



UNIVERSITAS NEGERI  
SURABAYA

# IMPLEMENTASI KRIPTOGRAFI MENGGUNAKAN AES-256, ED25519, SHA-256, & HS256 PADA LAYANAN PUNK RECORDS BERBASIS FASTAPI



Disusun oleh:  
**Kelompok 9**

Mata Kuliah:  
**Keamanan & Integritas Data**

Dosen Pengampu:  
**Hasanuddin Al-Habib, M.Si.**  
**Moh. Khoridatul Huda, S.Pd., M.Si., Ph.D.**

# ANGGOTA KELOMPOK

---



**Cantika Latifatul Nur Ella**  
**24031554023**



**Sofia Dwi Kinasih**  
**24031554079**



**Ilmin Nur Lailiyah**  
**24031554135**

# LATAR BELAKANG

Pemanfaatan Application Programming Interface (API) dalam sistem informasi modern terus meningkat, terutama pada layanan berbasis web yang melibatkan pertukaran data antar pengguna secara real time. Namun, REST API memerlukan mekanisme keamanan berlapis yang kuat untuk melindungi dari ancaman penyadapan, manipulasi pesan, dan pemalsuan identitas pengguna, karena API tanpa lapisan kriptografi yang memadai sangat rentan terhadap serangan man-in-the-middle (Shofyan & Shita, 2024). Punk Records-v1 menerapkan empat algoritma kriptografi: **AES-256** untuk enkripsi data client-side menjaga kerahasiaan, **SHA-256** untuk integritas file, **HS256** untuk autentikasi token JWT, dan **Ed25519** untuk verifikasi tanda tangan digital. Sistem ini berfungsi menyimpan public key peneliti, memverifikasi pesan, dan melakukan relay pesan aman antar pengguna pada REST API berbasis FastAPI. Kombinasi keempat algoritma ini memastikan kerahasiaan, integritas, dan keaslian data dalam lingkungan laboratorium Vegapunk yang sensitif dan kolaboratif.





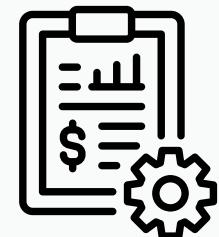
## TUJUAN

- 1.Untuk mengetahui bagaimana merancang dan mengimplementasikan layanan keamanan berbasis API yang menerapkan mekanisme kriptografi public key untuk menjamin keaslian dan integritas data, khususnya dalam proses pengiriman pesan dan unggah dokumen PDF.
- 2.Untuk mengetahui bagaimana penerapan tanda tangan digital menggunakan algoritma kriptografi public key dapat digunakan untuk memverifikasi bahwa data atau dokumen yang diterima tidak mengalami perubahan selama proses transmisi.
- 3.Untuk mengetahui bagaimana mekanisme autentikasi berbasis JWT diterapkan pada layanan API untuk memastikan bahwa setiap permintaan hanya dapat diakses oleh pengguna yang sah.



# TAHAPAN PENGERJAAN

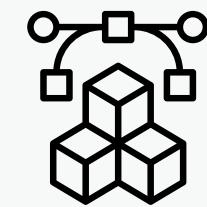
## Step 1



### **Menyiapkan *environment Project***

meliputi instalasi Python, pembuatan virtual environment, serta pemasangan library yang dibutuhkan seperti FastAPI, Uvicorn, cryptography, dan PyJWT.

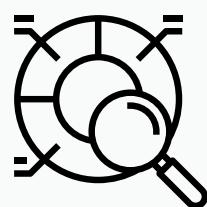
## Step 2



### **Membuat code client.py legkap**

Pembuatan pasangan public key dan private key untuk setiap pengguna (client). Serta client melakukan proses pembuatan digital signature terhadap pesan atau file (PDF).

## Step 3



### **Membuat code api.py legkap**

Server menyediakan berbagai endpoint seperti login (JWT), verify signature, upload PDF, relay pesan, health check, dan / (get\_index)

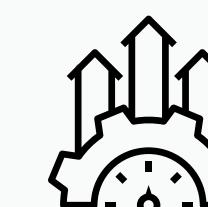
## Step 4



### **Menjalankan FASTAPI (main.py)**

Server dijalankan menggunakan Uvicorn. Permintaan client dan dokumentasi API dapat diakses melalui Swagger UI.

## Step 5



### **Pengujian Server pada localhost**

Tahap akhir adalah pengujian sistem secara menyeluruh pada localhost.



## KESIMPULAN

Server Punk Records-v1 berhasil diimplementasikan sebagai layanan keamanan berbasis API menggunakan FastAPI dengan penerapan kriptografi public key. Sistem mampu menjaga keaslian dan integritas data pada pengiriman pesan dan unggah dokumen PDF melalui tanda tangan digital Ed25519. Mekanisme autentikasi berbasis JWT (HS256) memastikan hanya pengguna yang sah dapat mengakses endpoint sensitif seperti /store, /verify, /relay, dan /upload-pdf. Secara keseluruhan, server telah memenuhi aspek keamanan yang ditetapkan, meliputi integrity check, variasi cipher, dukungan multiuser, dan secure session, sehingga meningkatkan keamanan pertukaran data secara signifikan.



## KENDALA

- Pada awal pengujian, beberapa endpoint menghasilkan status invalid akibat perbedaan penamaan public key antara kode client dan server. Solusi: menyamakan penamaan public key pada kedua sisi.
- Setiap pembaruan kode server mengharuskan menjalankan ulang uv run main.py agar Swagger API menampilkan endpoint versi terbaru. Solusi: selalu melakukan restart server setelah perubahan kode.



UNIVERSITAS NEGERI  
SURABAYA



# THANK YOU



Disusun oleh:  
**Kelompok 9**