**PEMBANGKITAN KUNCI YANG DIGUNAKAN UNTUK PENENTUAN**

**KONSTANTA P DAN Q YANG PRIMA BERDASARKAN**

**INFORMASI PERANTI**

**LAPORAN TUGAS AKHIR**



**Oleh:**

**YOGI ARIF WIDODO**

**NIM. 17 615 006**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN**

**POLITEKNIK NEGERI SAMARINDA**

**JURUSAN TEKNOLOGI INFORMASI**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**2020**

Kata Pengantar

Puji syukur Alhamdulillah panjatkan kehadirat Allah SWT yang telah melimpahkan rahmatnya serta hidayahnya sehingga mampu menyelesaikan Proposal Tugas Akhir dengan judul Pembangkitan Kunci Privat Pada Enkripsi RSA Menggunakan Infromasi Peranti.

Selawat Salam semoga selalu tercurahkan kepada Nabi Muhammad SAW Beserta keluarga dan para sahabatnya hingga pada umatnya sampai akhir zaman.

Proposal Tugas Akhir ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan jenjang pendidikan program Diploma III di Jurusan Teknologi Informasi, Politeknik Negeri Samarinda.

Dalam proses penyusunan Proposal Tugas Akhir ini, mendapatkan banyak sekali bantuan, bimbingan serta dukungan dari berbagai pihak, sehingga dalam kesempatan ini, bermaksud menyampaikan rasa terima kasih kepada:

1. Kedua orang tua dan keluarga yang selalu memberi dukungan moral dan materi.
2. Ansar Rizal, ST., M.Kom. selaku Ketua Jurusan Teknologi Informasi Politeknik Negeri Samarinda
3. Mulyanto, S.Kom., M.Cs. selaku promotor yang telah membimbing hingga terselesaikannya proposal tugas akhir ini.
4. Staf dosen, staf teknisi, dan staf administrasi jurusan yang telah membantu dalam segala hal yang berkaitan dengan perkuliahan.
5. Semua sahabat dan rekan-rekan mahasiswa jurusan Teknologi Informasi yang ikut memberi saran dan masukan.
6. Serta semua pihak lain yang ikut terlibat dalam penyelsaian Proposal Tugas Akhir ini

Semoga Allah SWT memberi balasan yang setimpal kepada semuanya.

Harapannya tugas akhir yang telah disusun ini bisa memberikan sumbangsih untuk menambah pengetahuan, dan perbaikan selanjutnya, selalu terbuka terhadap saran dan masukan, karena menyadari tugas akhir yang telah disusun ini memiliki banyak sekali kekurangan.

Samarinda, 10 Juli 2020

Yogi Arif Widodo

DAFTAR ISI

[Kata Pengantar i](#_Toc39905598)

[DAFTAR ISI iii](#_Toc39905599)

[DAFTAR GAMBAR v](#_Toc39905600)

[DAFTAR TABEL vii](#_Toc39905601)

[BAB I PENDAHULUAN 1](#_Toc39905602)

[1.1 Latar Belakang 1](#_Toc39905603)

[1.2 Rumusan Masalah 2](#_Toc39905604)

[1.3 Tujuan Penelitian 2](#_Toc39905605)

[1.4 Batasan Masalah 2](#_Toc39905606)

[1.5 Manfaat Penelitian 3](#_Toc39905607)

[BAB II TINJAUAN PUSTAKA 4](#_Toc39905608)

[2.1 Kajian Ilmiah 4](#_Toc39905609)

[2.2 Dasar Teori 5](#_Toc39905610)

[2.2.1 Kriptografi 5](#_Toc39905611)

[2.2.2 Informasi Peranti 10](#_Toc39905612)

[2.3.1 Teori Bilangan ( Relatif Prima ) 11](#_Toc39905613)

[2.3.2 Entropi dan Matrik 11](#_Toc39905614)

[BAB III METODE PENELITIAN 13](#_Toc39905615)

[3.1 Kerangka Konsep Penelitian 13](#_Toc39905616)

[3.1.1 Kriptografi 14](#_Toc39905617)

[3.2 Metodologi Penelitian 15](#_Toc39905618)

[3.2.1 Riset Awal 16](#_Toc39905619)

[3.2.2 Tahapan Menentukan Bilangan Prima 16](#_Toc39905620)

[3.2.3 Tahapan Pembangkitan Kunci 18](#_Toc39905621)

[3.2.4 Pengujian 19](#_Toc39905622)

[3.2.5 Analisa Hasil 19](#_Toc39905623)

[3.2.6 Variabel Penelitian 19](#_Toc39905624)

[3.2.7 Waktu dan Tempat Penelitian 19](#_Toc39905625)

[BAB IV HASIL DAN PEMBAHASAN 20](#_Toc39905626)

[4.1 Hasil Tahapan Menentukan Bilangan Prima 20](#_Toc39905627)

[4.1.1 Menentukan Batasan Angka Prima Sampai Jumlah Suatu *Char* Suatu Kata 20](#_Toc39905628)

[4.1.2 Menghitung Jumlah Angka Prima 22](#_Toc39905629)

[4.1.3 Menggunakan Informasi Peranti (Waktu) 23](#_Toc39905630)

[4.1.3.1 Pseudorandom 24](#_Toc39905631)

[4.1.5 Menentukan 2 Bilangan Prima (p dan q) 25](#_Toc39905632)

[4.2 Hasil Tahapan Pembangkitan Kunci 26](#_Toc39905633)

[4.2.1 Membangkitkan Kunci Publik dan Kunci Privat 26](#_Toc39905634)

[4.3 Pengujian 28](#_Toc39905635)

[4.3.1 Mengunci dan Membuka Teks 28](#_Toc39905636)

[4.3.1.1 Pengujian Pertama 28](#_Toc39905637)

[4.3.1.2 Pengujian Kedua 31](#_Toc39905638)

[4.3.2 Uji Pembangkitan Kunci 33](#_Toc39905639)

[4.4 Analisa Hasil 34](#_Toc39905640)

[4.4.1 Analisa Hasil Pengujian Pertama 34](#_Toc39905641)

[4.4.2 Analisa Hasil Pengujian Kedua 35](#_Toc39905642)

[BAB V PENUTUP 38](#_Toc39905643)

[5.1 Kesimpulan 38](#_Toc39905644)

[5.2 Saran 39](#_Toc39905645)

[RENCANA JADWAL PENGERJAAN 40](#_Toc39905646)

