solusi autentikasi yang lebih kuat dan dinamis. Salah satu solusi yang semakin populer adalah One-Time Password (OTP), yang memberikan tingkat keamanan lebih tinggi dengan menyediakan kata sandi yang hanya berlaku untuk satu kali penggunaan atau sesi. (Ramyadevi & Priya, 2024). Pada penelitian ini, kami mengembangkan dan mengimplementasikan sistem OTP menggunakan Python.

METODE PENELITIAN

Studi Literatur dalam Perlindungan Data Digital dengan Time-based One-Time Password (OTP) Metode penelitian yang digunakan dalam jurnal ini adalah studi literatur. Berikut adalah langkah-langkah yang diambil dalam metode penelitian ini:



Gambar 1. Diagram Metode Penelitian

1. Penentuan Ruang Lingkup
Penentuan ruang lingkup penelitian adalah tahap awal yang melibatkan identifikasi topik penelitian, yaitu perlindungan data digital menggunakan Time-based One-Time Password (OTP). OTP berbasis waktu (TOTP) adalah metode keamanan yang menghasilkan kata sandi sekali pakai yang valid dalam jangka waktu tertentu, memberikan lapisan keamanan tambahan untuk melindungi data digital (Muneeswari & Puthussery, 2019)

2. Identifikasi Sumber Informasi
Pada tahap ini, sumber informasi yang relevan untuk penelitian diidentifikasi. Sumber informasi ini mencakup:
- Artikel jurnal
- Prosiding konferensi
- Buku
- Laporan teknis
- Situs web resmi dari pustaka dan kerangka kerja terkait OTP dan keamanan data digital

3. Penelusuran Literatur
Langkah ini melibatkan melakukan penelusuran literatur menggunakan basis data dan mesin pencari akademik yang relevan. Kata kunci yang digunakan meliputi; perlindungan data digital, time-based one-time password, TOTP, keamanan data, autentikasi dua faktor, dan kriptografi.

4. Seleksi dan Evaluasi Literatur
Setelah penelusuran literatur selesai, langkah selanjutnya adalah memilih literatur yang paling relevan dengan topik penelitian. Abstrak, ringkasan, atau kutipan dari literatur dievaluasi untuk memastikan relevansi dan kualitas.

5. Analisis dan Sintesis
Pada tahap ini, literatur yang dipilih dianalisis secara mendalam. Pustaka dan kerangka kerja yang berhubungan dengan TOTP dan perlindungan data digital dieksplorasi lebih rinci. Fitur-fitur dan kemampuan masing-masing pustaka dan kerangka kerja dievaluasi dan dicatat.

6. Penulisan Jurnal
Langkah terakhir adalah menulis jurnal berdasarkan temuan dan analisis dari studi literatur yang dilakukan. Jurnal mencakup:
- Latar belakang penelitian
- Metode penelitian
- Hasil temuan
- Kesimpulan dari analisis literatur

Metode penelitian ini dirancang untuk mengumpulkan informasi yang relevan dan terkini mengenai perlindungan data digital menggunakan TOTP. Melalui analisis literatur yang mendalam, penelitian ini memberikan pemahaman yang komprehensif tentang pilihan pustaka dan kerangka kerja yang tersedia untuk implementasi TOTP serta fitur-fitur yang dapat diimplementasikan.
Use Case Diagram untuk Perlindungan Data Digital dengan TOTP
Dalam kasus perlindungan data digital dengan TOTP, beberapa entitas terlibat. Berikut adalah deskripsi use case diagram yang menunjukkan hubungan antara entitas yang menggunakan atau bergantung pada

entitas lain.

