

**PEMBANGKITAN KUNCI UNTUK PENENTUAN KONSTANTA
P DAN Q YANG PRIMA BERDASARKAN
INFORMASI PERANTI**

TUGAS AKHIR



Oleh :

**Yogi Arif Widodo
NIM. 17615006**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SAMARINDA
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA**

SAMARINDA 2020

HALAMAN JUDUL

Inisial sebagai index daftar isi

Print terpisah

HALAMAN PERNYATAAN ORISINALITAS

**Inisial sebagai index daftar isi
Print terpisah**

HALAMAN PENGESAHAN PEMBIMBING

Inisial sebagai index daftar isi

Print terpisah

HALAMAN PERSETUJUAN PENGUJI

**Inisial sebagai index daftar isi
Print terpisah**

KATA PENGANTAR

Puji syukur Alhamdulillah panjatkan kehadiran Allah SWT yang telah melimpahkan rahmatnya serta hidayahnya sehingga mampu menyelesaikan Proposal Tugas Akhir dengan judul **“Pembangkitan Kunci Untuk Penentuan Konstanta P dan Q yang Prima Berdasarkan Informasi 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 penyelesaian 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, Agustus 2020

Yogi Arif Widodo

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN PEMBIMBING	iii
HALAMAN PERSETUJUAN PENGUJI.....	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	x
ABSTRAK	xi
ABSTRACT.....	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian.....	3
BAB II LANDASAN TEORI	4
2.1 Kajian Ilmiah.....	4
2.2 Dasar Teori	5
2.2.1 Teori Bilangan.....	5
2.3 Kriptografi.....	12
2.4 Informasi Peranti	13
2.5 <i>Kotlin</i> dan Aliran Kontrol.....	14
2.6 <i>Exception Handling</i>	15
BAB III KERANGKA KONSEP DAN METODE PENELITIAN.....	17
3.1 Kerangka Konsep Penelitian	17

3.2	Metodologi Penelitian	20
3.2.1	Riset Awal.....	21
3.2.2	Tahapan Membangkitkan Bilangan Prima	21
3.2.3	Tahapan Mendapatkan Informasi Peranti.....	21
3.2.4	Tahapan Mengolah Informasi Peranti.....	21
3.2.3	Tahapan Penentuan Konstanta P dan Q Berdasarkan Informasi Peranti 23	
3.2.4	Pengujian dan Pembuktian.....	23
3.2.5	Analisa Hasil.....	23
3.2.6	Variabel Penelitian.....	24
3.2.7	Waktu dan Tempat Penelitian.....	24
BAB IV HASIL DAN PEMBAHASAN		25
4.1	Hasil Tahapan Membangkitkan Bilangan Prima.....	25
4.2	Hasil Tahapan Mendapatkan Informasi Peranti	26
4.3	Hasil Tahapan Mengolah Informasi Peranti.....	27
4.4	Hasil Tahapan Penentuan Konstanta P dan Q Berdasarkan Informasi Peranti.....	28
4.5	Hasil Pengujian dan Pembuktian.....	31
4.6	Analisa Hasil	34
BAB V PENUTUP.....		36
5.1	Kesimpulan.....	36
5.2	Saran.....	37
DATAR PUSTAKA		38

DAFTAR TABEL

Tabel 4.1 Hasil Pembangkitan Bilangan Prima	26
Tabel 4.3 Daftar Waktu Indonesia Tengah.....	27
Tabel 4.4 Hasil ($q_{keputusan}$) dan ($q_{ketentuan}$).....	30
Tabel 4.5 Hasil Pengujian dan Pembuktian Pembangkitan.....	32

DAFTAR GAMBAR

Gambar 2.6 <i>Flow Exception Handling</i>	16
Gambar 3.1. Diagram Alur Kerangka Konsep Penelitian.....	17
Gambar 3.2. Diagram Alur Metodologi Penelitian.....	20
Gambar 3.2.4 Tahapan Mengolah Informasi Peranti Waktu	22
Gambar 4.1 FlowChart Proses Naive Solution.....	25
Gambar 4.3 Hasil Informasi Peranti Waktu.....	28
Gambar 4.5.1 Hasil Pengujian Mendapati <i>NumberFormatException</i>	31
Gambar 4.5.2 Hasil <i>Logcat Android Studio</i> Pengecualian <i>NumberFormatException</i>	31
Gambar 4.5.3 Hasil Pengujian Mendapati <i>ArrayIndexOutOfBoundsException</i>	32
Gambar 4.5.4 Tampilan Aplikasi dan Proses Pembangkitan P dan Q.....	33
Gambar 4.6 Analisa Hasil GCD P dan Q.....	34

