ISSN : 2962 – 5688
Website : jurnal.umj.ac.id/index.php/TripleA
Email : triplea@umj.ac.id

TRIPLE A
Jurnal Pendidikan Teknologi Informasi

PERANCANGAN SISTEM OTENTIKASI BERBASIS *ONE TIME PASSWORSD (OTP)* DENGAN ALGORITMA *RSA* SEBAGAI METODE AUTENTIKASI: IMPLEMENTASI MENGGUNAKAN BAHASA PEMROGRAMAN *PHYTON*

Ivan Harry Cahyadi, Muhammad Akbar Hidayatullah, Sofia Nabila Ramdan

Politeknik Negeri Jakarta, Fakultas Teknik Informatika dan Komputer, Program Studi Teknik Multimedia dan Jaringan, Universitas Indonesia, Jl. Prof. DR. G.A. Siwabessy, Kukusan, Kecamatan Beji, Kota Depok, Jawa Barat 16425

ivan.harrycahyadi.tik20@mhsw.pnj.ac.id

muhammad.akbarhidayatullah.tik20@mhsw.pnj.ac.id

sofia.nabilaramdan.tik20@mhsw.pnj.ac.id

Abstract

Information security is an important aspect in the current digital era. Reliable authentication methods are needed to protect against unauthorized access to the system and ensure the confidentiality of user information. This research aims to design an authentication system that utilizes One-Time Password (OTP) as the authentication method, using the RSA algorithm in the Python programming language. The authentication system consists of an authentication server and a client, where the server generates a unique OTP that is valid only once, and the client encrypts the OTP using the RSA public key before sending it back to the server for verification. The advantages of this system include high security due to the robust RSA algorithm, and the one-time use feature of OTP to reduce the risk of replay attacks. The implementation of this research results in a reliable authentication system that protects the system and user information from unauthorized access, implemented using the Python programming language.

Keywords: *One-Time-Password (OTP), Algoritma RSA, Python*

Abstrak

Keamanan sistem informasi menjadi aspek penting dalam era digital saat ini. Metode otentikasi yang handal diperlukan untuk melindungi akses yang tidak sah ke sistem dan menjaga kerahaiaan informasi pengguna. Penelitian ini bertujuan unutk merancang sebuah otentikasi yang menggunakan One-Time-Password (OTP) sebagai metode autentikasi dengan menggunakan algoritma RSA dalam Bahasa pemograman Phyton. Sistem otentikasi ini terdiri dari server otentikasi dan klien, dimana server otentikasi akan menghasilkan OTP yang uni dan hanya berlaku sekali, sedangkan klien akan mengenkripsi OTP menggunakan kunci publik

RSA sebelum mengirimkannya kembali ke server untuk diverifikasi. Keuntungan dari sistem ini adalah tingkat keamanan yang tinggi berkat penggunaak algoritma RSA yang terbukti kuat, serta fitur OTP yang hanya berlaku sekali untuk mengurangi risiko serangan replay. Implementasi penelitian ini menghasilkan sebuah sistem otentikasi yang handal dalam melindungi sistem dan informasi pengguna dari akses yang tidak sah menggunakan Bahasa pemograman Phyton.

Keywords: *One-Time-Password (OTP), Algoritma RSA, Phyton*

PENDAHULUAN

Keamanan sistem informasi merupakan aspek yang krusial dalam era digital saat ini, dimana perlindungan terhadap akses yang tidak sah dan kerahasiaan infomasi penggina menjadi priotas utama. Metode autentikasi yang andal dan aman diperlukan untuk memasitikan bahwa hanya pengguna yang sah dapat mengakses sistem dan informasi terkaut. Salah satu metode autentikasi yang efektif dan sering digunakan adalah One Time Password (*OTP*), dimana pengguna menerima kode unik yang hanya berlaku sekali untuk setiap sesi otentikasi.