[DATAR PUSTAKA 41](#_Toc39905647)

DAFTAR GAMBAR

|  |  |
| --- | --- |
| Gambar 2.1 Teknik *Blocking*…………..…………………………...………. | 7 |
| Gambar 2.2 Teknik Pemampatan…………………………………...……… | 8 |
| Gambar 2.3 Teknik Permutasi……………..…….………………...……….. | 9 |
| Gambar 2.4 *FlowChart* Pembangkitan Kunci Algoritma RSA…....……….. | 10 |
| Gambar 3.1. Diagram Alir Kerangka Konsep Penelitian……….………….. | 12 |
| Gambar 3.2. Diagram Alir Metodologi Penelitian……………..……….….. | 14 |
| Gambar 3.2.2 *FlowChart* Proses Pembangkit Batas Atas………………….. | 17 |
| Gambar 3.2.3 *FlowChart* Proses Hasil Pembangkit Semua Angka Prima..... | 17 |
| Gambar 3.2.4 *FlowChart* Proses Terpilihnya konstanta atau orde P dan Q. | 16 |
| Gambar 3.2.5 *FlowChart* Proses Pembangkitan Kunci dengan Informasi Peranti……………………………………………………………………..... | 17 |
| Gambar 4.1.1.1 Hasil *JUnit* *Testing* Pengecekan Batas Atas……………..... | 21 |
| Gambar 4.1.1.2 Ilustrasi Hasil Pembangkitan Bilangan Atas……………… | 21 |
| Gambar 4.1.1.3 *FlowChart* Program Pembangkit Batas Atas……………… | 22 |
| Gambar 4.1.2.1 *FlowChart* Program Hasil Pembangkit Semua Angka Prima………………………………………………………………………... | 23 |
| Gambar 4.1.3.1 Daftar Waktu Indonesia Tengah………………………… | 24 |
| Gambar 4.1.4.1 Proses *Pseudorandom* Zona Waktu……………………….. | 24 |
| Gambar 4.1.5.1 *FlowChart* Program Terpilihnya konstanta atau orde P dan Q…………………………………………………………………………….. | 25 |
| Gambar 4.2.1 Hasil Pembangkitan Kunci Pada Peranti *Android*…………... | 27 |
| Gambar 4.3.1.5 Hasil Pengujian Pertama Dekripsi *PlainText*…………….... | 31 |
| Gambar 4.3.2.2 Hasil Pengujian Kedua Mengalami Null………………….. | 32 |
| Gambar 4.4.1.1 Analisa Hasil Probabilitas ASCII dan *CipherText*……….... | 34 |
| Gambar 4.4.2.1 Analisa Hasil Entropi PRNG Zona Waktu Dalam 5 Menit..... | 36 |

DAFTAR TABEL

|  |  |
| --- | --- |
| Tabel 4.3.1.1 Hasil Pengujian Pertama Enkripsi dan Dekripsi……………... | 28 |
| Tabel 4.3.1.2 Hasil Kode ASCII *PlainText*……………………………...…. | 29 |
| Tabel 4.3.1.3 Hasil Enkripsi atau *ChiperText*………………………………. | 29 |
| Tabel 4.3.1.4 Hasil Probabilitas *Binary ChiperText*………………………... | 30 |
| Tabel 4.3.2.1 Hasil Pengujian Kedua Pada orde P dan Q…………………... | 31 |
| Tabel 4.3.2 Uji Pembangkitan Kunci Pertama……………………………... | 33 |
| Tabel 4.4.1.2 Analisa Hasil Jarak Rentang Nilai P dan Q………………….. | 35 |

NOTASI

BAB I  
PENDAHULUAN

1.1 Latar Belakang

Bilangan prima adalah bilangan yang hanya memiliki dua faktor: 1 dan bilangan itu sendiri. Satu-satunya bilangan prima bernilai genap hanyalah 2 (Cahyo Dhea Arokhman Yusufi, 2020). Kemudian akan muncul pertanyaan mengenai apakah 1 bilangan prima? tentu saja tidak. 1 hanya memiliki 1 faktor pembagi. Kita tidak menghitung 1 sebanyak dua kali. ∀n∈N,n>1, maka n selalu memiliki setidaknya 1 faktor prima. Setiap bilangan asli lebih dari 1 yang tidak prima disebut bilangan komposit. Jika n adalah suatu bilangan komposit (Harahap, 2019), maka n memiliki setidaknya 1 faktor prima yang nilainya tidak lebih dari √n.

Bilangan prima > 3 memiliki keunikan yang selalu berbentuk antara (Chiewchanchairat dkk., 2016) 6k-1 atau 6k +1 (Ferreira, 2017). Setiap bilangan hanya memiliki 6 bentuk: 6k - 2,6k - 1,6k, 6k+1 ,6k + 2,6k +3. Tapi perhatikan bahwa 6k -2,6k, 6k +2 selalu genap. Sedangkan 6k +3 adalah kelipatan 3. Maka dari itu bilangan prima yang lebih dari 3 akan selalu memiliki antara dua bentuk tadi. Hasil selanjutnya yang didapat mengenai bilangan prima adalah bahwa bilangan prima ada tak hingga banyaknya (Sciences, 2016) (Meštrović, 2018). Hal ini mungkin terkesan sangat jelas tapi tidak semua orang bisa membuktikan pernyataan ini.

Bilangan Prima merupakan bilangan bulat positif, sifat pembagiannya (Firmansyah, 2015) melahirkan konsep-konsep aritmetika modulo, dan salah satu konsep bilangan bulat yang digunakan dalam penghitungan komputer. Dengan ditemukannya bilangan prima, teori bilangan berkembang semakin jauh dan lebih mendalam. Banyak dalil dan sifat dikembangkan berdasarkan bilangan prima.

Pada penelitian (Sari, 2017) bilangan prima merupakan bilangan istimewa dalam Al-Qur’an karena definisi bilangan prima yaitu bilangan yang tidak bisa dibagi dengan bilangan lain kecuali satu dan bilangan itu sendiri yang menampilkan sifat Allah yang tidak dapat dibagi dengan siapapun kecuali diri-Nya.

Berdasarkan penilitian (Kumari dkk., 2015) Pengecualian atau Exception Handling merupakan cara bersih memeriksa kesalahan tanpa mengacaukan kode dan mampu menangkap pengecualian sebuah aritmatika salah satunya NumberFormatException. Klausa tangkapan diikuti blok coba (try and catch), setiap blok tangkapan merupakan pengecualian yangmenangani jenis pengecualian, Pengecualian ini cocok digunakan pada Pengujian dan Pembuktian Terhadap proses Tahapan Penentuan Konstanta P dan Q berdasarkan Informasi Peranti.