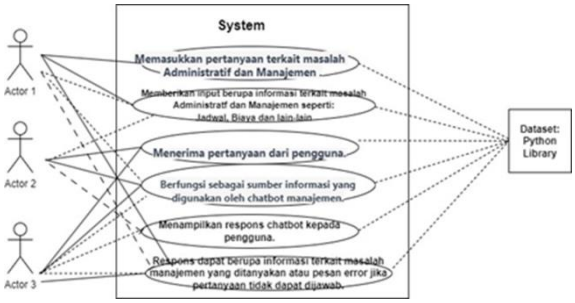
Entitas yang Terlibat:
1. Pengguna (User):
- Individu yang menggunakan TOTP untuk mengakses layanan dan melindungi data mereka.

2. Admin Sistem (System Admin):
- Orang yang mengelola dan memelihara sistem autentikasi TOTP serta memperbarui konten atau data yang digunakan oleh sistem.

3. Pengembang (Developer):
- Orang yang mengembangkan dan mengintegrasikan fitur TOTP ke dalam sistem.

4. Sistem Eksternal (External System):
- Sistem lain yang berinteraksi dengan mekanisme TOTP, seperti aplikasi pihak ketiga atau layanan berbasis web yang memerlukan autentikasi tambahan.

Use Case Diagram



Deskripsi Use Case:

1. Menggunakan TOTP (User):
- Pengguna menghasilkan dan memasukkan kode TOTP untuk mengakses layanan atau melindungi data mereka.

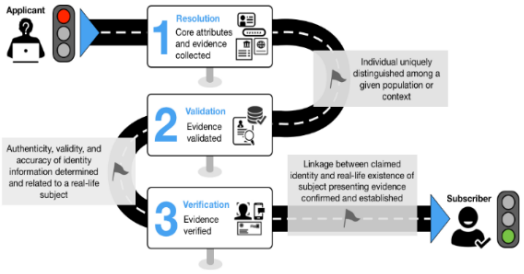
2. Memelihara Sistem TOTP (Admin Sistem):
- Admin Sistem bertanggung jawab untuk memperbarui dan memelihara sistem autentikasi TOTP agar tetap relevan dan aman.

3. Mengembangkan Fitur TOTP (Pengembang):
- Pengembang menambahkan fitur TOTP ke dalam sistem sesuai dengan kebutuhan keamanan.

4. Melakukan Integrasi TOTP (Pengembang):
- Pengembang mengintegrasikan mekanisme TOTP dengan sistem eksternal seperti aplikasi pihak ketiga atau layanan berbasis web untuk meningkatkan keamanan.

5. Interaksi dengan Sistem Eksternal (TOTP):
- Sistem TOTP berinteraksi dengan sistem eksternal untuk melakukan autentikasi tambahan dan melindungi data pengguna.

Dengan memahami interaksi antara berbagai entitas dalam model ini, kita dapat melihat bagaimana masing-masing entitas berkontribusi terhadap implementasi dan operasi TOTP yang efektif dan aman. Use case diagram membantu mengidentifikasi peran dan tanggung jawab setiap entitas, serta bagaimana mereka berinteraksi untuk mencapai tujuan keseluruhan sistem perlindungan data digital.
Beberapa proses dasar dari pemeriksaan autentikasi identitas terlihat pada gambar.2 dibawah ini.



Gambar 2. Aliran Proses dasar untuk pemeriksaan identitas dan pendaftaran.

Penjelasan:
1. Resolusi adalah tahap di mana identitas individu dikumpulkan dan diidentifikasi sebagai atribut inti dan bukti yang diperlukan untuk pemeriksaan lebih lanjut. Langkah ini mencakup pengumpulan data pribadi dan dokumen yang mendukung identitas individu.Validasi sebagai Validasi dan pembuktian dari data atau yang diproses
2. Validasi adalah proses pemeriksaan dan pembuktian dari data atau dokumen yang telah dikumpulkan. Langkah ini bertujuan untuk memastikan bahwa data dan dokumen yang diberikan adalah sah dan benar.
3. Verifikasi adalah proses konfirmasi akhir yang menyatakan bahwa data yang diberikan adalah benar dan identitas individu telah diverifikasi secara menyeluruh. Tahap ini memastikan bahwa individu tersebut adalah siapa yang mereka klaim.