Algirtma *RSA*, berdasarkan kunci publik, telah terbukti menjadi metode yang kuat dan aman dalam mengamankan komunikasi dan melindungi data sensitif. Dalam algoritma ini, pasangan kunci publik dan kunci privat digunakan untuk mengenkripsi dan mendekripsi pesan. Dengan menggunakan algoritma *RSA* sebagai metode autentikasi, keamanan sistem otentikasi dapat ditingkatkan, mengurangi risiko serangan kriptografis yang umumnya dilakukan oleh penyerang.

Penelitian ini bertujuan untuk merancang sebuah sistem otentikasi berbasis One Time Password (*OTP*) dengan menggunakan algoritma *RSA* sebagai metode autentikasi, dengan implementasi menggunakan Bahasa pemrograman *Phyton*. Dalam sistem otentikasi ini, pengguna akan menerima *OTP* yang dihasilkan secara unik dan hanya berlaku sekali untuk setiap sesi otentikasi. *OTP* akan dienkripsi menggunakan algoritma *RSA* dengan kunci publik sebelum dikirim kembali ke server untuk diverifikasi.

Pada pendahuluan ini, kami akan menjelaskan latar belakang pentingnya keamanan sistem informasi dan metode autentikasi yang handal. Kami juga akan menguraikan konsep dasar *One Time Password* (*OTP*) dan algoritma *RSA* sebagai metode autentikasi yang efektif. Selain itu, kami akan memaparkan pentingnya penggunaan bahasa pemrograman *Python* dalam implementasi sistem otentikasi ini, yang memberikan fleksibilitas dan kemudahan dalam pengembangan aplikasi.

One Time Password (OTP)

One Time Password (OTP) merupakan sebuah kode yang digunakan hanya sekali untuk otentiasi pengguna dalam sebuah sesi. Kode ini dihasilkan secara unik dan tidak dapat digunakan Kembali, sehingga memberikan tingkat keamanan yang lebih tinggi dalam melawan serangan seperti serangan replay.

Algoritma *RSA*

Algoritma *RSA* merupakan salah satu metode kriptografi kunci publik yang sangat kuat. Algortima ini melibatkan pasangan kunci, yaitu kunci publik dan kunci privat.

Kunci publik digunakan untuk mengenkripsi pesan, sedanglan kunci pivat digunakan untuk mendekripsi pesan yang telah dienkripsi menggunakan kunci publik. Dengan menggunakan algoritma *RSA* sebagai metode autentikasi, sistem otentikasi dapat memastikan keamanan komunikasi dan melindungi informasi pengguna dengan efektif.

Phyton

Phyton merupakan bahasa pemrograman yang popular dan banyak digunaan dalam pengembangan aplikasi. Kelebihan *Python* antara lain flesibilitas dan kemudahan dalam penulisan kode.

METODE PENELITIAN

One-Time Password (OTP) adalah sebuah password yang hanya dapat digunakan untuk satu kali login atau transaksi. *OTP* digunakan sebagai lapisan tambahan keamanan untuk melindungi akses yang membutuhkan autentikasi, seperti login ke akun online atau melakukan transaksi keuangan.

Pada *Python*, terdapat beberapa library yang dapat digunakan untuk menghasilkan *OTP*, salah satunya adalah *PyOTP*. *PyOTP* adalah library *Python* yang menyediakan fungsionalitas untuk menghasilkan *OTP* berdasarkan algoritma *Time-Based One-Time Password (TOTP)* dan *HMAC-Based One-Time Password (HOTP).*