Berdasarkan sifat bilangan prima maka pada penilitian ini menggabungkan informasi peranti waktu pada android mobile dan hingga membuktikan bilangan prima yang di dapat yaitu konstanta *p* dan *q.*

1.2 Rumusan Masalah

Dalam melaksanakan penelitian, masalah yang menjadi poin utama diskusi atau pembahasan, adalah “Bagaimana Melakukan Pembangkitan Kunci Yang Digunakan Untuk Penentuan Konstanta P dan Q Yang Prima Berdasarkan Informasi Peranti”.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Memanfaatkan Informasi Peranti Waktu ( jam, menit dan detik )
2. Memodifikasi Teknik Pembangkitan Kunci Yang Digunakan Untuk Penentuan Konstanta P dan Q Yang Prima

1.4 Batasan Masalah

Agar persepsi penilitian tepat dan sesuai rumusan masalah, memerlukan batasan masalah sebagai berikut:

1. Informasi peranti menggunakan waktu jam, menit dan detik.
2. Waktu yang dipakai adalah sekarang
3. Zona waktu adalah **GMT -11:00** sampai **GMT +13:00**.
4. Panjang kunci *p* dan *q* adalah 7 *bit* (2 digit) sampai 14 bit (4 digit).

1.5 Manfaat Penelitian

Harapan penelitian yang dilaksanakan, dapat memberikan manfaat:

1. Kunci konstanta *p* dan *q* memiliki pola tambahan berdasarkan informasi waktu jam, menit dan detik dari peranti android.
2. Dapat menjadi sumber referensi bagi pihak lain dalam menyusun karya ilmiah maupun penelitian yang berkaitan dengan judul pada penelitian.

BAB II  
TINJAUAN PUSTAKA

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

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 (Muhsetyo, 1997:1). Teori bilangan merupakan dasar perhitungan dan menjadi salah satu teori yang mendasari pemahaman kriptografi, khususnya sistem kriptografi kunci publik. Bilangan yang dimaksud hanyalah bilangan bulat (integer).

2.2.2 Bilangan Bulat

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 Ι 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 − (Abdussakir, 2009:102). Himpunan bilangan bulat dilengkapi dengan dua buah operasi, yaitu operasi penjumlahan dan perkalian, dilambangkan (Z, +,·) membentuk suatu sistem matematika yang disebut gelanggang atau ring (Abdussakir, 2009:102). Himpunan bilangan bulat berperan sangat penting dalam kriptografi karena banyak algoritma kriptografi yang menggunakan sifat-sifat himpunan bilangan bulat dalam melakukan proses penyandiannya.

2.2.3 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 (Muhsetyo, 1997:43).

2.2.4 Algoritma Pembagian

Jika a, b ∈ Z dan a > 0, maka ada bilangan q, r ∈ Z yang masing-masing tunggal sehingga b = qa + r dengan 0 r < a. Jika a∤b, maka r memenuhi ketidaksamaan 0 < r < a (Muhsetyo, 1997:50).

2.2.5 Fungsi Euler (φ)

Fungsi Euler digunakan untuk menyatakan banyaknya bilangan bulat < n yang relatif prima terhadap n.

Definisi 2.4

Suatu himpunan bilangan bulat \*r 1, r 2 , ... , r k + disebut dengan sistem residu tereduksi modulo

a) (r , ,

b) r jika: ) = 1( = 1, 2, ... , ). r (mod m) untuk semua ≠ .

c) Jika (x, ) = 1, maka x r mod ( ) (Muhsetyo, 1997:279).

Contoh:

Himpunan \*1,5+ adalah sistem tereduksi modulo 6 karena:

a. r 1 = 1, r 2 = 5, (r 1 , 6) = ( 1,6 ) = 1 dan (r 2 , 6) = (5,6) = 1

b. 1

5 (mod 6)18

c. (7,6) = 1

7

d. (11,6) = 1

11

1(mod 6)

5(mod 6), dan seterusnya

Teorema 2.4

Jika p adalah suatu bilangan prima, maka φ(p) = p – 1 (Muhsetyo,

1997:280).

Bukti:

Karena p adalah bilangan prima, maka setiap bilangan bulat positif kurang

dari p relatife prima terhadap p. Ini berarti bahwa sistem residu tereduksi modulo

padalah himpunan {1, 2, 3,...,p - 1} yang mana seluruh anggotanya sebanyak (p -

1) sehingga φ(p) = p – 1.

2.1.4 Aritmetika Modulo dan Kekongruenan

Definisi 2.6

Diketahui a, b,

a

b(mod

∈ Z. a disebut kongruen dengan b modulo

), jika (a − b) habis dibagi

b)tidak habis dibagi

, yaitu

, yaitu

, ditulis

|(a − b). Jika (a −

∤(a − b), maka ditulis a

b(mod

),

dibaca a tidak kongruen dengan b modulo m. Karena (a − b) habis dibagi

oleh jika dan hanya jika (a − b) habis dibagi oleh –

b(mod ) jika dan hanya jika b

a(mod

, maka: a

) (Muhsetyo, 1997:138).

Contoh:

1. 17 2(mod 3) (3 habis dibagi 17 − 2 = 15

15

3 = 5)

2. −7 15(mod 3) (3 tidak habis dibagi −7 − 15 = −22)22

2.2.6 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 (Munir, 2012:200). 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.

Definisi 2.8

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.

2.2.7 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 (Respatiadi, 2013).

Contoh:

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.

Contoh:

Bilangan 20 dan 3 adalah relatif prima karena FPB (20, 3) = 1, atau dapat

ditulis: 2·20 + (– 13)· 3 = 1, dengan = 2 dan n = – 13. Tetapi 20 dan 5

tidak relatif prima karena FPB (20, 5) = 5 ≠ 1 sehingga 20 dan 5 tidak dapat

dinyatakan dalam . 20 + n · 5 = 1.

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):

Menentukan nilai prima sebagai p dan q. Nilai kedua bilangan prima tersebut dianjurkan (p 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).

Mencari nilai n dengan memanfaatkan persamaan 2.1.  
……………….……………….………………….……… (2.1)

Mencari nilai ekuivalen dengan persamaan 2.2.