ABSTRAK

Jika difaktorkan hanya habis dibagi oleh angka 1 dan dengan dirinya sendiri disebut Bilangan Prima. Keunikannya selalu berbentuk antara $6k-1$ atau $6k+1$. Salah satu konsep atau metode berhubungan dengan bilangan yang prima dimiliki oleh Rivest Shamir Adleman (RSA), untuk pembangkitan kuncinya dibagi menjadi 2 buah pola yaitu variabel p dan q . Konstanta atau orde p dan q menjadi eksperimen aritmatika dalam kombinasi informasi peranti waktu pada *android mobile* dengan bentuk jam (HH), menit (mm) dan detik (ss). Greenwich Mean Time Zone (GMT) merupakan zona waktu informasi menjadikannya berpola deterministik menjadi probabilistik jika diolah menggunakan *pseudorandom* kemudian menghasilkan *index* waktu yang mengubah Zona Awal (ZA) 15:17:02 GMT + 8 ke Zona Lain (ZL) menjadi 10:17:03 GMT - 11. Waktu yang digunakan ketika terjadinya proses aritmatika yaitu ZL. HH berperan dalam pembentukan p sedangkan q dipengaruhi oleh mm dan ss dengan ketentuan sebagai *index* yang sedemikian rupa. Pembangkitan awal ditentukan dengan batas atas prima $n = 512$. Dengan teknik sederhana *naive solution* dimana 2 ke $n - 1$ menghasilkan *arrayListPrimeNumber* = 2,3,5,7,9..n. Kombinasi dan Aritmatika berhasil menentukan $p = 179$ dan $q = 419$. Hasil GCD ($p - 1, q - 1$) tidak terlalu besar menandakan pemfaktoran memakan waktu dan rentang p dan q memiliki jarak selisih yang jauh. Meningkatkan hasil Prima yang besar, dapat dilakukan dengan menaikkan nilai *inisial* dan n yang ditetapkan pada rumus $P_{penentuan}$ dan $q_{penentuan}$. Seluruh proses diuji keberhasilannya dengan pengecualian atau *Exception Handling*, hasilnya tidak ada *problem* pada *feedback monitoring* aplikasi android dan menggunakan *kotlin* sebagai *compiler*

Kata kunci : Bilangan Prima, Informasi Peranti Waktu, P dan Q

ABSTRACT

If it is factored, it can only be divided by the number 1 and by itself it is called a prime number. The uniqueness is always in the form between $6k-1$ or $6k+1$. One of the concepts or methods related to prime numbers is owned by Rivest Shamir Adleman (RSA). The key generation is divided into 2 patterns, namely variables p and q . Constants or orders p and q become arithmetic experiments in a combination of time device information on an android mobile in the form of hours (HH), minutes (mm) and seconds (ss). Greenwich Mean Time Zone (GMT) is an information time zone making it a deterministic pattern to be probabilistic if processed using pseudorandom then producing a time index that changes the Initial Zone (ZA) 15:17:02 GMT + 8 to Other Zones (ZL) to 10:17: 03 GMT - 11. The time used when the arithmetic process occurs is ZL. HH plays a role in the formation of p while q is influenced by mm and ss provided that it is an index in such a way. Initial generation is determined with an upper limit of prime $n = 512$. With a simple naive solution technique where 2 to $n - 1$ results in `arrayListPrimeNumber = 2,3,5,7,9..n`. Combination and Arithmetic succeeded in determining $p = 179$ and $q = 419$. The GCD results ($p - 1$, $q - 1$) were not too large, indicating factoring was time consuming and the range of p and q had a large difference. Increasing the large prime results, can be done by increasing the initial value and n set in the formula $P_{penentuan}$ dan $q_{penentuan}$. The entire process is tested for success with exception or Exception Handling, the result is there are no problems with monitoring feedback for the Android application and using Kotlin as the compiler

Keywords: Prime Number, Information Time Device, P and Q

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). Setiap bilangan asli lebih dari 1 yang tidak prima disebut bilangan komposit (Harahap, 2019). Jika n adalah suatu bilangan komposit, maka n memiliki setidaknya 1 faktor prima yang nilainya tidak lebih dari \sqrt{n} .

Bilangan prima yang lebih besar dari 3 memiliki keunikan yang selalu berbentuk antara (Chiewchanchairat dkk., 2016) $6k-1$ atau $6k+1$ (Ferreira, 2017), dimana k adalah bilangan prima yang diketahui. Maka dari itu bilangan prima yang lebih dari 3 akan selalu memiliki antara dua bentuk tadi. Hasil selanjutnya didapat mengenai bilangan prima adalah bahwa bilangan prima ada tak hingga banyaknya (Meštrović, 2018; Sciences, 2016). Bilangan Prima merupakan bilangan bulat positif, sifat pembagiannya (Firmansyah, 2015) melahirkan konsep-konsep aritmatika *modulo*, dan salah satu konsep bilangan bulat yang digunakan dalam penghitungan komputer.

