BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era digital yang terus berkembang saat ini, penggunaan pembayaran non-tunai telah menjadi bagian tidak terpisahkan dari aktivitas masyarakat modern. Berbagai platform pembayaran digital yang menggunakan QRIS seperti OVO, DANA, dan GoPay memungkinkan transaksi dilakukan dengan cepat, mudah, dan praktis. Salah satu teknologi utama yang mendukung metode pembayaran ini adalah *QR Code (Quick Response Code). QR Code* digunakan sebagai media penyimpanan informasi transaksi yang dapat dipindai oleh perangkat untuk memproses pembayaran secara instan.

Namun, kemudahan yang ditawarkan oleh teknologi ini tidak terlepas dari risiko keamanan, seperti pencurian data, manipulasi informasi transaksi, ataupun serangan oleh pihak yang tidak bertanggung jawab. Oleh karena itu, diperlukan sistem keamanan yang kuat untuk melindungi data transaksi dari potensi ancaman.

RSA *(Rivest-Shamir-Adleman)* merupakan suatu algoritma kriptografi kunci publik yang banyak digunakan untuk melindungi data dalam berbagai aplikasi digital. Algoritma ini memiliki tingkat keamanan yang tinggi karena didasarkan pada kompleksitas faktorisasi bilangan prima yang sangat sulit untuk dipecahkan. Dalam konteks sistem pembayaran berbasis *QR Code,* penerapan algoritma RSA dapat digunakan untuk mengenkripsi informasi transaksi sehingga hanya pihak yang berwenang dapat mengakses data asli (Surya et al., 2024).

Berdasarkan latar belakang tersebut, penulis tertarik mengangkat judul **"Enkripsi RSA Untuk Pengamanan Data Pada Sistem Pembayaran Digital Berbasis QR Code"** dimana pada penelitian ini berfokus pada penerapan algoritma

RSA untuk mengamankan data pada sistem pembayaran digital berbasis *QR Code*. Dengan pendekatan ini, diharapkan sistem dapat meningkatkan keamanan informasi transaksi sekaligus memberikan solusi praktis dan aman dalam penggunaan pembayaran non-tunai.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah pada penelitian ini adalah:

1. Bagaimana mekanisme penerapan algoritma RSA dalam mengenkripsi data transaksi pada sistem pembayaran digital berbasis *Qr-Code.*
2. Bagaimana Validasi keamanan data yang telah dienkripsi menggunakan algoritma RSA.
3. Bagaimana performa sistem dalam mendukung proses enkripsi, dekripsi, dan pemindaian *Qr-Code.*

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu :

1. Menerapkan algoritma RSA untuk mengenkripsi data transaksi pada sistem pembayaran digital berbasis *Qr-Code.*
2. Menganalisis Tingkat keamanan data hasil enkripsi dengan menggunakan algoritma RSA.

1.4 Manfaat Penelitian

Manfaat yang diharapkan pada penelitian ini yaitu :

1. Memberikan keamanan dalam melakukan transaksi non-tunai sehingga mengurangi risiko pencarian data
2. Menjadi referensi dalam pengembangan aplikasi pembayaran digital berbasis *Qr-Code* yang lebih aman.

1.5 Batasan Masalah

Adapun batasan masalah yang digunakan dalam penelitian ini yaitu :

1. Penelitian hanya mencakup proses enkripsi dan dekripsi menggunakan algoritma RSA pada data transaksi seperti :
    - Nomor rekening penerima.
    - Jumlah pembayaran.
    - Waktu transaksi.
    - Informasi identitas pengguna.
2. *Qr-Code* hanya digunakan untuk menyimpan data transaksi yang telah dienkripsi.
3. Tidak mencakup proses autentikasi pengguna pada aplikasi pembayaran digital.
4. Bahasa pemrograman yang digunakan untuk implementasi program yaitu dengan menggunakan Python dengan antarmuka berbasis GUI Tkinter.