…………………………....…………… (2.2)

Rekomendasi semakin besar maka semakin cepat pemfaktoran dan sebaliknya maka semakin lama (Muchlis dkk., 2017).

Memilih bilangan prima secara random antara 1 sampai untuk mendapatkan kunci publik e.

Menghitung kunci privat d dengan persamaan 2.3.

…………………………………………… (2.3)

Pasangan kunci yaitu kunci publik (e, n) dan kunci privat (d, n) telah dihasilkan.

Untuk enkripsi dan dekripsi .

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).

BAB III  
METODE PENELITIAN

3.1 Kerangka Konsep Penelitian

Kerangka konsep penelitian (teori atau konsep ilmiah yang digunakan sebagai dasar penelitian) menjelaskan hubungan atau gabungan alur sebagai ruang lingkup penelitian dan ruang lingkup ilmu

  
Gambar 3.1 Diagram Alir Kerangka Konsep Penelitian

3.1.1 Bilangan

Dalam bilangan terdapat berbagai jenis bilangan dua diantaranya yang berhubungan dekat dengan penilitian konstanta *p* dan *q*, yaitu:

1. Bilangan Komposit
2. Bilangan Prima

Macam-macam solusi yang mengenai bilangan prima dapat diolah seperti mengecek bilangan itu sendiri, diantaranya adalah:

1. *Naive Solution*
2. *Fermet*
3. *Miller-Rabbin*
4. *Solovay-Strassen*
5. *Lucas*

Dari banyaknya bilangan prima, menuangkan atau menentukan 2 buah nilai konstanta ini bisa ditentukan dengan sedemikian rupa dengan kombinasi sebuah angka yang didapat dari informasi peranti dan diantaranya yaitu waktu berupa jam, menit dan detik.

3.2 Metodologi Penelitian



Gambar 3.2 Diagram Alir Metodologi Penelitian

3.2.1 Riset Awal

Sebelum melakukan penelitian terlebih dahulu mempelajari hal yang terkait dengan topik penelitian. Bagian utama yang perlu dipelajari adalah:

1. Mengetahui fungsi konstanta *p* dan *q* yang prima
2. Mengetahui penggunaan informasi peranti
3. Landasan matematika (teori bilangan dan pemfaktoran bilangan bulat)

3.2.2 Tahapan Menentukan Bilangan Prima

Tahapan ini memiliki 2 langkah yakni Menentukan Batasan Angka Atas Prima Sampai Jumlah Suatu Char dan Membangkitkan Bilangan Prima.

Menentukan Batasan Angka Prima Sampai Jumlah Suatu Char dengan menggunakan persamaan 1.1 maka ilustrasinya dapat diperlihat seperti pada Tabel 1.

Tabel 1 Hasil Karakter ke *ASCII*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *char* | P | o | … | n |
| *ASCII* | 80 | 111 | … | n |

..………….….…………………………………(1.1)

Dimana :

Total = Batas Atas Prima

Ui = Nilai Karakter Pada ASCII

Membangkitkan Bilangan Prima dengan mengeliminasi angka bukan prima (TH & MB, 2017). Penerapanya sederhana dilakukan dengan *naive solution* sebagai berikut:

1. Ketika Melalui semua angka dari 2 ke *n*-1, maka setiap nomor periksa apakah ia membagi *n*.

2. Jika ditemukan angka yang dibagi, akan mengembalikan tanda *false*

3. Sebaliknya *true* dan simpan nilai *n* ke dalam *arrayListPrimeNumber*.



Gambar 4 FlowChart Naive Solution

Nilai *n* telah ditentukan pada tahap sebelumnya yaitu *n* = batas atas prima.

3.2.3 Tahapan Mendapatkan Informasi Peranti

Informasi Peranti yang didapatkan memiliki 3 variabe yaitu jam, menit, dan detik. Proses mendapatkannya dibaca oleh peranti *Mobile* *Android* dengan fungsi yang sudah tersedia di kotlin yang diperlihatkan pada Gambar 4. Maka data waktu yang didapat masih berupa nilai yaitu 1594886148236 keseluruhan waktu yang kemudian diformat untuk menjadikanya jam, menit dan detik

Dengan fungsi yang sudah tersedia di kotlin, dapat digunakan *syntax* sebagai berikut :

Maka akan didapatkan waktu sekarang sebagai Informasi Peranti dengan zona awal yang didapat GMT +8

3.2.3 Tahapan Mengolah Informasi Peranti

Informasi Peranti diolah kembali untuk menghasilkan informasi peranti yang probabilstik berdasarkan waktu jam, menit dan detik menggunakan *Greenwich Mean Time Zone* (GMT).

Gambar 4.1.4.1 Proses *Pseudorandom* Zona Waktu

Seluruh zona waktu telah didefinisikan sebelumnya ke dalam arrayTime sebagai zona lain.

Tabel 2 Daftar Waktu Indonesia Tengah



Pemilihan posisi atau index untuk arrayTime berdasarkan keluaran dari nilai *integer* oleh *sudorandom*, sebagai zona lain.

Dengan fungsi yang sudah tersedia di kotlin, dapat digunakan syntax sebagai berikut :

Maka hasil nilai sudoRandom = x. Sehingga didapat arrayTime [sudoRandom] = GMT -+x . Kemudian dilakukan konversi waktu sekarang xx:xx:xx GMT +8 ke GMT -+x. Informasi yang digunakan adalah zona lain, perubahan zona sendiri merupakan proses, tujuanya mengkonsumsi sebuah waktu ketika mendapatkan informasi waktu itu sendiri.

3.2.3 Penentuan Konstanta P dan Q Berdasarkan Informasi Peranti

Penentuan telah dilakukan dengan melihat syarat sebagai berikut:

1. Bilangan yang prima telah didapatkan dalam bentuk arrayListPrimeNumber hasilnya berdasarkan pada Tahapan Membangkitkan Bilangan .

2. Informasi Peranti telah didapatkan dalam bentuk bagian dari waktu jam, menit dan detik. Hasilnya diperlihatkan pada Gambar 6.

Kemudian menentukan p dan q dimana arrayListPrimeNumber = p = q untuk menghasilkan prima yang deterministik dari informasi peranti yang probabilistik.

a. Menentukan Konstanta P yang Prima , Penentuan ini sederhana, dengan menghitung persamaan 1.1 didapat indexP = 36.

(………………………………………………………………1.1)

hh \*n=indexP

Dimana :

hh = informasi peranti waktu jam

n = 4

Maka didapat nilai p[indexP] = 157.

