SECURITY SERVICE DENGAN DIGITAL SIGNATURE, ENKRIPSI, DAN SECURE SESSION

Disusun Oleh:

Nagatan Alief Putra Silahen	24031554086
Muslim Fazlur Rohman	24031554154
Muhammad Ramadhan Albaary Putra	24031554161
Kafka Praya Firmansyah	24031554182

Dosen Pengampu Mata Kuliah:

Moh. Khoridatul Huda, S.Pd., M.Si., Ph.D.

PROJECT AKHIR KEAMANAN DAN INTEGRITAS DATA

PROGRAM STUDI SAINS DATA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SURABAYA

2025

BAB I

PENDAHULUAN

A. Latar Belakang

Pada tugas ini, kelompok mengimplementasikan sebuah layanan API berbasis FastAPI yang berfungsi sebagai *trusted authority server* untuk sistem Punk Records. Layanan ini dirancang untuk menangani aspek keamanan komunikasi digital dengan menyediakan mekanisme penyimpanan *public key* pengguna serta verifikasi *digital signature* pada pesan yang dipertukarkan. Penerapan kriptografi kunci publik memungkinkan sistem memastikan bahwa pesan benar-benar berasal dari pengirim yang sah dan tidak mengalami perubahan selama proses transmisi. Pendekatan ini merupakan konsep fundamental dalam sistem keamanan modern dan banyak digunakan untuk menjamin autentikasi serta integritas data pada layanan berbasis jaringan (Diffie & Hellman, 1976; Rivest et al., 1978).

Selain itu, sistem yang dikembangkan oleh kelompok ini mendukung relay pesan aman dengan menerapkan enkripsi simetris menggunakan Advanced Encryption Standard (AES) untuk menjaga kerahasiaan komunikasi antar pengguna. Pengelolaan akses ke layanan dilakukan melalui mekanisme *secure session* berbasis token, sehingga hanya pengguna yang terautentikasi yang dapat mengakses endpoint penting. Sistem juga menyediakan fitur tanda tangan digital pada dokumen PDF, di mana isi dokumen di-*hash* dan ditandatangani secara kriptografis untuk menjamin keaslian serta integritas dokumen. Kombinasi penggunaan enkripsi simetris, tanda tangan digital, dan manajemen sesi yang aman merupakan pendekatan yang efektif dan banyak direkomendasikan dalam literatur kriptografi untuk membangun layanan API yang aman dan andal (Daemen & Rijmen, 2002; Katz & Lindell, 2020; Menezes et al., 2018)**.**

B. Rumusan Masalah

1. Apakah layanan API yang dibangun dapat berfungsi sebagai *trusted authority server* dalam mengelola penyimpanan *public key* dan mendukung keamanan komunikasi antar pengguna.
2. Bagaimana penerapan mekanisme *digital signature* dan enkripsi pesan dapat menjamin keaslian, integritas, dan kerahasiaan data yang dikirimkan melalui sistem.
3. Mengapa penerapan *secure session* dan tanda tangan digital pada dokumen PDF diperlukan untuk meningkatkan keamanan akses layanan dan menjamin keaslian dokumen digital.

C. Tujuan Masalah

1. Membangun dan mengimplementasikan layanan API sebagai *trusted authority server* yang mampu mengelola penyimpanan *public key* serta menjadi pusat pengelolaan keamanan komunikasi antar pengguna.
2. Menerapkan mekanisme keamanan data berupa *digital signature* dan enkripsi pesan untuk menjamin keaslian, integritas, dan kerahasiaan informasi yang dipertukarkan melalui sistem.
3. Mengimplementasikan *secure session* dan tanda tangan digital pada dokumen PDF guna memastikan akses layanan hanya dilakukan oleh pengguna yang terautentikasi serta menjamin keaslian dan integritas dokumen digital.

BAB II

LANDASAN TEORI

A. Kriptografi Kunci Publik

Kriptografi kunci publik merupakan metode kriptografi yang menggunakan sepasang kunci yang saling berkaitan, yaitu *public key* dan *private key*. Kedua kunci ini memiliki peran yang berbeda namun saling melengkapi dalam menjamin keamanan komunikasi digital. Konsep kriptografi kunci publik diperkenalkan untuk mengatasi permasalahan distribusi kunci pada sistem kriptografi simetris dan menjadi fondasi utama dalam sistem keamanan modern (Diffie & Hellman, 1976).