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. Dengan ditemukannya bilangan prima, teori bilangan berkembang semakin jauh dan lebih mendalam. Banyak dalil dan sifat dikembangkan berdasarkan bilangan prima (Firmansyah, 2015), salah satunya adalah Kriptografi *Rivest Shamir Adleman* (RSA) yang memiliki 2 buah pola bilangan prima dan ditetapkan sebagai variabel p dan q untuk pembangkitan kunci RSA (Sylfania dkk., 2019), Selain itu setiap angka genap yang cukup besar dapat ditulis sebagai jumlah dari beberapa bilangan prima dan nomor lain yang merupakan produk dari dua bilangan prima (Kumari dkk., 2015).

Pada penelitian (TH & MB, 2017) Pengecualian atau *Exception Handling* merupakan cara bersih memeriksa kesalahan tanpa mengacaukan kode dan mampu menangkap pengecualian sebuah. Aritmatika salah satunya *NumberFormatException*. Klausula tangkapan diikuti blok coba (*try and catch*), setiap blok tangkapan merupakan pengecualian yang menangani jenis pengecualian.

Berdasarkan sifat Bilangan Prima, maka penelitian ini mengkombinasikan informasi peranti waktu jam, menit dan detik pada android mobile menjadi teknik penentuan konstanta p dan q juga memastikan ketentuan prosesnya terpenuhi dan menghasilkan pola tersendiri tanpa ada pengecualian sebagai tanda berhasilnya proses pada Pengujian dan Pembuktian terhadap Tahapan Penentuan Konstanta P dan Q Berdasarkan Informasi Peranti

1.2 Rumusan Masalah

Dalam melaksanakan penelitian, masalah yang menjadi poin utama diskusi atau pembahasan, adalah “Bagaimana Melakukan Pembangkitan Kunci 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 Untuk Penentuan Konstanta P dan Q Yang Prima
3. Memonitoring konsep informasi peranti waktu dengan *Exception Handling*, sebagai indikator berjalannya konsep yang dibuat.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian sebagai berikut:

1. Informasi peranti menggunakan waktu jam, menit dan detik dan Zona waktu adalah **GMT -11:00** sampai **GMT +13:00**.
2. 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 *mobile android*.
2. Dapat menjadi sumber referensi bagi pihak lain dalam menyusun karya ilmiah maupun penelitian yang berkaitan dengan judul pada penelitian.

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 $\text{GCD}(p-1, q-1)$, 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 \mathbb{Z} yang diambil dari kata *Zahlen* dari bahasa Jerman atau dinotasikan dengan \mathbb{I} yang diambil dari huruf pertama kata *Integer* dari bahasa Inggris, adalah himpunan $\dots, -3, -2, -1, 0, 1, 2, 3, \dots$. Himpunan bilangan bulat dibagi tiga, yaitu bilangan bulat positif, yaitu bilangan bulat yang lebih besar dari nol yang dituliskan \mathbb{Z}^+ , nol, dan bilangan bulat negatif, yaitu bilangan bulat yang lebih kecil dari nol yang dituliskan \mathbb{Z}^- . Himpunan bilangan bulat dilengkapi dengan dua buah operasi, yaitu operasi penjumlahan dan perkalian, dilambangkan $(\mathbb{Z}, +, \cdot)$ 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 $a, b \in \mathbb{Z}$ dan $a > 0$, maka ada bilangan $q, r \in \mathbb{Z}$ yang masing-masing tunggal sehingga $b = qa + r$ dengan $0 \leq r < a$. 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 $\phi(p) = p - 1$. Karena p adalah bilangan prima, maka setiap bilangan bulat positif kurang dari p relatif prima terhadap p . Ini berarti bahwa sistem residu tereduksi modulo p adalah himpunan $\{1, 2, 3, \dots, p - 1\}$ yang mana seluruh anggotanya sebanyak $(p - 1)$ sehingga $\phi(p) = p - 1$.

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 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 $q \neq 1$ 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 \sqrt{p} . Jika p habis dibagi salah satu dari bilangan prima tersebut, maka p adalah bilangan komposit tetapi jika p tidak habis dibagi 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

$$(p - 1)! \equiv -1 \pmod{p}$$

$$\text{ATAU } (p - 1)! \equiv (p - 1) \pmod{p}$$

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

$$a^{n-1} \equiv 1 \pmod{n}$$

$$\text{ATAU}$$

$$a^{n-1} \% n = 1$$

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 $\text{FPB}/\text{GCD}(x, y) = 1$.