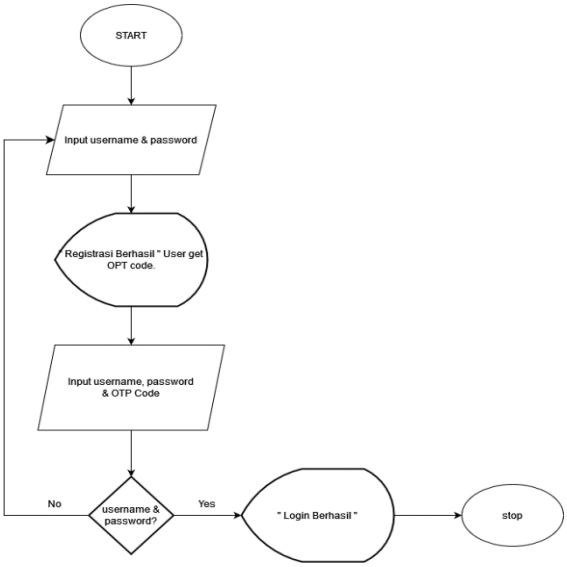
Dengan menggunakan *PyOTP*, kita dapat mengintegrasikan fungsi *OTP* ke dalam aplikasi *Python* untuk memberikan lapisan tambahan keamanan dalam proses autentikasi.

HASIL DAN PEMBAHASAN

Analisis Kebutuhan

Pada tahap ini akan dilakukan analisis terhadap kebutuhan rancang bangun sistem *autentikasi berbasis OTP*. Analisis ini meliputi analisis kebutuhan alat dan bahan yang dibutuhkan dalam pembuatan sistem, diantaranya yaitu, Laptop/PC, Aplikasi *Visual studio code, Python3, Library Pyotp* dan *Json.*

Flowchart



Pengujian Program

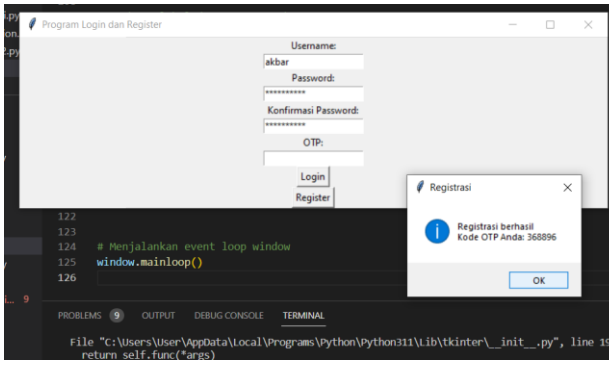
Penulis mengujikan program dengan menggunakan aplikasi visual studio code sebagai software editornya untuk menjalankan aplikasi python. Penulis juga menggunakan Bahasa python3 pada program serta *library pyotp* dan *json*. Cara kerja sistem *One-Time-Password* pada program ini adalah pengguna dapat memasukan *username, password* dan kode *OTP* yang diperlukan untuk login dan register.

Saat pengguna melakukan register sistem akan memeriksa apakah password yang dimasukan sudah benar dan cocok dengan konfirmasi password. Jika cocok, sistem akan menghasilkan sebuah kode *OTP Secret* yang digunakan untuk men-generate OTP.

Data pengguna (*username*, *password dan OTP Secret*) disimpan ke dalam sebuah file *JSON*.

Saat pengguna melakukan login, sistem akan memeriksa apakah data pengguna yang dimasukkan (*username*, *password*, *dan OTP*) sesuai dengan data pengguna yang tersimpan dalam *file JSON*. Sistem akan memverifikasi *OTP* yang dimasukkan menggunakan *OTP* Secret yang terkait dengan username pengguna. Jika data pengguna valid, pesan "Login berhasil" akan ditampilkan. Jika tidak valid, pesan "Login gagal" akan ditampilkan.