1.6 Sistematika Penulisan

Sistematika penyusunan skripsi ini dibagi menjadi lima bab, sesuai dengan sistematika/ketentuan dalam pembuatan skripsi, adapun pembagian bab-bab tersebut adalah:

BAB I : PENDAHULUAN

Pada bab ini diuraikan secara ringkas pembahasan tentang Latar Belakang, Identifikasi Masalah, Ruang Lingkup Masalah, Maksud dan Tujuan, Metode Penelitian, dan Sistematika Penulisan.

BAB II : LANDASAN TEORI

Pada bab ini diuraikan sekilas tentang pengertian Kriptografi, Algoritma RSA, Transaksi Non-Tunai, *Qr-Code,* dan juga sekilas tentang Python.

BAB III : METODE PENELITIAN

Pada bab ini menjelaskan metode penelitian yang digunakan meliputi desain sistem, tahapan penelitian, serta alat dan bahan yang digunakan.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini menguraikan proses implementasi algoritma RSA dalam sistem pembayaran digital berbasis *Qr-code,* serta menganalisis hasil dan pengujian sistem.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini menyajikan Kesimpulan dari hasil penelitian serta sasaran untuk pengembangan lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Kriptografi

Kriptografi merupakan salah satu metode untuk melindungi informasi data. Kriptografi adalah bidang ilmu yang mempelajari berbagai metode matematika terkait perlindungan informasi, termasuk aspek kerahasiaan, keaslian, integritas, serta autentikasi data. Hampir setiap aspek kehidupan saat ini menggunakan kriptografi sebagai sarana untuk memastikan keamanan dan kerahasiaan informasi, dengan memanfaatkan persamaan matematis untuk proses enkripsi dan dekripsi. Metode ini bekerja dengan mengonversi data menjadi kode tertentu, yang hanya bisa diakses dan dipahami oleh pihak yang memiliki otoritas (Ridho & Romli, 2024).

Kriptografi merupakan cabang ilmu komputer yang berhubungan dengan keamanan informasi digital. Kriptografi mengelola data dengan cara menjaga kerahasiaan dan integritas selama dan setelah proses transmisi. Algoritma kriptografi dibagi menjadi dua bagian berdasarkan kuncinya: kriptografi simetris dan kriptografi kunci publik. Kriptografi simetris memiliki kunci yang sama untuk enkripsi dan dekripsi data, sedangkan kriptografi kunci publik terdiri dari dua kunci yaitu kunci publik untuk enkripsi data dan kunci privat untuk dekripsi data (Budiman et al., 2023).

Kriptografi berkaitan dengan merancang dan menggunakan kode (atau *ciphers*) yang memungkinkan dua belah pihak mengirim pesan secara tersembunyi dari seorang *hacker* yang dapat memantau semua komunikasi antara mereka. Dalam bahasa modern, kode disebut skema enkripsi dan begitulah terminologi kriptografi.

Keamanan dari skema enkripsi klasik mengandalkan rahasia dan juga kunci yang dibagikan oleh pihak yang berkomunikasi sebelumnya dan tidak diketahui oleh penyadap.

Kriptografi telah digunakan selama ribuan tahun untuk membantu menyediakan rahasia komunikasi antara pihak-pihak yang saling percaya. Dalam bentuknya yang paling dasar, dua orang, sering dilambangkan sebagai Alice dan Bob, telah menyepakati kunci rahasia tertentu. Di lain waktu, Alice mungkin ingin mengirim pesan rahasia ke Bob (atau Bob mungkin ingin mengirim pesan ke Alice). Kunci digunakan untuk mengubah pesan asli (yang biasanya kita sebut dengan *plaintext*) menjadi bentuk acak yang tidak dapat dipahami kepada siapa saja yang tidak memiliki kunci. Proses ini disebut enkripsi, dan pesan yang diacak disebut *ciphertext*. Ketika Bob menerima *ciphertext*, dia dapat menggunakan kunci untuk mengubah *ciphertext* kembali menjadi *plaintext* atau teks asli, ini disebut dengan proses dekripsi (Stinson & Paterson, 2019).

2.2 Relatif Prima

Bilangan relatif prima merupakan dua buah bilangan bulat m dan n yang apabila hasil GCD (m, n) sama dengan 1 (Cormen et al., 2009).