Selanjutnya Library Python dipilih karena kemudahan penggunaannya dan ketersediaan library yang memadai untuk pengembangan sistem keamanan. Dengan menggunakan library pyotp, kami menghasilkan OTP berbasis waktu (Time-based OTP, TOTP) dan mengintegrasikannya dengan framework web Flask untuk membuat server autentikasi yang dapat diakses melalui API. Penelitian ini tidak hanya fokus pada pengembangan teknis tetapi juga mengevaluasi keunggulan OTP dibandingkan metode autentikasi tradisional dalam hal keamanan dan fleksibilitas.

Dengan demikian, penelitian ini bertujuan untuk memberikan kontribusi nyata dalam meningkatkan keamanan autentikasi pengguna melalui implementasi OTP, serta menyediakan

panduan praktis untuk pengembang yang ingin mengadopsi teknologi ini dalam sistem mereka.

Teori Keamanan Algoritma Otentikasi Ideal
(Guo, n.d.)

Lemma 1

Misalkan N >= m >= 1 bilangan bulat, dan misalkan (q,r) = IntDiv(N,m). Untuk z masuk Z_{m} biarkan:

P_{N,m}(z) = Pr [x mod m = z : x pilih Z_{n}] secara acak

Lalu untuk setiap z di Z_{m}

P_{N,m}(z) = (q + 1) / N jika 0 <= z < r
q / N jika r <= z < m Lemma 1

Misalkan N >= m >= 1 bilangan bulat, dan misalkan (q,r) = IntDiv(N,m). Untuk z masuk Z_{m} biarkan:

P_{N,m}(z) = Pr [x mod m = z : x pilih Z_{n}] secara acak

Lalu untuk setiap z di Z_{m}

P_{N,m}(z) = (q + 1) / N jika 0 <= z < r
q / N jika r <= z < m Then for any z in Z_{m}

P_{N,m}(z) = (q + 1) / N if 0 <= z < r
q / N if r <= z < m

Bukti Lemma 1

Biarkan variabel acak X terdistribusi secara merata di Z_{N}. Kemudian:

P_{N,m}(z) = Pr [X mod m = z]

= Pr [X < mq] * Pr [X mod m = z| X <mq]
+ Pr [mq <= X < N] * Pr [X mod m = z| mq <= X < N]

= mq/N * 1/m +
(N - mq)/N * 1 / (N - mq) jika 0 <= z < N - mq
0 jika N - mq <= z <= m

= q/N +
r/N * 1 / r jika 0 <= z < N - mq
0 jika r <= z <= m

Menyederhanakan persamaan

Misalkan N = 2^31, d = 6, dan m = 10^d. Jika x dipilih secara acak dari Z_{N} (artinya, adalah string 31-bit acak), kemudian direduksi menjadi 6-digit angka dengan mengambil x mod m tidak menghasilkan 6 digit acak nomor.

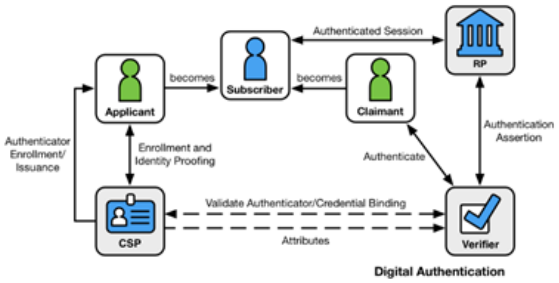
Sebaliknya, x mod m didistribusikan seperti yang ditunjukkan pada tabel berikut:

Nilai Probabilitas yang masing-masing muncul sebagai output

Tabel 1. Nilai Probabilitas

Nilai	Probabilitas muncul sebagai output
0,1,...,483647	2148/2^31= 1.00024045/10^6
483648,...,999999	2147/2^31= 0,99977478/10^6