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

Contohnya Bilangan 20 dan 3 adalah relatif prima karena $\text{FPB}(20, 3) = 1$, atau dapat ditulis: $2 \cdot 20 + (-13) \cdot 3 = 1$, dengan $m = 2$ dan $n = -13$. Tetapi 20 dan 5 tidak relatif prima karena $\text{FPB}(20, 5) = 5 \neq 1$ 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 penelitian (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 lainnya:

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), sehingga n mempunyai 200 digit lebih (Wulansari dkk., 2016).
2. Mencari nilai n dengan memanfaatkan persamaan 2.1.

$$n = p * q \dots\dots\dots (2.1)$$

3. Mencari nilai ekuivalen dengan persamaan 2.2.

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

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

Berdasarkan penelitian ini memberikan gagasan atau ide dalam eksperimen yang ditujukan pada pembangkitan kunci p dan q berdasarkan informasi 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 menunjukkan 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 penelitian 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.).

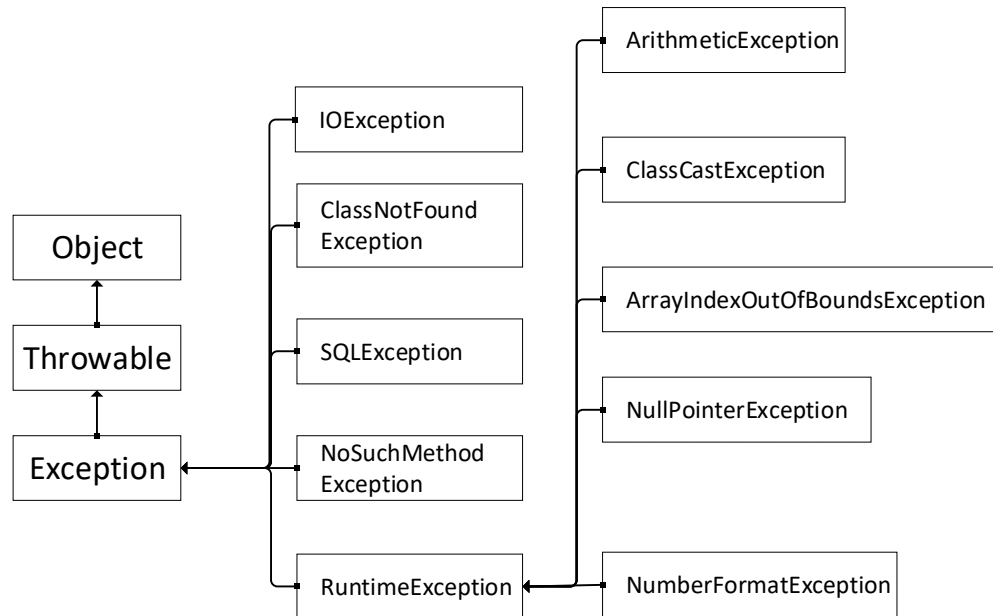
Pada penelitian ini operasi dan pengujian menggunakan bahasa *kotlin* yang dikompilasi atau dijalankan oleh *mobile android*. Beberapa aliran kontrolnya terdapat *if*, *when*, *for* dan *while*. Semua aliran digunakan menangani aritmatika dan konsep yang dibuat sesuai kebutuhan.

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. Hal ini menciptakan kebingungan antara kebenaran (memeriksa semua kemungkinan kesalahan) dan kejelasan (tidak mengacaukan aliran kode dasar dengan banyak pemeriksaan kesalahan). Jadi, untuk mengatasinya, diperlukan jalan tengah dan persyaratan ini dipenuhi oleh PENGECUALIAN.

Java menyediakan cara yang efisien untuk menangani kondisi tak terduga yang terjadi selama eksekusi program. Ketika ada kesalahan atau *bug* dalam program maka program akan segera berhenti setelah kesalahan ditemukan. Ini membuat program dalam keadaan tidak konsisten. Untuk menghindari situasi seperti itu, *Java* menggunakan *Exception Handling* yang merupakan keuntungan besar sehingga setiap kali ada kondisi luar biasa, program akan menanganinya dengan cara dan melanjutkan eksekusi program tanpa meninggalkannya dengan cara yang tidak konsisten. *java.lang*. Kelas *Throw* mampu menangani seluruh konsep pengecualian dan kesalahan. Dalam makalah ini kami fokus pada keuntungan dari penanganan *Exception* dan implementasinya dalam program untuk membuatnya kuat. *Exception* menyediakan cara yang bersih untuk memeriksa *error* tanpa kode yang berantakan. Pengecualian yang ditentukan pengguna untuk membuat program lebih dapat diandalkan. Penanganan pengecualian adalah fitur

penting untuk toleransi kesalahan perangkat lunak yang memungkinkan pengembang menghasilkan sistem perangkat lunak yang andal dan kuat



Gambar 2.6 *Flow Exception Handling*

Tujuan utama dari mekanisme penanganan *Exception* adalah menyediakan cara untuk 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:

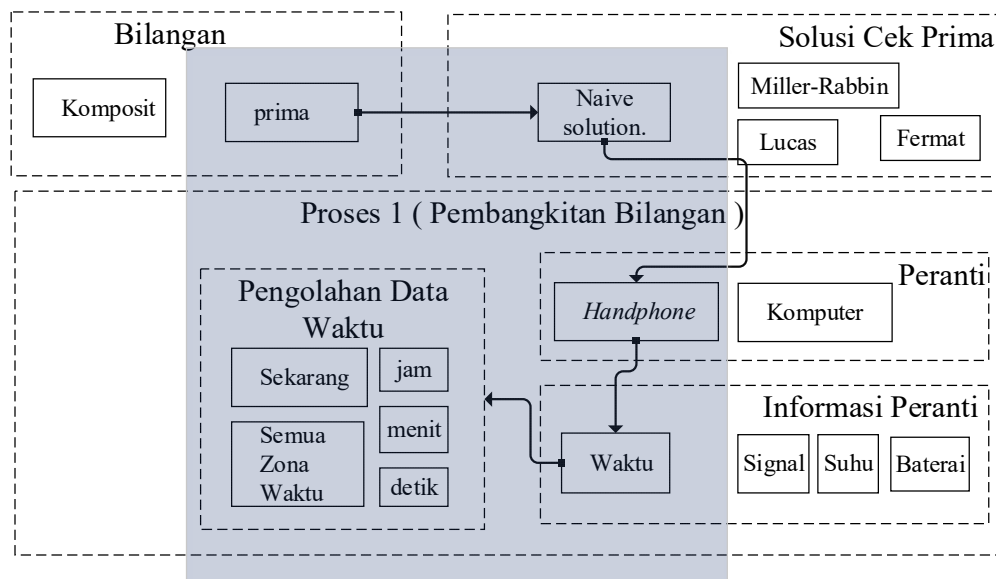
1. Menemukan masalah yaitu *Hit the Exception*
2. Menginformasikan bahwa telah terjadi kesalahan yaitu Lempar Pengecualian
3. Menerima informasi kesalahan yaitu Menangkap pengecualian
4. Ambil tindakan korektif, yaitu Menangani pengecualian Melempar

BAB III

KERANGKA KONSEP DAN 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 Alur Kerangka Konsep Penelitian

Berdasarkan Gambar 3.1 diagram alur kerangka konsep penelitian dapat dijelaskan secara singkat.

1. Prima

Pada penelitian ini objek atau bahan yang diolah adalah Bilangan Prima. Dalam bilangan terdapat berbagai jenis bilangan, dua diantaranya yaitu:

- a. Bilangan Komposit
- b. Bilangan Prima

Bilangan prima berhubungan dekat dengan penelitian ini sebagai konsep pemilihan konstanta p dan q , dimana bilangan yang prima tidak hanya unik melainkan memiliki bentuk $6k-1$ atau $6k+1$.

2. *Naive Solution*

Solusi sederhana untuk mengecek bilangan kecil yang prima tepatnya dapat ditangani dengan *Naive Solution*. Macam-macam solusi lain yang mengenai bilangan prima lebih luas, diantaranya adalah:

- a. *Fermet*
- b. *Miller-Rabbin*
- c. *Solovay-Strassen*
- d. *Lucas*

3. *Handphone*

Pengujian ini didasarkan pada peranti yang pemrosesan aritmatika dilakukan dengan *handphone* yang dapat menghasilkan sebuah *metadata* atau informasi yang beragam sesuai kebutuhan yaitu waktu jam menit dan detik.

4. Informasi Peranti Waktu

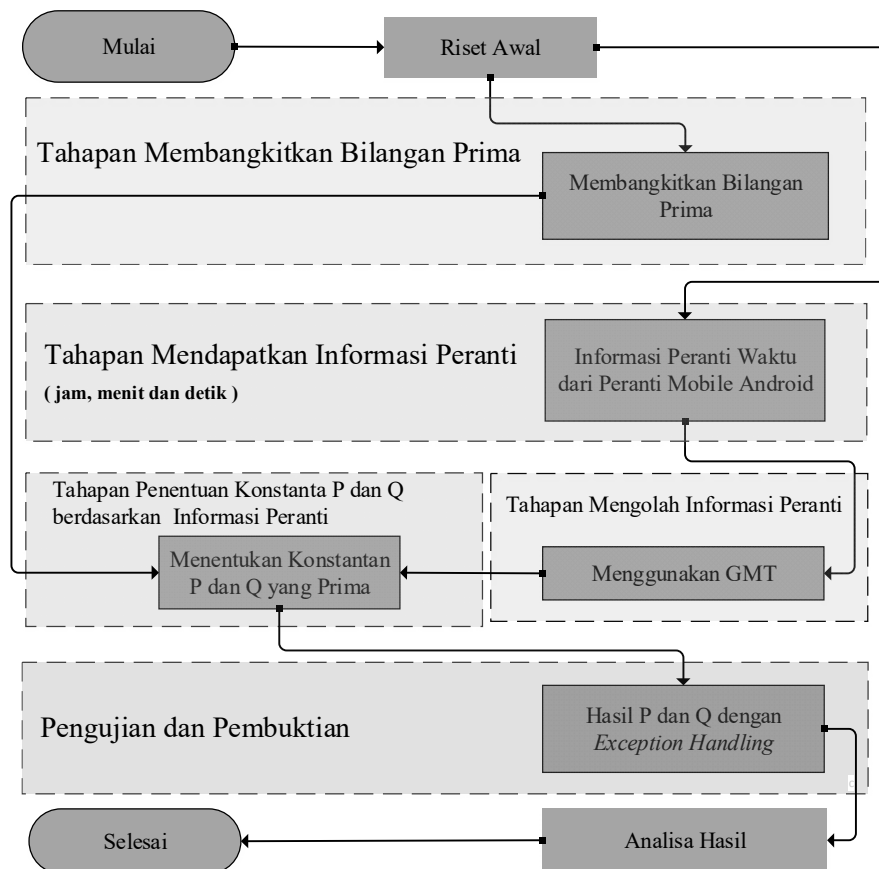
Waktu merupakan perhitungan masa dunia, dengan waktu dapat mengetahui kapan suatu hal terjadi. Khususnya proses aritmatika yang terjadi pada suatu perangkat atau peranti. Informasi ini dapat dengan mudah diperoleh dengan alat elektronik contohnya *handphone* atau komputer