Contoh :

Apakah 14 dan 25 relatif prima?

Maka cari $GCD\ (14, 25)$

$25\ mod\ 14\ =\ 11$

$14\ mod\ 11\ =\ 3$

$11\ mod\ 3\ =\ 2$

$3\ mod\ 2\ =\ 1$

$2\ mod\ 1\ =\ 0$

Jadi $GCD\ (14, 25)\ =\ 1$

Maka 14 dan 25 adalah relatif prima.

2.3 Modulo Eksponensial

Modulo eksponensial adalah operasi dalam perhitungan matematika yang meningkatkan suatu angka ke modulo nomor lain. Lebih khusus lagi, pemecahan masalah dalam komputasi $a^b\ mod\ n$. Dimana a dan b adalah bilangan bulat non-negatif dan n adalah bilangan bulat positif (Cormen et al., 2009).

2.4 GCD *(Greatest Common Divisor)*

Greatest Common Divisor atau yang disingkat dengan *GCD* merupakan Pembagi persekutuan terbesar dari dua buah bilangan bulat a dan b yang apabila bilangan bulat terbesar dapat membagi habis keduanya (Cormen et al., 2009). Adapun algoritma untukmenentukan *Greatest Common Divisor* atau *GCD* yaitu dengan menggunakan Algoritma *Euclidean* dengan perhitungan :

Algoritma *Euclidean* dengan perhitungan :

1. *Input* bilangan bulat a, b dimana $a \leq b$.
2. Bagi bilangan a dengan b, jika hasil sisanya $r = 0$ maka $GCD\ (a, b) = b$ (berhenti).
3. Jika hasil sisanya $r \neq 0$ maka ganti $a = b, b = r$. Kemudian ulangi langkah nomor 2 sampai hasil sisa $r = 0$.

   Contoh : $GCD\ (45, 200) = \cdots$

   $200\ mod\ 45 = 20.$

   $45\ mod\ 20 = 5.$

   $20\ mod\ 5 = 0$ (berhenti).

2.5 RSA

RSA adalah algoritma yang dikembangkan pada tahun 1977 oleh Ron, Rivest, Adi Shamir, dan Leonardo Adleman. Dalam kriptografi asimetris, kunci enkripsi adalah kunci publik dan kunci dekripsi yang berbeda dari kunci enkripsi yang dijaga kerahasiaannya. Karena dua kunci berbeda digunakan dalam enkripsi dan dekripsi, algoritma RSA juga disebut sebagai algoritma kriptografi asimetris (Ray et al., 2017).

Salah satu fitur utama RSA adalah modulus $N = p \times q$ di mana p dan q adalah bilangan prima besar yang memenuhi persamaan $q < p$. Misalkan $\varphi(N) = (p - 1)(q - 1)$ adalah fungsi totient Euler. Kemudian untuk nilai e dan d ditetapkan sebagai kunci *public* dan *private* (Nitaj et al., 2022).

Skema RSA adalah skema yang paling banyak digunakan pada sistem enkripsi kunci *public*. RSA mengeksploitasi properti matematis dari eksponensial modular dan bergantung pada sulitnya memfaktorkan bilangan yang besar (Zhang et al., 2020).

Algoritma RSA terdiri dari 3 langkah utama dalam enkripsi dan dekripsi. Adapun langkah-langkahnya sebagai berikut :

A. Pembangkitan Kunci

RSA melibatkan kunci publik dan kunci pribadi. Dari kedua kunci ini kunci publik digunakan untuk mengenkripsi pesan dan dapat diketahui semua orang. Pesan yang dienkripsi dengan kunci publik didekripsi menggunakan kunci privat. Adapun proses pembuatan kunci yaitu sebagai berikut :

1. Pilih dua buah bilangan prima, p dan q.
2. Hitung nilai dengan cara $n = p \times q$

3. Hitung $\phi\ (n)\ =\ (p-1)(q-1)$
4. Pilih sebuah bilangan bulat e untuk kunci *public*, e relatif prima terhadap $\phi(n)$
5. Hitung kunci dekripsi d, dengan persamaan $ed \equiv e^{-1}\ mod\ (\phi(n))$

   Kunci *public* = (n, e) Kunci *Private* = (n, d)

B. Proses Enkripsi

Adapun langkah-langkah dalam mengenkripsi yaitu :

1. Nyatakan pesan menjadi blok-blok *plainteks* : $M_1 M_2, M_3 \ldots$ (Syarat : $0\ \leq\ M, \leq\ n\ -\ 1$)
2. Hitung blok *chiperteks* C_i untuk blok *plainteks* p_i dengan persamaan $C_i = M_i^e\ mod\ n$yang dalam hal ini, e adalah kunci publik.