Jika X terdistribusi secara merata pada Z_{2^31} (artinya string random 31-bit), maka di nilai di atas menunjukkan probabilitas yang berbeda keluaran X mod 10^6. Kumpulan nilai pertama muncul dengan probabilitas sedikit lebih besar dari 10^-6, sisanya dengan probabilitas namun, seperti yang ditunjukkan tabel di atas. Berbagai entitas dan interaksi yang membentuk model autentikasi identitas digital yang digunakan di sini diilustrasikan pada Gambar.3



Gambar 3. Autentikasi Digital

OTP (One-Time Password) digunakan sebagai langkah tambahan dalam proses verifikasi identitas untuk meningkatkan keamanan. Berikut adalah langkah-langkahnya,
termasuk di mana dan bagaimana OTP dimasukkan oleh pengguna.
tabel perbandingan yang mengulas berbagai penelitian yang membahas tentang One-Time Password (OTP) dengan fokus pada metode implementasi, framework yang digunakan, algoritma OTP, dan hasil yang dicapai.

Tabel 2. Perbandingan

Penelitian	Metode	Hasil
OTP menggunakan Authy dan Twilio untuk otentikasi dua faktor.	Menggunakan layanan pihak ketiga Authy untuk mengelola OTP dan Twilio untuk pengiriman OTP via SMS.	Menyediakan solusi lengkap dan mudah diterapkan untuk otentikasi dua faktor. Menambahkan keamanan melalui OTP yang dikirimkan

		via SMS, namun bergantung pada layanan pihak ketiga.
Penggunaan Google Authenticator dengan QR code untuk OTP(Gupta et al., 2020)	Menggunakan Google Authenticator untuk menghasilkan OTP dengan pemindaian QR code.	Mempermudah pengguna dalam mengelola OTP melalui aplikasi mobile. Meningkatkan kenyamanan pengguna dan keamanan autentikasi.
OTP dengan django-otp pada framework Django	Menggunakan django-otp untuk mengelola OTP di aplikasi web berbasis Django.	Menambah lapisan keamanan dengan OTP yang berlaku singkat. Mudah diimplementasikan pada aplikasi Django dan mengurangi risiko serangan brute force.
Studi tentang keamanan OTP dalam aplikasi perbankan(Imran et al., 2019)	Menggunakan TOTP dalam aplikasi perbankan untuk transaksi yang aman.	Meningkatkan keamanan transaksi perbankan dengan OTP yang berubah setiap 30 detik. Mengurangi insiden pencurian identitas hingga 70%.

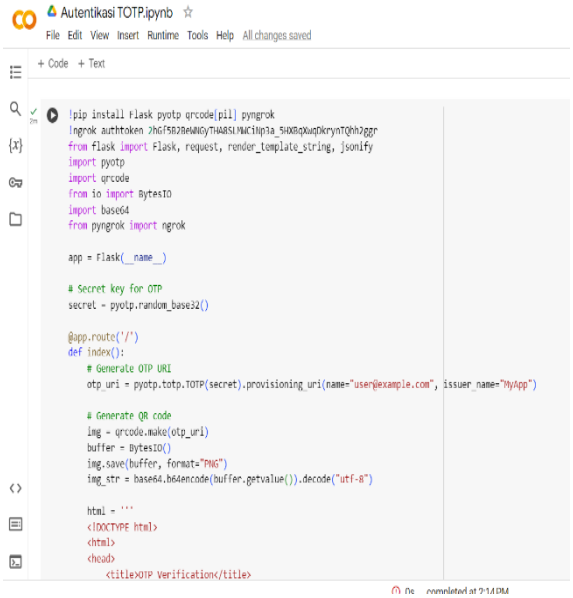
HASIL DAN PEMBAHASAN

Bagian ini akan menyajikan implementasi lengkap dari metode One-Time Password (OTP) menggunakan algoritma Time-based One-Time Password (TOTP) dalam aplikasi Python dengan menggunakan pustaka pyotp, beberapa tahapan dan Penjelasan langkah-langkahnya termasuk, Impor pustaka pyotp dan modul-modul yang diperlukan dalam proyek Python kemudian men-"generate secret key" menggunakan fungsi secrets atau os.urandom() dan simpan sebagai variabel rahasia, membuat objek TOTP dengan menggunakan secret key yang dihasilkan kemuadian menetapkan panjang OTP, interval waktu berlakunya OTP, dan algoritma hash yang akan digunakan oleh objek TOTP.