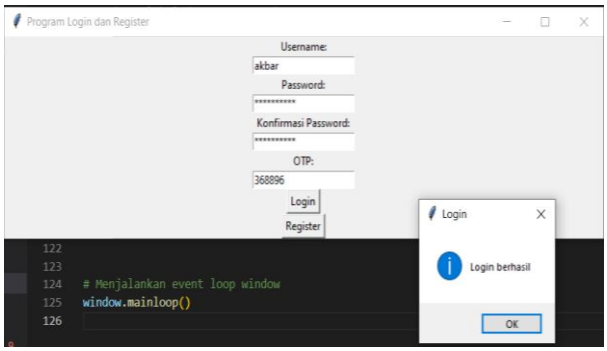
Hasil pengujian program register user:



Dari hasil gambar tersebut dapat dilihat bahwa user memasukan nama dan password serta mengkonfirmasi password untuk mendapatkan kode *otp* dari sistem. Jika proses registrasi berhasil program akan mengeluarkan kode *otp* untuk user bisa login. Saat pengujian program register, terlihat *one-time password* yang dibangkitkan adalah 368896.

Password ini dihasilkkan Saat pengguna melakukan registrasi, fungsi `register()` dipanggil. Di dalam fungsi `register()`, OTP Secret yang merupakan sebuah string acak dihasilkan menggunakan `pyotp.random_base32()`. Setelah mendapatkan *OTP Secret*, data pengguna (*username*, *password*, *dan OTP Secret*) disimpan ke dalam *file JSON* menggunakan fungsi `save_user_data()`. Kemudian, *OTP* Secret digunakan untuk membuat objek `pyotp.TOTP(otp_secret)`. Objek *TOTP* tersebut digunakan untuk memperoleh kode *OTP* saat ini menggunakan `otp.now()`. Kode *OTP* yang dihasilkan ditampilkan dalam pesan dialog menggunakan fungsi `show_message()`.

Hasil pengujian login pada user



Dari gambar diatas terlihat user memasukan username dan password serta mengkonfirmasi password. Untuk berhasil masuk user harus memasukan kode *OTP* yang diberikan pada tahap registrasi. kode *OTP* yang dimasukkan oleh user dibaca dari *input field* dengan menggunakan otp_entry.get() pada fungsi login(). Saat tombol "Login" ditekan, fungsi `login()` dipanggil. Di dalam fungsi `login()`, kode *OTP* yang dimasukkan oleh pengguna diambil menggunakan `otp_entry.get()`. Kode *OTP* ini kemudian disimpan dalam variabel `otp_code`. Kemudian terdapat Fungsi `check_user_data()` dipanggil dengan menyediakan *username*, *password*, dan kode *OTP* yang dimasukkan oleh pengguna. Di dalam fungsi `check_user_data()`, data pengguna dibaca dari file JSON menggunakan fungsi `json.load(file)`.

Data pengguna yang cocok dengan username yang dimasukkan oleh pengguna diambil.Jika data pengguna ditemukan dan username, password, dan kode *OTP* yang dimasukkan oleh pengguna cocok, maka fungsi `verify_otp()` dipanggil untuk memverifikasi kode *OTP*, jika kode *OTP* terverifikasi dengan benar, pengguna diberikan pesan dialog "Login berhasil" menggunakan fungsi `show_message()`. Namun Jika kode *OTP* tidak terverifikasi, pengguna diberikan pesan dialog "Login gagal" menggunakan fungsi `show_message()`.

SIMPULAN

Sistem otentikasi berbasis *One Time Password (OTP)* merupakan metode keamanan yang melibatkan penggunaan kata sandi sekali pakai yang berlaku hanya untuk satu sesi otentikasi. Algoritma *RSA (Rivest-Shamir-Adleman)* merupakan algoritma kriptografi asimetris yang sering digunakan untuk keperluan enkripsi, dekripsi, dan tanda tangan digital.