1. ***Public key***
   Public Key adalah kunci yang dapat dibagikan secara bebas kepada pihak lain dan digunakan untuk proses verifikasi atau enkripsi data. Dalam konteks digital signature, *public key* digunakan untuk memverifikasi keaslian tanda tangan yang dibuat menggunakan *private key* yang bersesuaian. Keamanan sistem tetap terjaga meskipun *public key* diketahui oleh banyak pihak, karena kunci ini tidak dapat digunakan untuk menghasilkan tanda tangan digital (Rivest et al., 1978). Pada Project ini, *public key* pengguna disimpan di server dan digunakan dalam proses verifikasi tanda tangan digital serta autentikasi pesan.
2. ***Private Key***
   Private key merupakan kunci yang bersifat rahasia dan hanya boleh diketahui oleh pemiliknya. Kunci ini digunakan untuk melakukan proses penandatanganan pesan atau dokumen secara digital. Keamanan kriptografi kunci publik sangat bergantung pada kerahasiaan *private key*, karena pihak yang memiliki kunci ini dapat menghasilkan tanda tangan digital yang valid. Oleh karena itu, *private key* tidak pernah dikirimkan ke server atau pihak lain (Diffie & Hellman, 1976; Katz & Lindell, 2020).
   Dalam sistem yang dibangun, *private key* disimpan dan digunakan di sisi klien untuk menandatangani pesan maupun dokumen sebelum diverifikasi oleh server.

B. Digital Signature (Tanda Tangan Digital)

Digital signature merupakan mekanisme kriptografi yang digunakan untuk menjamin keaslian dan integritas suatu pesan atau data. Tanda tangan digital dibuat dengan menggunakan kunci privat pengirim dan dapat diverifikasi ***(/verify)*** menggunakan kunci publik yang sesuai. Jika pesan mengalami perubahan setelah ditandatangani, maka proses verifikasi akan gagal. Konsep digital signature menjadi dasar dalam sistem keamanan modern karena mampu membuktikan identitas pengirim sekaligus memastikan bahwa data tidak dimodifikasi selama transmisi

(Rivest et al., 1978).
Pada sistem ini, digital signature digunakan untuk memverifikasi pesan pengguna serta sebagai dasar dalam proses tanda tangan dokumen PDF ***(/signpdf)***.

C. Enkripsi Simetris (Advanced Encryption Standard / AES)

Enkripsi simetris adalah metode enkripsi yang menggunakan satu kunci yang sama untuk proses enkripsi dan dekripsi data. Advanced Encryption Standard (AES) merupakan algoritma enkripsi simetris yang telah distandarisasi dan banyak digunakan karena memiliki tingkat keamanan yang tinggi serta efisiensi komputasi yang baik. AES dirancang untuk melindungi kerahasiaan data dalam berbagai sistem informasi modern (Daemen & Rijmen, 2002). Dalam proyek ini, AES digunakan untuk mengenkripsi pesan yang di relay ***(/relay)*** antar pengguna sehingga isi pesan tidak dapat dibaca oleh pihak yang tidak berwenang.

D. Secure Session (JSON Web Token / JWT)

Secure session merupakan mekanisme untuk mengatur dan membatasi akses pengguna terhadap layanan sistem. JSON Web Token (JWT) adalah salah satu pendekatan *token-based authentication* yang memungkinkan server melakukan autentikasi tanpa menyimpan status sesi secara langsung. Token JWT berisi informasi pengguna yang ditandatangani secara kriptografis sehingga keasliannya dapat diverifikasi oleh server. Pendekatan ini banyak digunakan pada sistem berbasis API karena bersifat efisien, stateless, dan aman (Katz & Lindell, 2020). Pada proyek ini, JWT digunakan untuk memastikan bahwa hanya pengguna yang terautentikasi yang dapat mengakses endpoint penting.