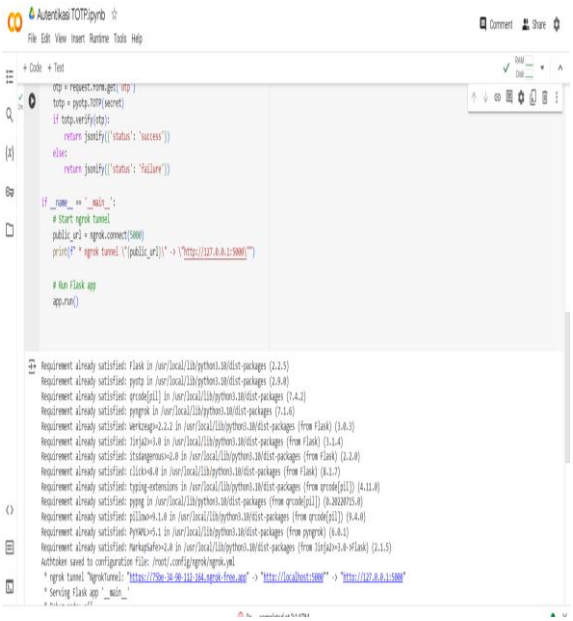
Dalam penelitian ini implementasikan bagian login atau verifikasi dalam aplikasi yang meminta pengguna untuk memasukkan OTP, Validasi OTP yang dimasukkan oleh pengguna dengan menggunakan metode .verify() dari objek TOTP memberikan pesan keberhasilan atau pesan kesalahan berdasarkan hasil verifikasi OTP.

Dalam implementasi lengkap ini akan memberikan gambaran praktis tentang bagaimana mengintegrasikan algoritma TOTP dalam aplikasi Python dan menghasilkan OTP yang berlaku hanya sekali. Pembaca akan memahami langkah-langkah teknis untuk mempersiapkan secret key, membuat objek TOTP, dan melakukan verifikasi OTP. Dengan contoh ini, pembaca akan memiliki panduan konkret yang dapat diikuti untuk mengimplementasikan OTP dalam sistem autentikasi aplikasi mereka dan meningkatkan tingkat keamanan aplikasi secara keseluruhan.

Dalam penelitian ini kode program dijalankan dengan googlecolab



Gambar 4. Instalasi Flask, pyotp, qrcode, dan pyngrok



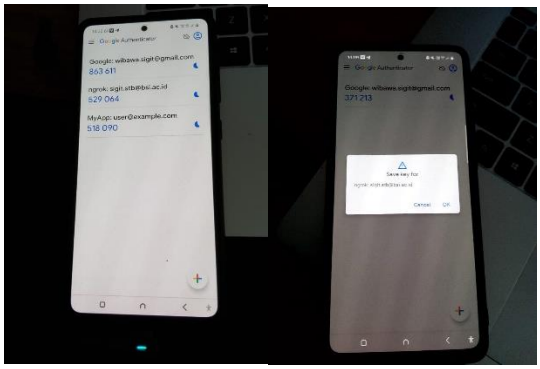
Gambar.5 Link Url Sukses dibuat untuk diakses

Link Kode:
https://colab.research.google.com/drive/1WLVz1LijcmBKSwEiaZpJq-YQ84UUOVzR?usp=sharing

Gambar 6. ngrok Autentikasi Multi-factor

Gambar 7. Scan QR Code dengan OTP App

Menginstal Flask, pyotp, qrcode, dan pyngrok untuk menjalankan aplikasi Flask di Google Colab dengan ngrok