Jika n memiliki nilai yang lebih besar dari 2 , misal 3 maka memiliki tujuan terbentuknya p yang prima cukup besar.

Dengan p yang besar, memiliki kesempatan Greatest Common Divisor GCD(p, q) yang hasilnya kecil atau proses pemfaktoran yang memakan waktu.

b. Menentukan Konstanta Q yang Prima nilai q memiliki aturan mirip dengan nilai p, tetapi memiliki 5 keputusan perhitungan () dari 6 ketentuanya ().

()…………………………………………………………..…(2.1)

K1 = p

K2 = informasi peranti waktu menit

K3 = informasi peranti waktu detik

K4 = K2 + K3

K5 = K1 \* K2

K6 = K2 \* K3

()…………………………………………………………….(2.2)

Dimana :

K[n] = arrayListPrimeNumber [n]

jml prima = arrayListPrimeNumber.size

3.2.4 Pengujian dan Pembuktian

Hasil kombinasi konstanta p dan q, dalam pembangkitan prima berdasarkan informasi waktu peranti jam, menit dan detik, diuji dengan monitoring *Exception* *Handling* atau pengecualian memilih konstanta p dan q.

3.2.5 Analisa Hasil

Hasil yang diperoleh dari pengujian kemudian dianalisa terutama pada proses terpilihnya p dan q untuk pembangkitan kunci.

3.2.6 Variabel Penelitian

Fokus penelitian tugas akhir ini dituangkan dalam variabel yaitu Modifikasi konstanta atau orde p dan q berdasarkan waktu informasi peranti yaitu jam , menit dan detik.

3.2.7 Waktu dan Tempat Penelitian

Penelitian dilaksanakan bulan Desember 2019 sampai bulan Februari 2020 di Politeknik Negeri Samarinda.

BAB IV  
HASIL DAN PEMBAHASAN

4.1 Hasil Tahapan Menentukan Bilangan Prima

Sebelum mengkombinasikan waktu peranti, dan mengolahnya menjadi lebih berpola dalam pembangkitan kunci *p* dan *q*, bilangan prima yang digunakan, ditentukan sedemikian rupa oleh jumlah karakter dari suatu kata melalui proses input, sehingga cukup panjang untuk memfaktorkannya. jika hasil tersebut semakin tinggi maka proses eliminasi menambah sekian detik waktu. Hal tersebut juga menghasilkan angka-angka yang berbeda di setiap variabel (pada semua bilangan tanpa batasan). Alur menentukan bilangannya, di atur dengan proses berikut:

4.1.1 Menentukan Batasan Angka Prima Sampai Jumlah Suatu *Char*

Pembatasan Bilangan Prima dimaksudkan menjaga ruang memori atau proses dalam menentukan bilangan normal ke prima (eliminasi angka bukan prima)*. American Standard Code for Information Interchange* (ASCII) digunakan dalam masukan batasan.

a. Menentukan Batasan Angka Prima Sampai Jumlah Suatu *Char*, Misalnya dari kalimat “Politeknik Negeri Samarinda Tahun 2020” Diuraikan menjadi kode *ASCII* yang diperlihatkan pada Tabel 1.

Tabel 1 Hasil Karakter ke *ASCII*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *char* | P | o | … | n |
| *ASCII* | 80 | 111 | … | n |

Kemudian dengan persamaan 1.1 didapat totalnya = 3400.

..………………………………………….…….….…(1.1)

Dimana :

Total = Batas Atas Prima

Ui = Nilai Karakter Pada *ASCII*

Matematikawan membuktikan bahwa bilangan prima terbesar itu tidak ada, bilangan prima ‘terbesar’ ditemukan, yaitu 277.232.917 – 1 yang diketahui pada Juli 2018 (*Untuk Apa Mencari Bilangan Prima Terbesar? - Anak Bertanya*, n.d.)diatasnya masih ada. Proses pembatasan prima mengkonsumsi sebuah waktu yang berhubungan dengan tahapan pengolahan informasi peranti yaitu jam, menit dan detik.

b. Membangkitkan Bilangan Prima dengan mengeliminasi angka bukan prima (TH & MB, 2017). Penerapanya sederhana dilakukan dengan *naive solution* sebagai berikut:

1. Ketika Melalui semua angka dari 2 ke *n*-1, maka setiap nomor periksa apakah ia membagi *n*.

2. Jika ditemukan angka yang dibagi, akan mengembalikan tanda *false*

3. Sebaliknya *true* dan simpan nilai *n* ke dalam *arrayListPrimeNumber*.

// this analisa hasil will move laters

Proses memasukan kalimat ASCII, memiliki aturan diatas batas nilai sekitar 2000 - 3000, bertujuan meluaskan rentang waktu berjalan pembangkitan angka prima menuju pada penggunaannya, sekitar 6.606 milidetik untuk contoh kalimat dan ilustrasi dijelaskan pada Gambar 4.1.1.2 Ilustrasi Hasil Pembangkitan Bilangan Atas, yang dilakukan uji coba dengan melihat waktu selesai *compiler*. Setiap tempo yang dihasilkan dipengaruhi oleh kondisi kecepatan peranti dalam memproses membaca program.

Gambar 4.1.1.2 Ilustrasi Hasil Pembangkitan Bilangan Atas.

Keseluruhan uji proses menghasilkan nilai yang logika (urut), tetapi saat peranti menjalankan banyak proses, menghasilkan rentang waktu yang berbeda. Logika yang berjalan dari Gambar 3.2.2 *FlowChart* Proses Pembangkit Batas Atas dan Gambar 4.1.1.2 Ilustrasi Hasil Pembangkitan Bilangan Atas, dimuat dalam *flowchart* program yang disajikan pada Gambar 4.1.1.3 *FlowChart* Program Pembangkit Batas Atas.