E. Tanda Tangan Digital Dokumen PDF

Tanda tangan digital pada dokumen PDF bertujuan untuk menjamin keaslian dan integritas dokumen digital. Proses ini dilakukan dengan menghitung nilai *hash* dari isi dokumen, kemudian menandatangani hash tersebut menggunakan kunci privat. Dengan mekanisme ini, setiap perubahan pada dokumen setelah proses penandatanganan dapat terdeteksi. Penerapan tanda tangan digital pada dokumen merupakan praktik umum dalam sistem keamanan digital untuk menjamin keabsahan dokumen elektronik (Menezes et al., 2018). Dalam sistem yang dibangun, fitur tanda tangan PDF digunakan untuk memastikan bahwa dokumen yang ditandatangani oleh server tidak mengalami perubahan dan dapat diverifikasi keasliannya.

BAB III

PEMBAHASAN

A. Implementasi Layanan API (*api.py*)

api.py berfungsi sebagai server yang mengelola proses autentikasi pengguna, penyimpanan *public key*, pengiriman pesan, verifikasi tanda tangan digital, serta penandatanganan dokumen PDF.

layanan ini menyediakan beberapa fitur seperti *login* yang digunakan untuk melakukan autentifikasi pengguna dan menghasilkan token, *store public key* untuk menyimpan *public key* pengguna, *verifikasi signature* untuk memastikan bahwa suatu pesan benar-benar ditandatangani oleh pemilik kunci privat yang sah, *relay* untuk mengirim pesan antar pengguna, *decryption-demo* untuk mengubah menjadi plainteks, dan *sign pdf* menggunakan *private key server.*

B. Implementasi Entry Point Server *(main.py*)

main.py berfungsi sebagai tombol untuk mengaktifkan server kita. Ketika file ini dijalankan, ia akan memanggil *uvicorn* (web server) dan memberitahukan untuk membuka file aplikasi FastAPI yang berada di *api.py.* Hal ini membuat web server bisa diakses di jaringan lokal kita. Perlu digarisbawahi bahwa, penting untuk memisah antara *main.py* (tombol on) dan *api.py* (logika server) agar bisa fokus mengembangkan fitur di *api.py.*

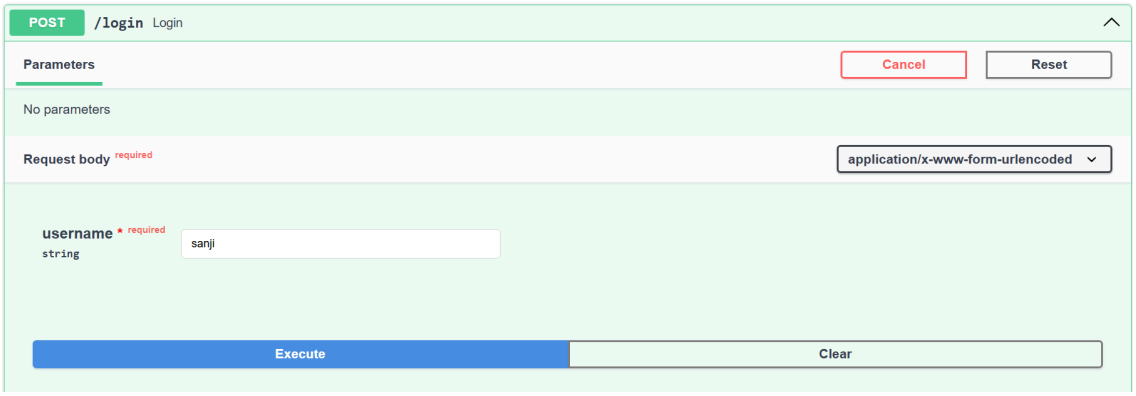
C. Implementasi Program Klien (*client.py*)

client.py berfungsi untuk menangani proses kriptografi di sisi pengguna, yaitu pembuatan kunci dan penandatanganan pesan. pada tahap awal, program menghasilkan pasangan *private key* dan *public key*. privat key digunakan untuk memproses tanda tangan pesan sedangkan public key akan dikirim ke server untuk proses registrasi kunci.