Dalam konteks diatas, perancangan sistem otentikasi berbasis *OTP* menggunakan algoritma *RSA* sebagai metode autentikasi mungkin melibatkan beberapa langkah dasar berikut:

1. Pembangkitan kunci *RSA*: Sistem akan menghasilkan sepasang kunci *RSA*, yaitu kunci publik dan kunci pribadi. Kunci publik digunakan untuk mengenkripsi *OTP*, sedangkan kunci pribadi digunakan untuk mendekripsi *OTP*.
2. Pembangkitan *OTP*: Setiap kali pengguna ingin melakukan otentikasi, sistem akan menghasilkan *OTP* secara acak.
3. Enkripsi *OTP*: *OTP* yang dihasilkan akan dienkripsi menggunakan kunci publik *RSA*. Hal ini memastikan bahwa hanya pemilik kunci pribadi yang dapat mendekripsi *OTP* tersebut.
4. Pengiriman *OTP*: *OTP* yang telah dienkripsi akan dikirimkan kepada pengguna melalui saluran komunikasi yang aman, misalnya melalui pesan teks atau aplikasi otentikasi.
5. Dekripsi *OTP*: Pengguna menerima *OTP* yang dienkripsi dan menggunakan kunci pribadi *RSA* untuk mendekripsinya.
6. Verifikasi *OTP*: Sistem akan membandingkan *OTP* yang didekripsi dengan *OTP* yang asli yang dihasilkan oleh sistem. Jika kedua *OTP* cocok, pengguna dianggap berhasil melakukan otentikasi.

Keuntungan dari menggunakan *OTP* dengan algoritma *RSA* sebagai metode autentikasi termasuk keamanan yang tinggi karena penggunaan *OTP* sekali pakai dan kekuatan enkripsi yang dihasilkan oleh algoritma *RSA*.

DAFTAR PUSTAKA

Calvin Christian, Sampe Hotlan Sitorus, Irma Nirmala. "IMPLEMENTASI ALGORITMA RSA DAN ONE TIME PASSWORD (OTP) UNTUK PENGAMANAN DATA PENGGUNA DAN PROSES TRANSAKSI PADA WEBSITE E-COMMERCE". Coding : Jurnal Komputer dan Aplikasi. Volume 11, No. 01 (2023), hal 62-72.

Ding, X., Zhang, W., & Li, Y. (2012). Research on OTP algorithm based on RSA in the secure authentication

system. In 2012 International Conference on Computer Science and Service System (pp. 2216-2219). IEEE.

Dwi Mulyanto, R., Ruyani, A., & Saifudin, Z. (2019). Implementation of One Time Password Algorithm (OTP) Using RSA in Building Web-based Authentication System. International Journal of Information Technology and Electrical Engineering, 8(6), 399-404.

Kabir, R., Gondal, I., & Ahmed, S. (2019). An Improved RSA Based One Time Password Authentication Protocol. In 2019 12th International Conference on Developments in eSystems Engineering (DeSE) (pp. 123-128). IEEE.

Meng, W., Zhang, H., & Lu, L. (2017). Secure OTP algorithm based on RSA in information system. Journal of Physics: Conference Series, 840(1), 012086.

Nani Sarah Hapsari, Yenni Fatman, Isbandi. "Implementasi Metode One Time Password pada Sistem Pemesanan Online". JURNAL MEDIA INFORMATIKA BUDIDARMA. Volume 4, Nomor 4, Oktober 2020, Page 930-939.

Priyanka Nema. (2011). An Innovative Approach for Dynamic Authentication in Public Cloud: Using RSA, Improved OTP and MD5. International Journal of Innovative Research in Computer and Communication Engineering. ISSN(Online): 2320-9801.

Rizki, Sri Mulyati. "Implementasi One Time Password Menggunakan Algoritma SHA-512 Pada Aplikasi Penagihan Hutang PT. XHT". Edumatic: Jurnal Pendidikan Informatika Vol. 4 No. 1, Juni, 2020.

Singh, P., & Joshi, R. C. (2013). A secure one-time password authentication scheme using RSA. In 2013 International Conference on Communication Systems and Network Technologies (pp. 237-241). IEEE.

Yhopi Suhelna. "Perancangan Aplikasi Penyandian Pesan Teks dengan Menggunakan Algoritma Digraph Cipher". JUKI : Jurnal Komputer Dan Informatika, 2(1), 25–34. 2020.