5. Pengolahan Data Waktu

Waktu memiliki berbagai macam jenis yang dapat diolah sebagai konsep pemanfaatan nilainya, seperti jam menit dan detik ataupun zona waktu. Dengan begitu data waktu dapat dirancang sedemikian rupa untuk menghasilkan bilangan yang prima berdasarkan kejadian atau waktu yang diperoleh serta menjadikanya sebuah posisi dalam memilih bilangan yang prima.

3.2 Metodologi Penelitian

Metode Penelitian menjelaskan mengenai tahapan-tahapan pengerjaan dari penelitian yang dilakukan. Metode penelitian ini bertujuan agar penyelesaian penelitian ini tidak terlepas dari penggunaan metode yang dikerjakan.



Gambar 3.2 Diagram Alur Metodologi Penelitian

Pada Gambar 3.2, tahapan dari metodologi penelitian yang dilakukan dimulai pada riset awal, kemudian membangkitkan bilangan prima dan pada bagian alur kedua mendapatkan informasi peranti waktu, mengolah informasi waktu menggunakan GMT, setelah itu penentuan konstanta p dan q berdasarkan informasi peranti, pengujian dan pembuktian, terakhir adalah analisa hasil.

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 waktu
3. Landasan matematika (teori bilangan dan pempfaktoran bilangan bulat)

3.2.2 Tahapan Membangkitkan Bilangan Prima

Membangkitkan Bilangan Prima dengan mengeliminasi angka bukan prima (TH & MB, 2017). Penerapannya sederhana akan dilakukan dengan *naive solution* dengan hasil yang tersimpan dalam *ArrayList* sehingga dapat diolah lebih lanjut berdasarkan informasi peranti.

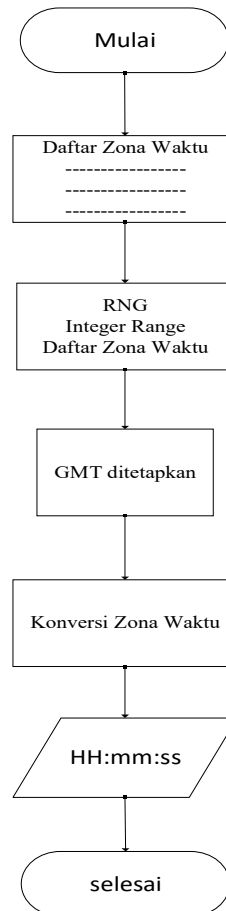
3.2.3 Tahapan Mendapatkan Informasi Peranti

Informasi Peranti yang didapatkan berupa 3 variabel yaitu jam, menit, dan detik. Proses mendapatkannya dibaca oleh peranti *Mobile Android*. Data waktu yang didapat bukan berupa nilai seperti 1594886148236 melainkan jam menit dan detik serta zona waktu sebagai contoh informasi yang dimaksud seperti 15:07:00 GMT+8.

3.2.4 Tahapan Mengolah Informasi Peranti

Informasi Peranti diolah kembali untuk menghasilkan informasi peranti yang probabilistik berdasarkan angka pseudorandom dari jumlah 24 zona waktu dalam artian 1 sampai 24, kemudian angkanya akan menentukan zona yang

terdaftar atau konversi terhadap data zona waktu yang telah didapatkan seperti yang diperlihatkan pada Gambar 3.2.4.