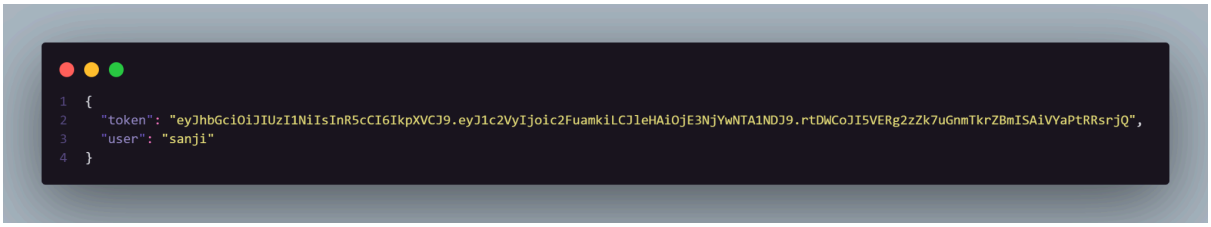
BAB IV

LANGKAH PENGERJAAN

A. Login

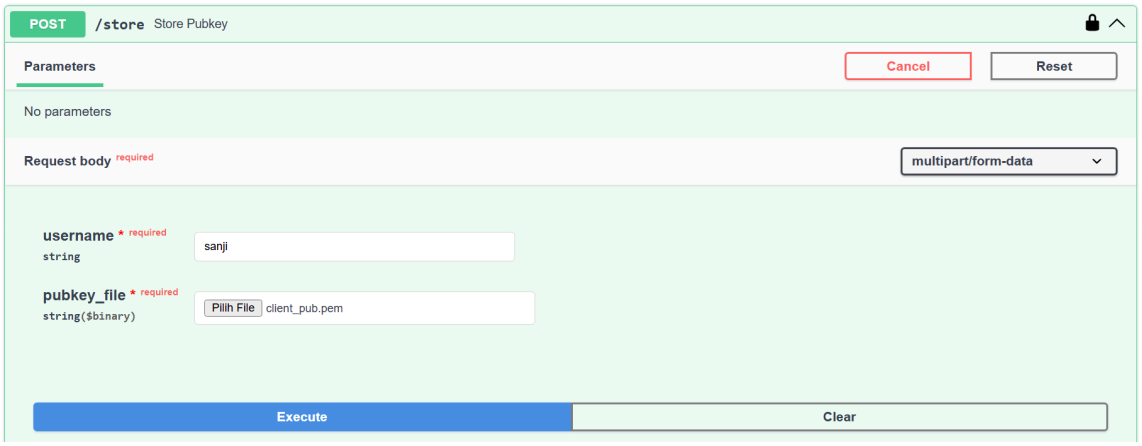


Pada langkah ini, pengguna melakukan proses login melalui endpoint **/login** pada Swagger UI dengan memasukkan username yang telah ditentukan. Proses login ini bertujuan untuk memperoleh token autentikasi (JWT) yang akan digunakan sebagai *secure session* dalam mengakses endpoint lain yang bersifat terlindungi pada sistem.



Kemudian sistem menampilkan token JWT dan username sebagai hasil login yang berhasil. Token ini digunakan untuk proses autentikasi pada endpoint selanjutnya.

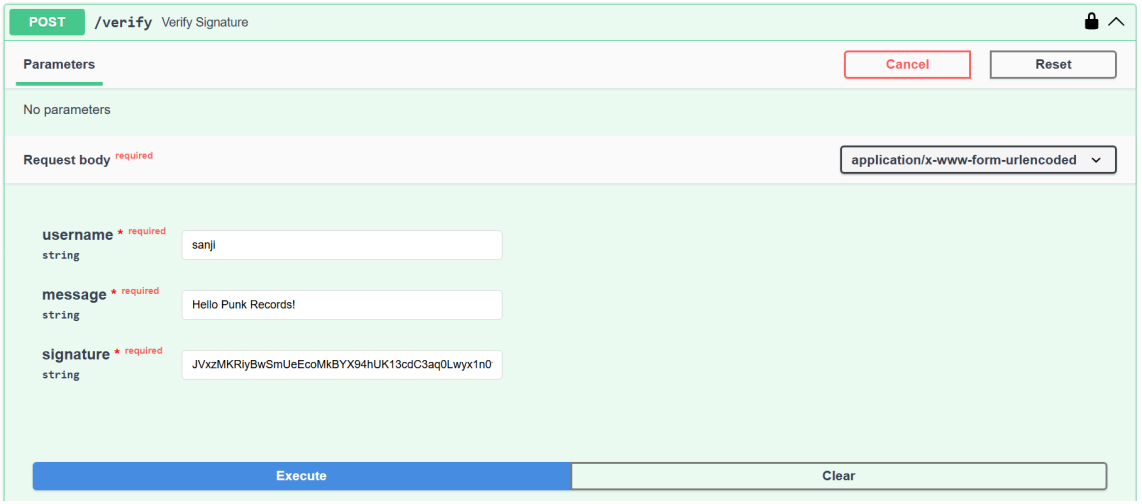
B. Simpan Public Key (/store)