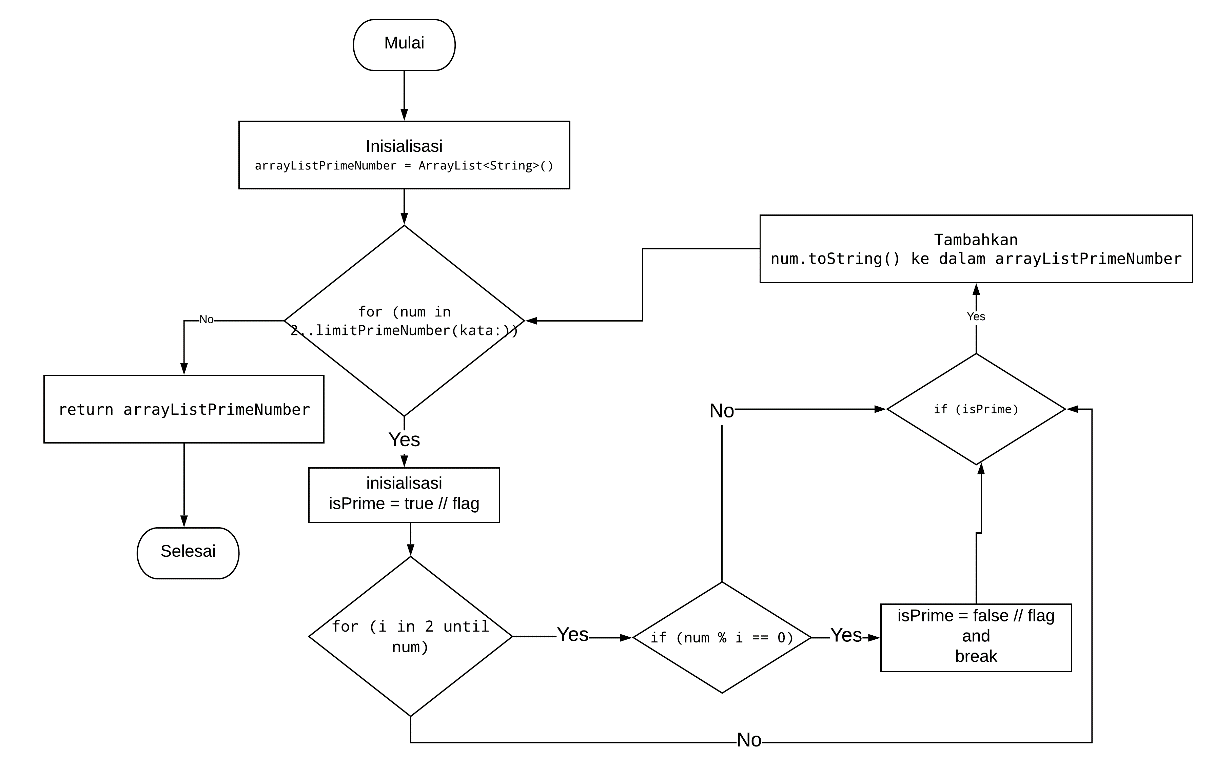


Gambar 4.1.1.3 *FlowChart* Program Pembangkit Batas Atas

4.1.2 Membangkitkan Bilangan Prima

Jumlah prima dihitung dengan mengeliminasi Angka Bukan Prima (TH & MB, 2017), yang dimaksud adalah mencari bilangan prima itu sendiri, dengan menyatakan proses menghasilkan bukan nol adalah benar dan sebaliknya adalah salah. Perhitungan rumus menggunakan sisa bagi, jika A = 3 dan nilai pembaginya (sisa bagi) B = 2, maka ditandai sebagai benar.

Pada pembatsan bilangan prima sebelumnya adalah 3400, dengan ketentuan sisa bagi, maka dihasilkan angka prima sebanyak 478, dan rentang waktu sekitar 16.908 milidetik. Waktu ini mempengaruhi penggunaan informasi peranti sebelum terbangkitnya bilangan p dan q, sehingga waktu sekarang adalah **14:05:30 GMT+8** dan didefinisikan sebagai *arrayListPrimeNumber*. Alur eliminasi sendiri memuat batas atas prima di dalam prosesnya, diperlihatkan pada Gambar 4.1.2.1 *FlowChart* Program Hasil Pembangkit Semua Angka Prima.

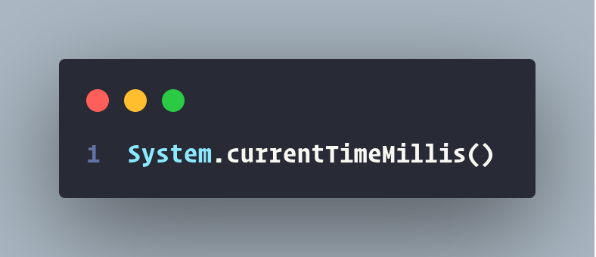


Gambar 4.1.2.1 *FlowChart* Program Hasil Pembangkit Semua Angka Prima

4.2 Tahapan Mendapatkan Informasi Peranti

Seluruh Zona waktu merupakan bagian dari informasi waktu yang digunakan. Greenwich Mean Time Zone (GMT) menyesuaikan waktu peranti dan hasil menentukan bilangan prima menggunakan Waktu Indonesia Tengah (WITA) dalam format **24 jam (HH:mm:ss)**.

Informasi Peranti yang didapatkan memiliki 3 variabe yaitu jam, menit, dan detik. Proses mendapatkannya dibaca oleh peranti *Mobile* *Android* dengan fungsi yang sudah tersedia di kotlin yang diperlihatkan pada Gambar 4.



Gambar 4. Potongan Kode Mendapatkan Informasi Peranti Waktu Sekarang

Data waktu yang didapat masih berupa nilai keseluruhan waktu yang kemudian diformat menjadi (HH:mm:ss) untuk menjadikanya jam, menit dan detik.

Dengan fungsi yang sudah tersedia di kotlin, dapat digunakan *syntax* sebagai berikut :

Maka didapatkan waktu sekarang 06:05:30 dengan zona awal GMT +8 sebagai Informasi Peranti.

4.3 Tahapan Mengolah Informasi Peranti

Informasi Peranti diolah kembali untuk menghasilkan informasi peranti yang probabilstik berdasarkan waktu jam, menit dan detik menggunakan *Greenwich Mean Time Zone* (GMT) sebagai pengubah. Seluruh zona waktu telah didefinisikan sebelumnya ke dalam *arrayTime* sebagai zona lain.



Gambar 4.1.3.1 Daftar Waktu Indonesia Tengah

Pemilihan posisi atau *index* untuk *arrayTime* berdasarkan keluaran dari nilai *integer* oleh *sudorandom*, sebagai zona lain.