C. Proses Dekripsi

Proses dekripsi dilakukan dengan menggunakan persamaan $M_i = C_i^d\ mod\ n$ yang dalam hal ini, d adalah kunci *privat*.

Contoh perhitungan RSA :

1. Pilih 2 bilangan prima p dan q dimana $p\ \neq q$.

   Pada proses perhitungan ini saya memilih $p = 7$ $q = 13$.
2. Hitung modulus n dengan cara mengalikan factor prima p dan q.

   Yaitu $n = p \times q$

   $n = 7 \times 13 = 91$
3. Hitung nilai $\phi(n) = (p-1) \times (q-1)$

   $\phi(n) = (7-1) \times (13-1)$

   $\phi(n) = 6 \times 12 = 72$
4. Cari nilai *eksponen publickey* "e" dengan menggunakan GCD $(e, \phi(n)) = 1$ dan harus terletak diantara $1 < e < \phi(n)$.

$e = 29$

Pembuktian GCD (29, 72) = 1

$$29\ mod\ 72 = 29$$

$$72\ mod\ 29 = 14$$

$$29\ mod\ 14 = 1$$

5. Cari nilai eksponen *private key* "d" yang memenuhi persamaan $(d \times e)\ mod\ \phi(n) = 1$

   $(d \times 29)\ mod\ \phi(n) = 1$

   $d = 5$

   Pembuktian $d = (5 \times 29)\ mod\ 72 = 1$

Tabel 2.1 Pembuktian $(5 \times 29)\ mod\ 72 = 1$

d	**e**	$\boldsymbol{d \times e\ mod\ \phi(n) = 1}$
1	29	$1 \times 29\ mod\ 72 = 29$
2	29	$2\ \times 29\ mod\ 72 = 58$
3	29	$3\ \times 29\ mod\ 72 = 15$
4	29	$4\ \times 29\ mod\ 72 = 44$
5	29	$5\ \times 29\ mod\ 72 = 1$

6. Konversi *Plaintext* kedalam ASCII.

   Misalkan kalimat asli yang akan kita enkripsi adalah "**UISU**" maka kita perlu mengkonversi kalimat asli tersebut kedalam bilangan ASCII. Adapun hasil konversi kalimat asli kedalam bilangan ascii dapat dilihat pada table 2.2 berikut ini :

Tabel 2.2 Konversi Plaintext Ke ASCII

Plaintext	**ASCII**
U	85
I	73
S	83
U	85

7. Enkripsi pesan.

   Untuk mengenkripsi pesan kita harus menggunakan rumus :

$$C = p^e \ mod \ n$$

Hitung satu per satu untuk setiap huruf. Contoh :

1. $C = \boldsymbol{U} = 85^{29} \ mod \ 91$

   $C = \mathbf{50}$

2. $C = \boldsymbol{I} = 73^{29} \ mod \ 91$

   $C = 47$

3. $C = \boldsymbol{S} = 83^{29} \ mod \ 91$

   $C = \mathbf{83}$

4. $C = \boldsymbol{U} = 85^{29} \ mod \ 91$

   $C = 50$

Maka didapatlah *ciphertext* = 50 47 83 50

8. Dekripsi Pesan

Untuk mendekripsi pesan kita harus menggunakan rumus :

$$P = C^d \ mod \ n$$

Hitung satu per satu untuk setiap huruf. Contoh :

