BAB II

LANDASAN TEORI

2.1 Kajian Ilmiah

Hasil penelitian yang telah dilakukan para peneliti dapat dijadikan dasar atau kajian untuk mempermudah dalam melakukan penelitian. Beberapa diantaranya adalah penelitian dengan judul Teknik Pemecahan Kunci Algoritma Rivest Shamir Adleman (RSA) dengan Metode Kraitchik. Peneliti mencari kunci privat algoritma RSA dengan memfaktorkan kunci publik n dengan Metode *Kraitchik*, kemudian dilihat efisiensi waktu pemfaktorannya. Hasil penelitian memperlihatkan, bahwa semakin besar selisih antara faktor kunci *p* dan *q*, maka semakin besar pula waktu pemfaktorannya. Pemfaktoran kunci publik (n) sebesar 19 digit (152 *bit*) dengan selisih faktor kunci (p-q) = 22641980 membutuhkan waktu 93,6002 ms lebih cepat dibandingkan dengan panjang kunci 15 digit (120 bit) dengan selisih faktor kunci (p-q) = 23396206 yang membutuhkan waktu selama 5850,0103 ms. Faktor lain yang juga memengaruhi adalah GCD(p-1, q-1), panjang kunci dan faktor prima (p-1), (q-1). (Muchlis dkk., 2017)

Dan mengambil konstanta p dan q sebagai acuan dari penelitian ini dengan judul Mengukur Kecepatan Enkripsi dan Dekripsi Algoritma RSA pada Pengembangan Sistem Informasi *Text Security*. Objek penelitian ini adalah proses implementasi algoritma kriptografi RSA pada nilai parameter n dengan ukuran

1024 *bit* dan 2048 *bit*. Proses yang diamati adalah kompleksitas waktu yang dihasilkan oleh instruksi enkripsi dan dekripsi. Tahapan yang dilakukan adalah studi pendahuluan, mengumpulkan data, menganalisis kebutuhan, pengembangan dan pengujian sistem informasi serta penarikan kesimpulan. Hasil pengujian menyatakan algoritma RSA 1024 bit memiliki rata-rata kecepatan enkripsi sebesar 352.488 nano second dan rata-rata kecepatan dekripsi sebesar 109.347.917 *nano second*, sedangkan pada algoritma RSA 2048 *bit* memiliki rata-rata kecepatan enkripsi sebesar 1.772.900 *nano second* dan rata-rata kecepatan dekripsi sebesar 775.282.334 *nano second*. (Wulansari dkk., 2016)

2.2 Dasar Teori

Hubungan matematika dengan kriptografi sangat erat sekali, karena matematika adalah konsep dasar yang berhubungan dengan kriptografi terutama matematika diskrit (Firmansyah, 2015). Dalam bab 2 ini, akan menjelaskan konsep matematis yang melandasi pembentukan konstanta p dan q dengan algoritma bilangan prima, seperti teori bilangan bulat, keterbagian, sifat-sifat pembagian, algoritma euklid, aritmatika modulo, logaritma diskrit dan bilangan prima.

2.2.1 Teori Bilangan

Teori bilangan salah satunya bilangan prima dengan berbagai metode, banyak yang dapat dipelajari, namun masih sebatas bilangan prima dengan jumlah digit yang sederhana (kecil) misalnya 2, 3, 5, 7, 11, 17, 23, 29 dan seterusnya Sehingga metode untuk mendapatkan bilangan prima yang besar perlu dikupas lagi. (Harahap, 2019) mengeluarkan bilangan prima yang terbesar saat ini ditemukan oleh manusia adalah $2^{74207281}$-1 dengan jumlah 22.338.618 digit

Dalam pengertian yang ketat, kajian tentang sifat-sifat bilangan asli disebut dengan teori bilangan. Dalam pengertian yang lebih luas, teori bilangan mempelajari bilangan dan sifat-sifatnya. Sebagai salah satu cabang matematika, teori bilangan dapat disebut sebagai "aritmetika lanjut (*advanced aritmetics*)" karena terutama berkaitan dengan sifat-sifat bilangan asli. Teori bilangan merupakan dasar perhitungan dan menjadi salah satu teori yang mendasari pemahaman kriptografi, bilangan yang dimaksud hanyalah bilangan bulat (integer).