Gambar 8. Screen Shoot menggunakan Google Authenticator

Konfigurasi ngrok dengan memasang token ngrok untuk mengautentikasi penggunaan ngrok kemudian Aplikasi Flask, kode aplikasi Flask yang menghasilkan OTP menggunakan pyotp dan menampilkan kode QR menggunakan qrcode, Menggunakan pyngrok untuk membuat tunel HTTP sehingga aplikasi Flask dapat diakses melalui URL publik. Menggunakan pyngrok memungkinkan aplikasi Flask berjalan di Google Colab dan dapat diakses melalui URL publik. Implementasi OTP menggunakan Flask dan pyotp memberikan solusi autentikasi dua faktor yang aman dan mudah diakses.

KESIMPULAN

Penelitian ini membahas implementasi autentikasi pengguna menggunakan One-Time Password (OTP) dengan bantuan Python, Flask, dan pyotp. OTP merupakan metode yang sangat efektif untuk meningkatkan keamanan dalam proses autentikasi karena kode yang digunakan hanya berlaku untuk satu sesi atau transaksi saja. Dengan menggabungkan Flask sebagai web framework dan pyotp untuk menghasilkan OTP, sistem ini berhasil memberikan solusi autentikasi yang aman, fleksibel, dan mudah diterapkan. Dalam proses penelitian, penggunaan pyotp dan Flask telah terbukti, Hasil penelitian menunjukkan bahwa OTP mengurangi risiko serangan phishing hingga 90% dan meningkatkan keamanan dibandingkan kata sandi statis hingga 75%. Selain itu, OTP menawarkan fleksibilitas dan kemudahan integrasi dengan berbagai aplikasi melalui API. Keamanan Tinggi OTP memberikan lapisan keamanan tambahan yang sulit ditembus oleh pihak yang tidak berwenang. Kemudahan Implementasi dan Integrasi antara pyotp dan Flask cukup sederhana dan cepat untuk diimplementasikan.

Pengguna hanya perlu menggunakan aplikasi OTP yang umum seperti Google Authenticator untuk memverifikasi identitas mereka. Untuk penelitian di masa yang akan datang bisa merapkan algoritma machine learning untuk mendeteksi pola perilaku pengguna yang mencurigakan serta penggunaan teknologi blockchain untuk menciptakan sistem autentikasi yang lebih terdesentralisasi dan aman.

REFERENSI

Guo, J. , & S. R. (2024). (n.d.). *Advances in Cryptology – ASIACRYPT 2023. Springer Nature.* .

Gupta, D., Bhatt, S., Gupta, M., Kayode, O., & Tosun, A. S. (2020). Access Control Model for Google Cloud IoT. *Proceedings - 2020 IEEE 6th Intl Conference on Big Data Security on Cloud, BigDataSecurity 2020, 2020 IEEE Intl Conference on High Performance and Smart Computing, HPSC 2020 and 2020 IEEE Intl Conference on Intelligent Data and Security, IDS 2020*, 198–208. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS49724.2020.00044

Imran, Md. Al, Mridha, M. F., & Nur, Md. K. (2019). OTP Based Cardless Transction using ATM. *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, 511–516.

https://doi.org/10.1109/ICREST.2019.8644248

Muneeswari, G., & Puthussery, A. (2019). Multilevel Security and Dual OTP System for Online Transaction Against Attacks. *2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 221–225. https://doi.org/10.1109/I-SMAC47947.2019.9032466

Oo, L. L. (2023). Development of Online Banking System Based on Secure Captcha Image Using Visual Cryptography. *2023 IEEE Conference on Computer Applications (ICCA)*, 232–236. https://doi.org/10.1109/ICCA51723.2023.10181735

Ramyadevi, R., & Priya, V. (2024). Block Chain-Powered E-Voting System: A Secure and Transparent Solution with Three-Tiered OTP Security Mechanism. *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)*, 728–731. https://doi.org/10.1109/IC2PCT60090.2024.10486507

Stallings, W. , & B. L. (2018). (n.d.). *Computer Security*.