1. $P = 50^5 \ mod \ 91$

   $P = 85$ = **U**

2. $P = 47^5 \ mod \ 91$

   $P = 73$ = **I**

3. $P = \ 83^5 \ mod \ 91$

   $P = 83$ = **S**

4. $P = \ 50^5 \ mod \ 91$

   $P = 85$ = **U**

Maka didapatlah *plaintext* $\frac{85}{\boldsymbol{U}} \frac{73}{\boldsymbol{I}} \frac{83}{S} \frac{85}{U}$

2.6 *Qr-Code (Quick Response Code)*

Quick Response Code (QR-Code) adalah bentuk evolusi kode batang dari satu dimensi menjadi 2 dimensi. Gagasan di balik pengembangan kode QR adalah keterbatasan kapasitas informasi barcode hanya dapat menampung 20 karakter alfanumerik (Munawar, 2019).

QR Code merupakan sebuah gambar dua dimensi yang menyajikan sebuah data, terutama data berbentuk teks. *QR Code* merupakan perkembangan dari sebuah barcode yang awalnya satu dimensi menjadi dua dimensi. *QR Code* berisi informasi baik diarah vertical dan horizontal, sedangkan barcode berisi data dalam satu arah saja. *QR Code* memiliki ukuran yang lebih besar informasi dari *barcode* (Anjani et al., 2021).

Gambar 2.1 *Qr-Code*

2.7 Sistem Pembayaran Digital

Sistem pembayaran digital adalah metode pembayaran yang dilakukan melalui perangkat elektronik, seperti ponsel, komputer, atau kartu pembayaran (debit, kredit, atau prabayar). Sistem ini memanfaatkan teknologi keuangan (FinTech) untuk memungkinkan transaksi yang lebih cepat, mudah, dan aman dibandingkan pembayaran konvensional berbasis tunai. Digitalisasi dalam sistem pembayaran telah menjadi tren global yang signifikan, didorong oleh meningkatnya adopsi internet dan perangkat teknologi informasi. Peralihan dari transaksi berbasis uang tunai menuju digital telah memperkuat inklusi keuangan dan memodernisasi cara individu serta bisnis menjalankan transaksi sehari-hari (Eren, 2024).

Kemajuan teknologi yang pesat memberikan dampak signifikan terhadap perkembangan sistem pembayaran dalam dunia bisnis, khususnya dalam menjaga kelangsungan hubungan bisnis antar pihak. Sistem pembayaran, yang merupakan salah satu pilar penting dalam menjaga stabilitas keuangan, telah mengalami transformasi dari yang awalnya berbasis uang tunai menjadi sistem pembayaran digital atau electronic money (e-money). Inovasi teknologi ini telah menggeser peran uang tunai (currency) sebagai alat pembayaran ke arah sistem pembayaran non-tunai yang lebih efisien dan ekonomis. Pembayaran non-tunai kini dilakukan tanpa menggunakan uang fisik, melainkan melalui transfer antar bank atau transfer

intra bank menggunakan jaringan internal bank. Selain itu, pembayaran non-tunai juga dapat dilakukan menggunakan kartu sebagai alat transaksi, seperti kartu ATM, kartu debit, maupun kartu kredit (Tarantang et al., 2019).

2.8 *Flowchart*

Flowchart adalah salah satu visualisasi aliran logis yang berlaku untuk sebagian besar disiplin ilmu. Ini cukup standar. Sintaksnya sangat sederhana, yang membuat bahasanya unik itu sangat kuat sehingga Anda bisa idealnya membuat konsep aliran logis apa pun dengannya.

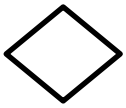
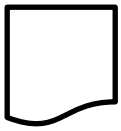
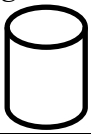
Sebagaimana dinyatakan di atas, diagram alur dapat digunakan untuk membangun aliran logis dari program perangkat lunak. *Flowchart* memudahkan pengembang untuk memahami, men-debug, dan memeriksa validitas kode untuk pengembang. Ada banyak paket perangkat lunak untuk menghasilkan kode dari diagram *Flowchart* .