Berikut penjelasan mengenai bilangan bulat, keterbagian, algoritma pembagian, fungsi eular, bilangan prima, relatif prima, modulus dan *Greatest Common Divisor* (GCD).

1 Bilangan Bulat

Bilangan bulat positif yang lebih besar dari 1 dan hanya habis dibagi dirinya sendiri dan bilangan 1 disebut Bilangan Prima (Harahap, 2019). Bilangan bulat adalah bilangan yang tidak mempunyai pecahan desimal. Himpunan semua bilangan bulat yang dinotasikan dengan Z yang diambil dari kata Zahlen dari bahasa Jerman atau dinotasikan dengan I yang diambil dari huruf pertama kata Integer dari bahasa Inggris, adalah himpunan *... , −3, −2, − 1,0,1,2,3 ... +. Himpunan bilangan bulat dibagi tiga, yaitu bilangan bulat positif, yaitu bilangan bulat yang lebih besar dari nol yang dituliskan Z + , nol, dan bilangan bulat negatif, yaitu bilangan bulat yang lebih kecil dari nol yang dituliskan Z −. Himpunan bilangan bulat dilengkapi dengan dua buah operasi, yaitu operasi penjumlahan dan perkalian, dilambangkan (Z, +,·) membentuk suatu sistem matematika yang disebut gelanggang atau ring. Himpunan bilangan bulat berperan sangat penting dalam kriptografi karena banyak

algoritma kriptografi yang menggunakan sifat-sifat himpunan bilangan bulat dalam melakukan proses penyandiannya.

2 Keterbagian

Sifat-sifat yang berkaitan dengan keterbagian (*divisibility*) merupakan dasar pengembangan teori bilangan. Jika suatu bilangan bulat dibagi oleh suatu bilangan bulat yang lain, maka hasil pembagiannya adalah bilangan bulat atau bukan bilangan bulat.

3 Algoritma Pembagian

Jika a, b $\in$ Z dan a > 0, maka ada bilangan q, r $\in$ Z yang masing-masing tunggal sehingga b = qa + r dengan 0 r < a. Jika a$\nmid$b, maka r memenuhi ketidaksamaan 0 < r < a. Algoritma pembagian adalah suatu cara atau prosedur yang dapat dipakai untuk mendapatkan faktor persekutuan terbesar (FPB) (Fernanda, 2020).

4 Fungsi Euler (φ)

Fungsi Euler digunakan untuk menyatakan banyaknya bilangan bulat < n yang relatif prima terhadap n. Jika p adalah suatu bilangan prima, maka $\varphi(p) = p - 1$, Karena p adalah bilangan prima, maka setiap bilangan bulat positif kurang dari p relatife prima terhadap p. Ini berarti bahwa sistem residu tereduksi modulo p adalah himpunan {1, 2, 3,...,p - 1} yang mana seluruh anggotanya sebanyak (p - 1) sehingga $\varphi(p) = p - 1$.

5 Bilangan Prima

Bilangan bulat positif yang mempunyai aplikasi penting dalam ilmu komputer dan matematika diskrit adalah bilangan prima. Bilangan prima adalah bilangan bulat positif yang lebih dari 1 yang hanya habis dibagi oleh 1 dan dirinya 23 sendiri. Sifat pembagian pada bilangan bulat melahirkan konsep-konsep bilangan prima dan aritmetika modulo, dan salah satu konsep bilangan bulat yang digunakan dalam penghitungan komputer adalah bilangan prima. Dengan ditemukannya bilangan prima, teori bilangan berkembang semakin jauh dan lebih mendalam. Banyak dalil dan sifat dikembangkan berdasarkan bilangan prima. Bilangan prima juga memainkan peranan yang penting pada beberapa algoritma.
Jika p suatu bilangan bulat positif lebih dari 1 yang hanya mempunyai pembagi positif 1 dan p, maka p disebut bilangan prima. Jika suatu bilangan bulat q 1 bukan suatu bilangan prima, maka q disebut bilangan komposit. Untuk menguji apakah p merupakan bilangan prima atau bilangan komposit, dapat menggunakan cara yang paling sederhana, yaitu cukup membagi p dengan sejumlah bilangan prima, yaitu 2, 3, ... , bilangan prima √p. Jika p habis dibagi salah satu dari bilangan prima tersebut, maka p adalah bilangan komposit tetapi jika p tidak habis di bagi oleh semua bilangan prima tersebut, maka p adalah bilangan prima.