Pada langkah selanjutnya, pengguna melakukan proses penyimpanan public key melalui endpoint **/store** dengan memasukkan username dan mengunggah file *public key* yang setelah didapatkan dari menjalankan kode *client.py* yang bernama *client_pub.pem*. Public key yang diunggah akan disimpan di server dan digunakan pada proses verifikasi tanda tangan digital serta pengamanan komunikasi selanjutnya.

Public key yang ada di file client_pub.pem didaftarkan sebagai public key milik user sanji.

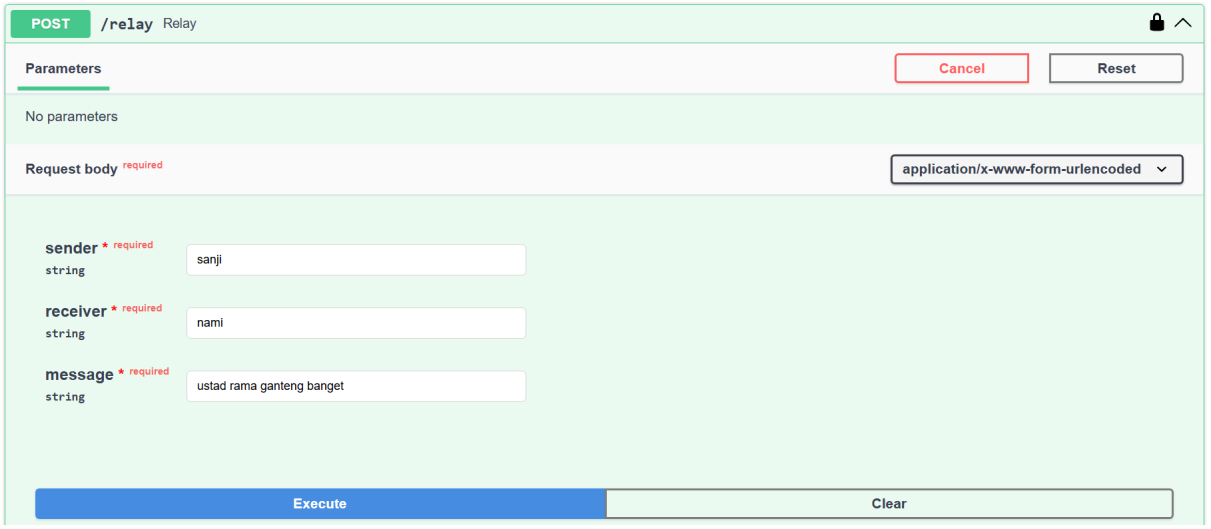
C. Verifikasi Signature



Langkah berikutnya, pengguna melakukan verifikasi tanda tangan digital melalui endpoint **/verify** dengan memasukkan username, pesan, dan signature yang dihasilkan output sebelumnya dari menjalankan kode *client.py*. Proses ini bertujuan untuk memastikan bahwa pesan benar-benar ditandatangani oleh pemilik *private key* yang sah dan bahwa isi pesan tidak mengalami perubahan.

Sistem menampilkan pesan “Signature VALID” yang menandakan bahwa tanda tangan digital berhasil diverifikasi dan sesuai dengan *public key* pengguna. Jika gagal diverifikasi akan tertulis “Signature INVALID” dikarenakan message dan signature yang dimasukkan tidak cocok.