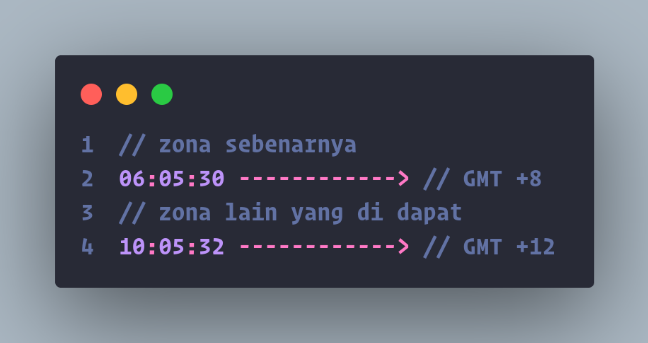
Dengan fungsi yang sudah tersedia di kotlin, dapat digunakan syntax sebagai berikut :

Maka hasil nilai *sudoRandom* = 22. Sehingga didapat *arrayTime* [*sudoRandom*]= GMT +12. Kemudian dilakukan konversi waktu sekarang 06:05:30 GMT +8 ke GMT +12 yang diperlihatkan pada Gambar 5 dan Gambar 6



Gambar 5. Potongan Kode Konversi Zona Waktu

Informasi yang digunakan adalah zona lain, perubahan zona sendiri merupakan proses, tujuanya mengkonsumsi sebuah waktu ketika mendapatkan informasi waktu itu sendiri.



Gambar 5. Hasil Zona Awal dan Zona Lain

4.4 Penentuan Konstanta P dan Q Berdasarkan Informasi Peranti

Gambar 4.1.5.1 *FlowChart* Program Terpilihnya konstanta atau orde P dan Q

Penentuan telah dilakukan dengan melihat syarat sebagai berikut:

1. Bilangan yang prima telah didapatkan dalam bentuk *arrayListPrimeNumber* hasilnya diperlihatkan pada Gambar 3.

2. Informasi Peranti telah didapatkan dalam bentuk bagian dari waktu jam, menit dan detik. Hasilnya diperlihatkan pada Gambar 5.

Kemudian menentukan *p* dan *q* dimana untuk menghasilkan prima yang deterministik dari informasi peranti yang probabilistik:

1. Menentukan Konstanta P yang Prima penentuan ini sederhana, dengan menghitung persamaan 1.1 didapat *indexP* = 20.

(………..….……………(1.1)

Dimana :

hh = informasi peranti waktu jam

n = 2

Maka didapat nilai *p*[*indexP*] = 73.

Jika n memiliki nilai yang lebih besar dari 2 , misal 3 maka memiliki tujuan terbentuknya *p* yang prima cukup besar.

Dengan *p* yang besar, memiliki kesempatan *Greatest* *Common* *Divisor* GCD(p, q) atau proses pemfaktoran yang memakan waktu lebih lama.

1. Menentukan Konstanta Q yang Prima nilai *q* memiliki aturan mirip dengan nilai *p*, tetapi memiliki 5 keputusan perhitungan () dari 6 ketentuanya ().

()…………………..…(2.1)

K1 = *p*

K2 = informasi peranti waktu menit

K3 = informasi peranti waktu detik

K4 = K2 + K3

K5 = K1 \* K2

K6 = K2 \* K3

()…………….……….(2.2)

Dimana :

K[n] = *arrayListPrimeNumber* [n]

jml prima = *arrayListPrimeNumber.size*

Dengan persamaan 2.1 dan 2.2 didapat K[n] = K[31] = *indexQ*.

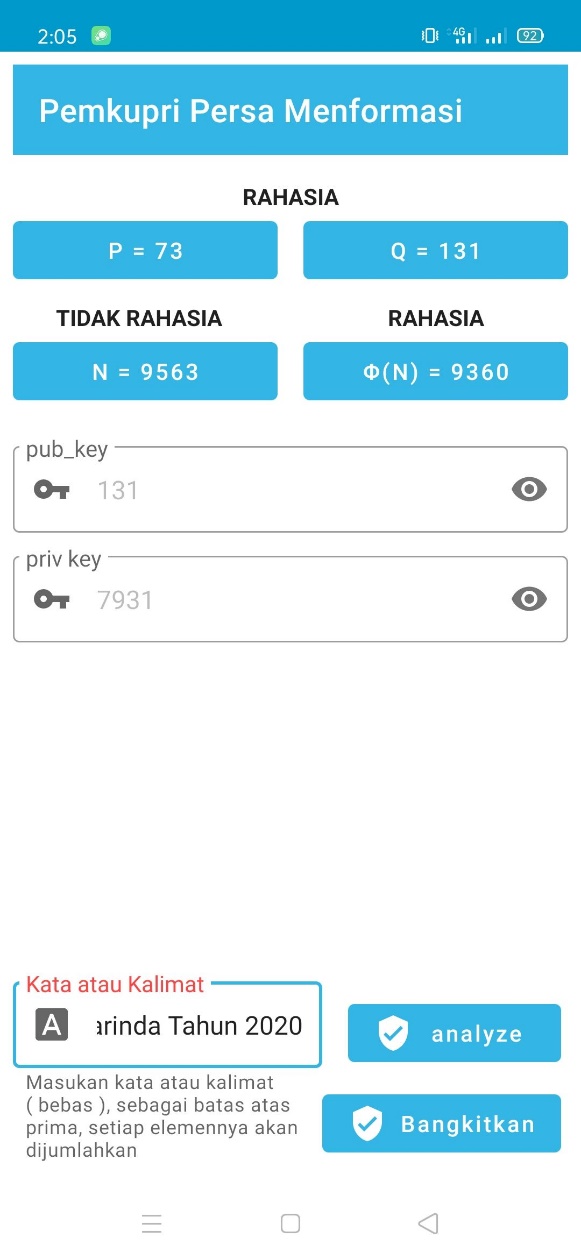
Maka didapat nilai *q*[*indexQ*] = 131.

4.2 Pengujian dan Pembuktian

asda

4.2.1 Analisa Hasil

Hasil *Greatest Common Divisor* (GCD) dari berjumlah sebanyak 2303, dimana i adalah 2 sampai 9360 (. Data disimpan secara urut (*ascending*) posisinya ke dalam daftar *array*. Satu data diambil berdasarkan nilai Q = 131 sebagai posisinya yang dimana isi *array* menunjukan nilai yang persis secara kebetulan yaitu e = 131.



Gambar 4.2.1 Hasil Pembangkitan Kunci Pada Peranti *Android*

Nilai N atau didefinisikan menjadi rentang 1 sampai d, dimana menghasilkan nilai 1, sehingga didapat . Label rahasia merujuk pada besaran-besaran algoritma rsa dan kunci publik (pub\_key) adalah e dan kunci privat (priv\_key) adalah d.