Solusi sederhana menentukan bilangan prima dalam deret bilangan dapat dilakukan dengan *naive solution* dimana $2\ ke\ n - 1$ menghasilkan $arrayListPrimeNumber = 2, 3, 5..n$ dan n atau batas prima merupakan bilangan yang ditentukan. Solusi lain memiliki kebutuhan yang beragam atau

kompleks, sedangkan penelitian ini membutuhkan konsep yang sederhana dan fleksibel untuk diterapkan dalam berbagai konsep.

Beberapa fakta menarik tentang bilangan prima (*Prime Numbers - GeeksforGeeks*, n.d.)

1. Dua adalah satu-satunya bilangan prima genap.
2. Setiap bilangan prima dapat direpresentasikan dalam bentuk 6n + 1 atau 6n-1 kecuali 2 dan 3, di mana n adalah bilangan asli.
3. Dua dan Tiga hanyalah dua bilangan asli berurutan yang juga merupakan bilangan prima.
4. Dugaan Goldbach: Setiap bilangan bulat genap yang lebih besar dari 2 dapat diekspresikan sebagai jumlah dari dua bilangan prima.
5. Teorema Wilson: Teorema Wilson menyatakan bahwa bilangan asli p > 1 adalah bilangan prima jika dan hanya jika

$(p - 1)! \equiv -1 \bmod p$

ATAU $(p - 1)! \equiv (p - 1) \bmod p$

6. Teorema Kecil Fermat: Jika n adalah bilangan prima, maka untuk setiap a, 1 <= a <n,

$$an - 1 \equiv 1 \pmod n$$

ATAU

$$an - 1 \,\%\, n = 1$$

7. Teorema Bilangan Perdana: Probabilitas suatu bilangan yang dipilih secara acak n adalah bilangan prima berbanding terbalik dengan jumlah digitnya, atau dengan logaritma dari n.

8. Dugaan Lemoine: Setiap bilangan bulat ganjil yang lebih besar dari 5 dapat diekspresikan sebagai jumlah bilangan prima ganjil (semua bilangan prima selain 2 adalah ganjil) dan semiprime genap. Bilangan semiprima adalah hasil perkalian dua bilangan prima. Ini disebut dugaan Lemoine.

6 Relatif Prima

Secara ringkas, relatif prima merupakan dua buah bilangan bulat a dan b dikatakan relatif prima jika GCD atau FPB (a, b) = 1, maka terdapat bilangan bulat m dan n sedemikian hingga ma + nb = 1. Disebut bilangan prima, jika pembaginya hanya 1 dan bilangan itu sendiri. Contoh angka 13 habis dibagi oleh 1 dan 13 (Firmansyah, 2015). Teori ini merupakan hal yang mendasar untuk memahami algoritma kriptografi (Qorny, 2018). Dua buah bilangan bulat dan dikatakan relative prima. Jika FPB/GCD (x,) = 1.

Contohnya 20 dan 3 relatif prima sebab FPB (20, 3) = 1. Begitu juga 7 dan 11 relatif prima karena FPB (7, 11) = 1. Tetapi 20 dan 5 tidak relatif prima sebab FPB (20, 5) = 5 dan 1. Jika x dan relatif prima, maka terdapat bilangan bulat sehingga: x + n dan n sedemikian = 1.

Contohnya Bilangan 20 dan 3 adalah relatif prima karena FPB (20, 3) = 1, atau dapat ditulis: $2 \cdot 20 + (-13) \cdot 3 = 1$, dengan = 2 dan n = – 13. Tetapi 20 dan 5 tidak relatif prima karena FPB (20, 5) = $5 \neq 1$ sehingga 20 dan 5 tidak dapat dinyatakan dalam $20 + n \cdot 5 = 1$.

7 Modulus

Dalam matematika dan pemrograman komputer, operasi modulus adalah sebuah operasi yang menghasilkan sisa pembagian dari suatu bilangan terhadap bilangan lainnya. Dalam bahasa pemrograman operasi ini umumnya dilambangkan dengan simbol %, mod atau modulo, tergantung bahasa pemrograman yang digunakan.