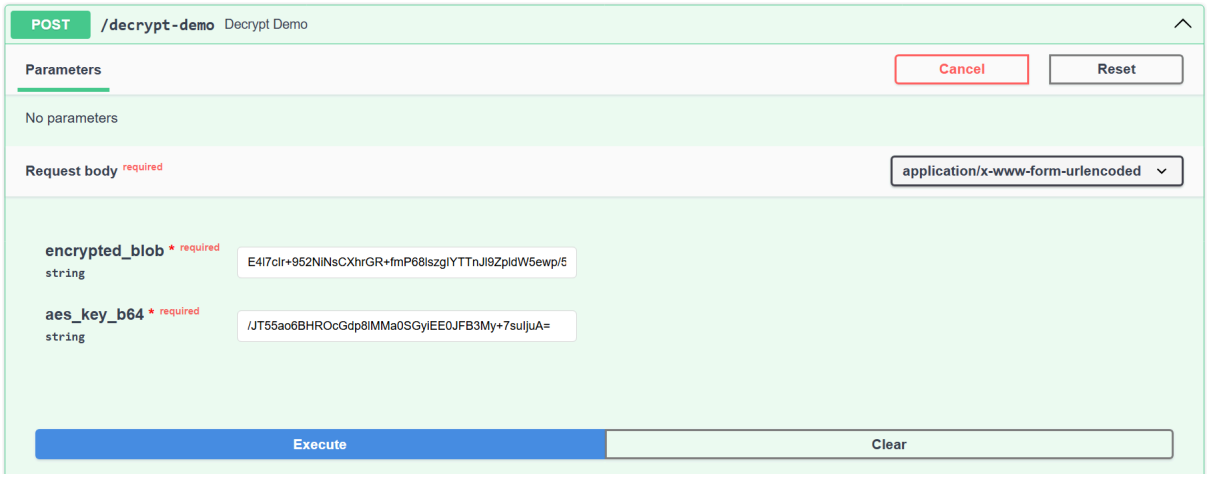
D. Relay Massage



Langkah selanjutnya, pengguna melakukan proses relay pesan melalui endpoint /relay dengan memasukkan pengirim, penerima, dan isi pesan. Pesan yang dikirim akan diproses oleh server sebagai perantara dan dienkripsi menggunakan mekanisme enkripsi simetris, kemudian diteruskan kepada penerima. Proses relay ini hanya dapat dilakukan apabila penerima telah memiliki pasangan kunci kriptografi (*public key* dan *private key*) yang valid, sehingga server dapat melakukan pengamanan pesan. Mekanisme ini bertujuan untuk memastikan bahwa isi pesan tetap bersifat rahasia selama proses pengiriman antar pengguna

Sistem akan menampilkan status relay berhasil, beserta pesan terenkripsi dan kunci AES dalam format *base64* sebagai hasil proses relay pesan antar pengguna. Jika pesan gagal di relay penyebabnya adalah apabila penerima belum memiliki *public key* yang belum tersimpan di server. Dalam kondisi tersebut, server tidak dapat melakukan proses enkripsi dan relay pesan karena kunci yang dibutuhkan untuk pengamanan komunikasi belum tersedia.