4.4 Analisa Hasil

Analisa hasil P dan Q dilakukan untuk melihat kunci privat sesuai (enkripsi dan dekripsi) dengan informasi peranti waktu yang berbeda. P dan Q dipilih berdasarkan nilai posisi secara acak (*pseudorandom*) serta waktu awal proses (HH:mm:ss) sampai perhitungan batas atas prima (hh:mm:ss) sehingga membuat P dan Q lebih tidak terduga dengan adanya 24 macam atau jenis *Greenwich Mean Time Zone* (GMT). Analisa memiliki 2 hasil yang saling berhubungan.

BAB V PENUTUP

5.1 Kesimpulan

Penlitian dan percobaan yang telah dilakukan menghasilkan kesimpulan sebagai berikut:

1. Proses mendapatkan waktu (HH:mm:ss dan hh:mm:ss) sekarang yang diterapkan bergantung peranti yang digunakan, ketika peranti memiliki ruang *memory* pengunaan yang besar, mampu melakukan perhitungan dan proses lebih cepat (berbeda). Sehingga data waktu dan perhitungan membuat hasil P dan Q lebih efisien dengan melihat hasil tidak terlalu besar dan rentang dua variabel itu sendiri.
2. Dari Analisa Hasil P dan Q
   1. Pengujian pertama dengan panjang kunci 2 *bit* sampai 14 *bit*, keduanya telah membangkitkan kunci privat yang mampu mendekripsi kode ASCII. Entropi (bagian yang terenkripsi) Blok *CipherText* menunjukan indikasi setengah ideal yaitu 4.814863028233948 dan probabilitas elemen *binary cipherText* berjumlah 58. P dan Q memiliki rentang jarak nilai rata-rata 269.3 dalam waktu 5 menit dan seluruh data memiliki rata-rata 120.4.
   2. Pengujian kedua dengan menaikan pemilihan P adalah hh \* 4 dan ditambahkanya ketentuan Q adalah batas prima dikurang posisi P, menghasilkan P dan Q yang memiliki kemungkinan rentang cukup jauh pada saat menit dan detik kecil antara 0 – 20 dan posisi P adalah puluhan atau lebih besar dari mm:ss. Kedua variabel menghasilkan modus GCD adalah 2.

5.2 Saran

Asd

RENCANA JADWAL PENGERJAAN



DATAR PUSTAKA

Firmansyah, F. F. 2015. Kajian matematis dan penggunaan bilangan prima pada algoritma kriptografi RSA (Rivest, Shamir, dan Adleman) dan algoritma kriptografi Elgamal [skripsi]. Malang (ID): Universitas Islam Negeri Maulana Malik Ibrahuim Malang.

Wicaksono, L. 2013. Kajian matematis dan penggunaan bilangan prima pada algoritma kriptografi RSA (Rivest, Shamir, dan Adleman) dan algoritma kriptografi Elgamal [skripsi]. Malang (ID): Universitas Islam Negeri Maulana Malik Ibrahuim Malang.

Cahyo Dhea Arokhman Yusufi. (2020). *Heuristic - For Mathematical Olympiad Approach*. Math Heuristic. https://books.google.co.id/books?id=OJriDwAAQBAJ&pg=PA18&lpg=PA18&dq=6k+%2B+1+selalu+prima+?&source=bl&ots=aWNDfVbx9w&sig=ACfU3U3JQyCKsvq5\_G4JUSbp8WKZhr\_7Tw&hl=en&sa=X&ved=2ahUKEwiW9eXuhr\_qAhUkheYKHfrHAJ4Q6AEwCnoECAoQAQ#v=snippet&q=prima&f=false

Chiewchanchairat, K., Bumroongsri, P., & Kheawhom, S. (2016). Improving fermat factorization algorithm by dividing modulus into three forms. *KKU Engineering Journal*, *40*(March), 131–138. https://doi.org/10.14456/kkuenj.2015.1

Ferreira, J. W. P. (2017). The Pattern of Prime Numbers. *Applied Mathematics*, *08*(02), 180–192. https://doi.org/10.4236/am.2017.82015

Harahap, M. K. (2019). *Membangkitkan Bilangan Prima Marsenne dengan metode Bilangan Prima Probabilistik Solovay – Strassen*. *1*(Oktober).

Kumari, J., Singh, S., & Saxena, A. (2015). *An Exception Monitoring Using Java*. *3*(2), 12–18.

Meštrović, R. (2018). *Euclid’s theorem on the infinitude of primes: a historical survey of its proofs (300 B.C.--2017) and another new proof*. http://arxiv.org/abs/1202.3670

Muchlis, B. S., Budiman, M. A., & Rachmawati, D. (2017). Teknik Pemecahan Kunci Algoritma Rivest Shamir Adleman (RSA) dengan Metode Kraitchik. *SinkrOn*, *2*(2), 49–64. http://jurnal.polgan.ac.id/index.php/sinkron/article/view/75

Nisha, S., & Farik, M. (2017). RSA Public Key Cryptography Algorithm A Review. *International Journal of Scientific & Technology Research*, *06*(07), 187–191.

Sari, R. H. (2017). Apakah Integrasi Islam dapat Membudayakan Literasi Matematika ? *Seminar Matematika dan Pendidikan Matematika UNY*, 655–662.

Sciences, T. (2016). *Dirichlet ’ s Theorem Related Prime Gap*. *10*, 305–310.

TH, A., & MB, B. (2017). The Unique Natural Number Set and Distributed Prime Numbers. *Journal of Applied & Computational Mathematics*, *06*(04). https://doi.org/10.4172/2168-9679.1000368

*Untuk Apa Mencari Bilangan Prima Terbesar? - Anak Bertanya*. (n.d.). Diambil 18 Juni 2020, dari https://anakbertanya.com/untuk-apa-mencari-bilangan-prima-terbesar/

Wulansari, D., Alamsyah, Setyawan, F. A., & Susanto, H. (2016). Mengukur Kecepatan Enkripsi dan Dekripsi Algoritma RSA pada Pengembangan Sistem Informasi Text Security. *Seminar Nasional Ilmu Komputer (SNIK 2016)*, *Snik*, 85–91.

Zulfikar, M. I., Abdillah, G., Komarudin, A., Informatika, J., & Sains, F. (2019). Kriptografi untuk Keamanan Pengiriman Email Menggunakan Blowfish dan Rivest Shamir Adleman (RSA). *Seminar Nasional Aplikasi Teknologi Informasi (SNATi) 2019*, *2*(1), 19–26.