Pada penelitian ini modulus digunakan karena operasi ini memiliki atau berhubungan dengan bilangan yang prima berdasarkan pada penilitian (Serdano dkk., 2019)

8 GCD

Greatest Common Divisor (GCD) atau sehari – hari kita sebut dengan Faktor Persekutuan Terbesar yaitu bilangan bulat N yang paling besar yang habis membagi dua buah bilangan bulat (Harahap, 2019). Misalnya dua buah bilangan bulat 12 dan 8.

12 habis dibagi oleh: 1, 2, 3, 4, 6, 12.

8 habis dibagi oleh: 1, 2, 4, 8.

Berdasarkan pembagian di atas maka dapat disimpulkan bahwa GCD dari 12 dan 8 adalah 4.

Contoh lainya:

1. GCD (24, 12) =12 (Artinya 12 merupakan bilangan terbesar yang membagi 24 dan 12)
2. GCD (24, 9)= 3 (Artinya 3 merupakan bilangan terbesar yang membagi 24 dan 9)

Cara yang digunakan pada penelitian ini dalam menemukan GCD atau dalam metode ini, bilangan bulat yang lebih kecil dikurangi dari bilangan bulat yang lebih besar, dan hasilnya diberikan ke variabel yang memiliki bilangan bulat yang lebih besar. Mencari GCD memiliki berbagai macam teknik dan berdasarkan konsep yang dipilih tidak menjadi masalah untuk menerapkan konsep salah satunya.

2.3 Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu *"cryptos"* yang berarti rahasia dan *"graphein"* yang berarti tulisan. Dapat dikatakan kriptografi berarti suatu ilmu yang mempelajari data secara rahasia dengan teknik matematika tertentu.

Kriptografi adalah ilmu mengenai teknik enkripsi teks asli (*plaintext*) diubah menggunakan suatu kunci enkripsi menjadi teks acak yang sulit dibaca (*ciphertext*) dan hanya seseorang yang memiliki kunci dekripsi mudah membaca.

Salah satu implementasi kriptografi asimetris adalah Rivest Shamir Adleman (RSA). Langkah-langkah (yang diteliti yaitu pada no 1 – 3 tepatnya konstanta *p* dan *q*) untuk membangkitkan kunci RSA adalah (Nisha & Farik, 2017):

1. Menentukan nilai prima sebagai p dan q. Nilai kedua bilangan prima tersebut dianjurkan ($p \neq q$). (Zulfikar dkk., 2019) Sebaiknya bilangan yang besar agar tingkat keamanannya juga meningkat, rekomendasi prima adalah 100 digit (desimal), sehinga n mempunyai 200 digit lebih (Wulansari dkk., 2016).
2. Mencari nilai n dengan memanfaatkan persamaan 2.1.

$$n = p * q \quad \text{(2.1)}$$

3. Mencari nilai ekuivalen dengan persamaan 2.2.

$\phi(n) = (p-1) * (q-1)$ ……...……...…… (2.2)

Rekomendasi $Gcd(p-1, q-1)$ semakin besar maka semakin cepat pemfaktoran dan sebaliknya maka semakin lama (Muchlis dkk., 2017).

Berdasarkan penilitian ini memberikan gagasan atau ide dalam eksperimen yang ditujukan pada pembangkitan kunci p dan q berdasarkan infomasi peranti yaitu waktu.

2.4 Informasi Peranti

Informasi peranti adalah komponen perangkat lunak yang mengizinkan sebuah sistem komputer untuk berkomunikasi dengan sebuah perangkat keras. Data peranti memiliki cakupan luas, salah satu di antaranya adalah:

1. Waktu (meliputi: 12 atau 24 jam format dan zona waktu).
2. Sinyal (terdiri dari jangkauan area, tegangan, arus dan lainnya).
3. Suhu (skala: Celsius, Kelvin, Fahrenheit, dan Reamur).
4. Baterai (voltase, daya atau persen dan lainnya).

Baterai adalah alat elektro kimia yang berfungsi untuk menyimpan tenaga listrik dalam bentuk tenaga kimia. Tenaga listrik yang tersimpan akan dialirkan untuk memberikan arus listrik. Daya baterai biasanya bernilai 1 – 100% yang terlihat pada ponsel contohnya.