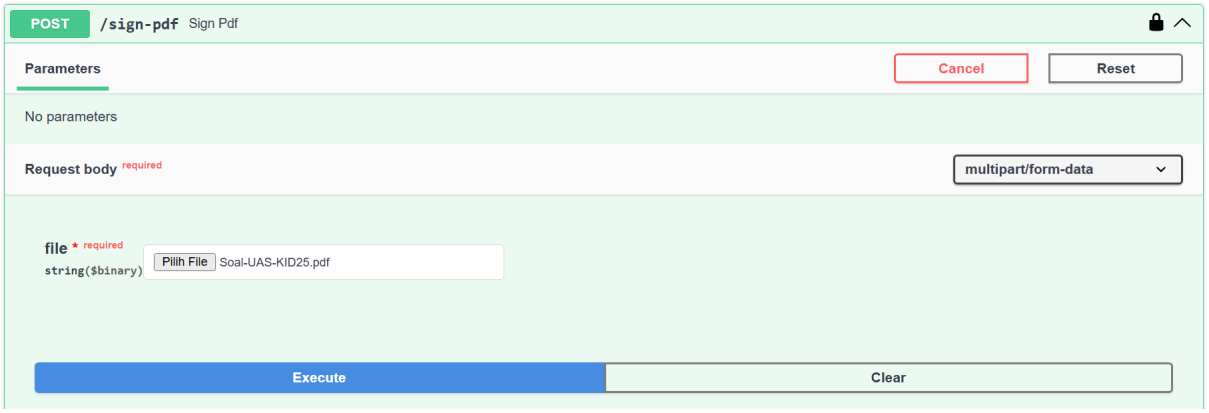
E. Decryption message



Pada langkah ini, dilakukan proses dekripsi pesan melalui endpoint **/decrypt-demo** dengan cara memasukkan encrypted message dan kunci AES dalam format *base64* yang diperoleh dari proses relay sebelumnya. Proses dekripsi ini menggunakan algoritma Advanced Encryption Standard (AES) dengan panjang kunci 256-bit, sehingga menjamin tingkat keamanan yang tinggi. Endpoint ini digunakan sebagai demonstrasi untuk membuktikan bahwa pesan yang telah dienkripsi oleh server dapat dikembalikan ke bentuk plaintext menggunakan kunci yang sesuai.

Sistem menampilkan kembali isi pesan dalam bentuk plain text, yang menandakan bahwa proses dekripsi menggunakan AES 256-bit berhasil dilakukan.

F. Tandatangan PDF



Langkah terakhir, dilakukan proses tanda tangan digital dokumen PDF melalui endpoint **/sign-pdf** dengan mengunggah file PDF yang akan ditandatangani. Sistem akan memproses dokumen dengan menghitung nilai *hash* dari isi file, kemudian menandatangani hash tersebut menggunakan private key server sebagai *trusted authority*. Proses ini bertujuan untuk menjamin keaslian dan integritas dokumen PDF, sehingga setiap perubahan pada dokumen setelah penandatanganan dapat terdeteksi.

Sistem menampilkan pesan PDF signed successfully, beserta nilai hash dokumen dan signature digital sebagai hasil proses tanda tangan dokumen PDF yang berhasil.

BAB V

KESIMPULAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa layanan API yang dibangun pada projek ini telah berhasil berfungsi sebagai *trusted authority server* sesuai dengan tujuan yang ditetapkan. Sistem mampu mengelola penyimpanan *public key* pengguna serta melakukan verifikasi *digital signature* untuk memastikan keaslian dan integritas pesan yang dipertukarkan antar pengguna. Dengan adanya mekanisme ini, pesan yang dikirim dapat dipastikan berasal dari pengirim yang sah dan tidak mengalami perubahan selama proses pengiriman.

Selain itu, sistem juga berhasil menerapkan mekanisme relay pesan aman dengan menggunakan enkripsi simetris AES 256-bit, sehingga kerahasiaan isi pesan tetap terjaga selama proses komunikasi antar pengguna. Penerapan *secure session* menggunakan JSON Web Token (JWT) juga berperan penting dalam membatasi akses ke layanan, sehingga hanya pengguna yang telah terautentikasi yang dapat menggunakan endpoint tertentu. Hal ini menunjukkan bahwa sistem telah menerapkan pengamanan akses yang cukup baik.

Fitur tanda tangan digital pada dokumen PDF yang diimplementasikan pada server juga dapat berjalan dengan baik. Dokumen yang ditandatangani menghasilkan nilai *hash* dan *signature* yang dapat digunakan untuk memverifikasi keaslian dokumen tersebut. Dengan demikian, sistem secara keseluruhan telah memenuhi aspek penilaian yang diberikan pada tugas, baik dari sisi fungsionalitas maupun penerapan konsep keamanan informasi, serta dapat dijadikan sebagai simulasi penerapan kriptografi pada layanan berbasis API.

DAFTAR PUSTAKA

Diffie, W., & Hellman, M. (1976). *New directions in cryptography*. IEEE Transactions on Information Theory, 22(6), 644–654. https://doi.org/10.1109/TIT.1976.1055638

Rivest, R. L., Shamir, A., & Adleman, L. (1978). *A method for obtaining digital signatures and public-key cryptosystems*. Communications of the ACM, 21(2), 120–126. https://doi.org/10.1145/359340.359342

Daemen, J., & Rijmen, V. (2002). *The Design of Rijndael: AES – The Advanced Encryption Standard*. Springer. https://doi.org/10.1145/359340.359342

Katz, J., & Lindell, Y. (2020). *Introduction to Modern Cryptography* (3rd ed.). CRC Press.

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2018). *Handbook of Applied Cryptography*. CRC Press.

LAMPIRAN

Link Githup: https://github.com/Albaary/UAS-Keamanan-dan-Integritas-Data