Gambar 3.2.4 Tahapan Mengolah Informasi Peranti Waktu

Perubahan zona sendiri merupakan proses, tujuannya mengkonsumsi sebuah waktu ketika mendapatkan informasi waktu itu sendiri sehingga menyerupai data yang probabilistik, pada intinya informasi yang akan dipakai bukan waktu dengan zona awal melainkan zona lain yang ditetapkan dari daftar zona secara pseudorandom.

3.2.3 Tahapan Penentuan Konstanta P dan Q Berdasarkan Informasi

Peranti

Penentuan konstanta p dan q akan dilakukan berdasarkan informasi peranti dengan melihat syarat sebagai berikut:

1. Bilangan yang prima telah didapatkan dalam bentuk `arrayListPrimeNumber` hasilnya berdasarkan pada Tahapan Menentukan Bilangan Prima.
2. Informasi Peranti telah didapatkan dalam bentuk bagian dari waktu jam, menit dan detik.

Kemudian tahapan penentuan p dan q dapat diproses lebih lanjut dengan menggabungkan syaratnya, syarat dua akan menjadi posisi yang menjadikan syarat pertama menjadi outputnya sedemikian rupa, dimana p dipengaruhi oleh nilai jam sedangkan q dipengaruhi oleh menit dan detik.

3.2.4 Pengujian dan Pembuktian

Pengujian dilakukan dengan monitoring konsep menggunakan *Exception Handling* atau pengecualian sehingga akan membuktikan kombinasi peranti berhasil dilakukan dalam memilih konstanta p dan q yang prima jika tidak ada pengecualian yang dihasilkan.

3.2.5 Analisa Hasil

Menganalisa hasil dan proses terpilihnya kunci p dan q yang akan dikaitkan dengan $\text{GCD}(p-1, q-1)$ dan rentang diantara keduanya dengan pembangkitan selama 1 jam dan membandingkan seluruh hasilnya berdasarkan proses kombinasi waktu.

3.2.6 Variabel Penelitian

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

3.2.7 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Jurusan Teknologi Informasi Politeknik Negeri Samarinda dengan waktu pengerjaan berdasarkan jadwal pengerjaan tugas akhir.

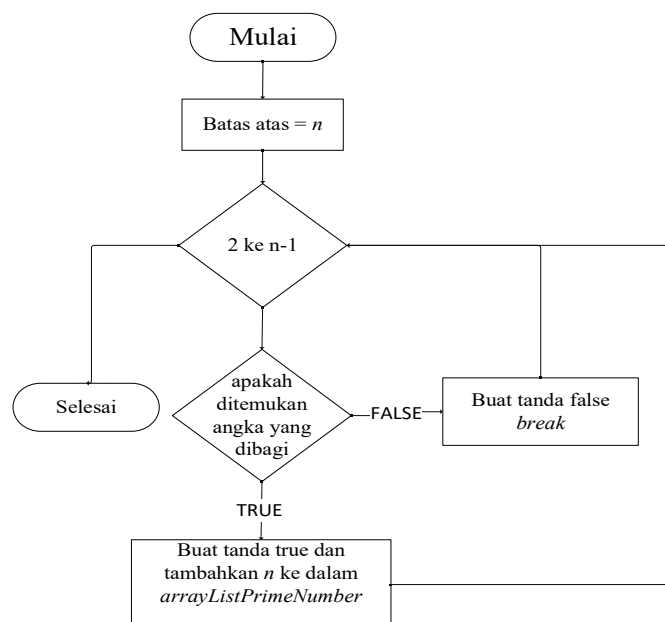
BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Tahapan Membangkitkan Bilangan Prima

Membangkitkan Bilangan Prima dengan mengeliminasi angka bukan prima (TH & MB, 2017). Penerapannya 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 4.1 FlowChart Proses Naive Solution

Tabel 4.1 Hasil Pembangkitan Bilangan Prima

<i>arrayListPrimeNumber</i>	prima	2	3	5	..	509
	size	1	2	3	..	97

Jadi pada penelitian ini *Naive Solution* membangkitkan bilangan yang prima sebanyak 97 dan bilanganya dimulai dari 2,3,5,7 sampai 509 seperti yang diperlihatkan pada Tabel 4.1

Nilai n telah ditentukan sebelumnya, n merupakan batas atas prima yang di atur dan bernilai 512. Jika n memiliki nilai yang lebih besar dari 512, maka memiliki tujuan menaikkan nilai *inisial* pada rumus penentuan p dan q sehingga membangkitkan hasil prima yang cukup besar.

4.2 Hasil Tahapan Mendapatkan Informasi Peranti

Pada tahapan ini dilakukan ketika proses sebelumnya telah usai dikerjakan sehingga informasi yang didapat menyerupai aturan probabilistik. Informasi Peranti yang didapatkan memiliki 3 variabel yaitu jam, menit, dan detik dan Tambahan Zona Waktu. Proses mendapatkannya dibaca oleh peranti *Mobile Android* dengan fungsi yang sudah tersedia di *kotlin* menggunakan *Package Kotlin System*.