Suhu adalah suatu besaran yang menunjukan derajat panas khususnya pada benda. Benda yang mempunyai panas maupun dingin, pada umumnya ponsel dilengkapi indikator derajat.

Sinyal adalah suatu besaran fisis yang berubah terhadap waktu, ruang, ataupun dapat berubah terhadap variabel bebas lainnya. Ponsel harus memiliki sebuah sinyal ketika melakukan komunikasi.

Waktu atau masa menurut Kamus Besar Bahasa Indonesia adalah seluruh rangkaian saat ketika proses, perbuatan, atau keadaan berada atau berlangsung. Dalam hal ini, skala waktu merupakan interval antara dua buah keadaan/kejadian, atau bisa merupakan lama berlangsungnya suatu kejadian

Pada penilitian ini menggunakan waktu, sebagai eksperimen dan memanfaatkan data jam menit dan detik yang menjadi 3 variabel fokus untuk penentuan p dan q.

2.5 *Kotlin* dan Aliran Kontrol

Kotlin adalah sebuah bahasa pemrograman dengan pengetikan statis yang berjalan pada Mesin Virtual Java ataupun menggunakan kompiler LLVM yang dapat pula dikompilasikan kedalam bentuk kode sumber JavaScript. Pengembang utamanya berasal dari tim developer dari JetBrains yang bermarkas di Rusia (*FAQ - Kotlin Programming Language*, n.d.).

Pada penelitian ini operasi dan pengujian menggunakan bahasa *kotlin* yang dikompilasi atau dijalankan oleh *mobile android*. Beberapa aliran kontrolnya terdapat *if*, *when*, *for* dan *while*. Semua aliran digunakan menangani aritmatika dan konsep yang dibuat sesuai kebutuhan.

2.6 *Exception Handling*

Memeriksa semua kemungkinan kesalahan atau yang disebut *Exception Handling* untuk setiap metode karena ini dapat membuat kode tidak dapat dipahami jika setiap pemanggilan metode memeriksa semua kemungkinan kesalahan sebelum menjalankan pernyataan berikutnya. Kelas *Throw* mampu menangani seluruh konsep pengecualian dan kesalahan. Tujuan utama dari mekanisme penanganan *Exception* adalah mendeteksi dan melaporkan "keadaan pengecualian" sehingga tindakan yang sesuai dapat diambil (Kumari dkk., 2015). Mekanisme tersebut menyarankan penggabungan kode penanganan kesalahan terpisah yang melakukan tugas berikut:

1. Menemukan masalah yaitu *Exception*
2. Menginformasikan bahwa telah terjadi kesalahan yaitu Pengecualian
3. Menerima informasi kesalahan yaitu Menangkap pengecualian
4. Ambil tindakan korektif, yaitu Menangani pengecualian Melempar

2.7 Shannon Entropy

Entropi merupakan konsep dasar yang dikemumakan pada teori informasi Shannon, ide ini diadopsi dari salah satu cabang ilmu fisika yaitu termodinamika. Dalam hal ini Entropi (H) digunakan untuk mengukur keacakan data (Rihartanto dkk., 2020) dimana terdapat suatu keadaan yang tidak dapat dipastikan kemungkinannya. Entropi timbul jika prediktabilitas/kemungkinan rendah (*low predictable*) dan informasi yang ada (*high information*). Entropi dihitung menggunakan formula entropi Shannon yang ditunjukkan pada Persamaan 2.6.1. Nilai entropi tertinggi yang dapat dicapai pada sebuah citra adalah 8, sementara

pada teks yang hanya menggunakan ASCII standar entropi tertinggi yang mungkin diperoleh adalah 7. Semakin tinggi nilai entropi menunjukkan tingkat keacakan yang semakin tinggi.

$$H(x) = -\sum_{i=1}^{n} P(x_i) log2\ P(x_i)$$

Dalam penelitian ini, entropi dihitung berdasarkan besarnya peluang jarak antar karakter dalam teks. Hasil entropi kemudian dibandingkan antara *p* dan *q* yang ditentukan secara *default* dengan *p* dan *q* yang ditentukan berdasarkan informasi peranti waktu.