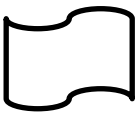
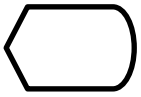
Tapi tidak ada cara untuk membuat diagram alur dari kode sumber. Bagaimana jika kita memodelkan solusi untuk mengonversi kode sumber menjadi diagram alur yang dapat memudahkan pengembang perangkat lunak dalam memahami, men-debug, dan memvalidasi kode.

Meskipun *Flowchart* adalah alat diagram, itu dapat direpresentasikan dalam sintaks seperti grafik terarah. Masalah yang dihadapi adalah semacam masalah terjemahan/transformasi. Masalah terjemahan bahasa pemrograman biasanya diselesaikan menggunakan kompiler. Karena masalah yang dihadapi adalah menerjemahkan kode sumber program ke diagram alur, kami mengusulkan solusi untuk meniru arsitektur kompiler dalam teori *compiler* (Pan et al., n.d.).

Adapun arti dari simbol *Flowchart* adalah sebagai berikut :

Tabel 2. 1 Simbol Flowchart

Simbol	Arti
Input / *Output* 	Mempresentasikan *input* data atau *output* data yang diproses atau informasi
Proses 	Mempresentasikan Operasi
Penghubung 	Keluar ke atau masuk dari bagian lain *Flowchart* khususnya halaman yang sama
Anak Panah 	Mempresentasikan alur kerja
Keputusan 	Keputusan dalam program
Predefined Process 	Rincian operasi berada di tempat lain
Preparation 	Pemberian harga awal
Punched Card 	Input / Output yang menggunakan kartu berlubang
Dokumen 	Input / Output dalam format yang dicetak
Magnetic Disk 	*Input* / *Output* yang menggunakan *Disk Magnetic*

Magnetic Drum 	*Input* / *Output* yang menggunakan *Drum Magnetic*
Punched Tape 	*Input* / *Output* yang menggunakan pita kertas berlubang
Manual Input 	*Input* yang dimasukkan secara manual dari *keyboard*
Display 	*Output* yang ditampilkan pada terminal
Manual Operation 	Operasi Manual

2.9 Python

Python adalah bahasa pemrograman interpretative multiguna dengan filosofi perancagan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif (Sinaga, 2017).

Python diciptakan pertama kali oleh Guido Van Rossum di belanda pada tahun 1990 dan namanya diambildari acara televisikesukaanGuido Monty Python's Flying Circus. Van Rossum mengembangkan Python sebagaihobi, kemudian Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak sintuitif dan memiliki pustaka yang luas (Romzi & Kurniawan, 2020).

Cara Kerja Python :

Python adalah bahasa pemrograman interpretatif, yang berarti kode dieksekusi baris per baris oleh interpreter tanpa perlu dikompilasi terlebih dahulu ke dalam bentuk binary seperti bahasa pemrograman terkompilasi (misalnya, C atau Java). Hal ini memungkinkan pengembang untuk menjalankan dan menguji kode dengan lebih cepat, serta mempermudah proses debugging.

` Saat program Python dijalankan, interpreter membaca dan menerjemahkan kode sumber ke dalam bytecode, yang kemudian dieksekusi oleh Python Virtual Machine (PVM). Python juga memiliki garbage collector otomatis yang mengelola alokasi dan dealokasi memori, sehingga pengembang tidak perlu mengelola memori secara manual.

Python mendukung berbagai paradigma pemrograman, seperti:

1. **Pemrograman Berorientasi Objek (OOP) –** Memungkinkan penggunaan kelas dan objek untuk mengorganisasi kode.
2. **Pemrograman Fungsional –** Mendukung fungsi tingkat tinggi, seperti lambda function dan higher-order function.
3. **Pemrograman Prosedural –** Menggunakan pendekatan berbasis prosedur dengan fungsi dan modul.

Selain itu, Python dapat digunakan untuk berbagai kebutuhan, seperti pengembangan web, kecerdasan buatan, analisis data, dan komputasi ilmiah. Dengan ekosistem pustaka yang luas, seperti NumPy, Pandas, TensorFlow, dan Django, Python menjadi alat yang sangat fleksibel dalam berbagai bidang pengembangan perangkat lunak (Masnur et al., 2024).