Data waktu yang didapat masih berupa nilai keseluruhan waktu 1594886148236, kemudian diformat menjadi (HH:mm:ss) untuk menjadikanya jam, menit dan detik. Dengan fungsi yang sudah tersedia di *kotlin* menggunakan *Open Class SimpleDateFormat*.

Maka hasil yang informasi peranti waktu yang didapatkan 15:17:02 dengan zona awal GMT +8.

4.3 Hasil Tahapan Mengolah Informasi Peranti

Informasi Peranti diolah kembali untuk menghasilkan informasi peranti yang probabilistik berdasarkan waktu jam, menit dan detik serta menggunakan *Greenwich Mean Time Zone* (GMT) sebagai pengubah Zona Awal ke Zona Lain. Seluruh zona waktu telah didefinisikan sebelumnya ke dalam *arrayTime* sebagai zona lain yang diperlihatkan pada Tabel 4.3

Tabel 4.3 Daftar Waktu Indonesia Tengah

Waktu Tengah Dunia			
GMT (-)		GMT (+)	
GMT-1	GMT-6	GMT+1	GMT+6
GMT-2	GMT-7	GMT+2	GMT+7
GMT-3	GMT-8	GMT+3	GMT+8
GMT-4	GMT-9	GMT+4	GMT+9
GMT-5	GMT-10	GMT+5	GMT+10
	GMT-11		GMT+11
			GMT+12
			GMT+13

Pemilihan posisi atau *index* untuk *arrayTime* berdasarkan keluaran dari nilai *integer* oleh *sudoRandom*, sebagai zona lain. Dengan fungsi yang sudah tersedia di *kotlin* menggunakan *Package Kotlin Random*.

Pada penelitian ini hasil nilai *sudoRandom* = 22, maka didapat *arrayTime* [*sudoRandom*] = GMT +12. Kemudian dilakukan konversi waktu sekarang 15:17:02 GMT +8 ke GMT -11 Dengan fungsi yang sudah tersedia di *kotlin* menggunakan *Open Class SimpleDateFormat* dan hasilnya akhir diperlihatkan pada Gambar 4.3



```
1 // zona sebenarnya
2 06:05:30 -----> // GMT +8
3 // zona lain yang di dapat
4 10:05:32 -----> // GMT +12
```

Gambar 4.3 Hasil Informasi Peranti Waktu

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

4.4 Hasil Tahapan Penentuan Konstanta P dan Q Berdasarkan Informasi Peranti

Berdasarkan penentuan yang 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 4.2.

Kemudian tahapan penentuan p dan q dapat diproses lebih lanjut dengan menggabungkan syaratnya, syarat dua telah menjadi posisi yang menjadikan syarat pertama menjadi outputnya sedemikian rupa, dimana p dipengaruhi oleh nilai jam sedangkan q dipengaruhi oleh menit dan detik sebagaimana tahapan berikut:

A. Menentukan Konstanta P yang Prima, penentuan ini sederhana, dengan menghitung persamaan 1.1 didapat $i = 40$.

$$(P_{penentuan}) \dots \dots \dots (1.1)$$

$$Pi = hh * inisial$$

Dimana :

Pi = List Array Ke-i

I = Index arrayListPrimeNumber

$inisial$ = 4

hh = Informasi Peranti Waktu Jam

Maka didapatkan nilai $Pi = 179$. Jika n memiliki nilai yang lebih besar dari 4 misal 5 maka memiliki tujuan terbentuknya p yang prima cukup besar. Dengan p yang besar, memiliki kesempatan *Greatest Common Divisor* $GCD(p-1, q-1)$ atau proses pemfaktoran yang memakan waktu lebih lama.

B. Menentukan Konstanta Q yang Prima, nilai q memiliki aturan mirip dengan nilai p , tetapi memiliki 2 keputusan perhitungan ($q_{keputusan}$) dari 2 ketentuannya ($q_{ketentuan}$).

$$(q_{ketentuan}) \dots \dots \dots (2.1)$$

$K1$ = informasi peranti waktu menit

$K2$ = informasi peranti waktu detik

$$(q_{keputusan}) \dots \dots \dots (2.2)$$

$$Q_i = \begin{cases} \text{inisial} * (K1 * K2) \bmod q.size, & K1 < K2 \\ \text{inisial} * (K1 * K2) \bmod q.size, & K1 > K2 \end{cases}$$

Dimana:

Q_i = List Array Ke-i

i = Index arrayListPrimeNumber

$inisial$ = 4

$q.size$ = arrayListPrimeNumber.size

Dengan persamaan 2.1 dan 2.2 didapat $K1 > K2$ seperti yang diperlihatkan pada Tabel 4.3

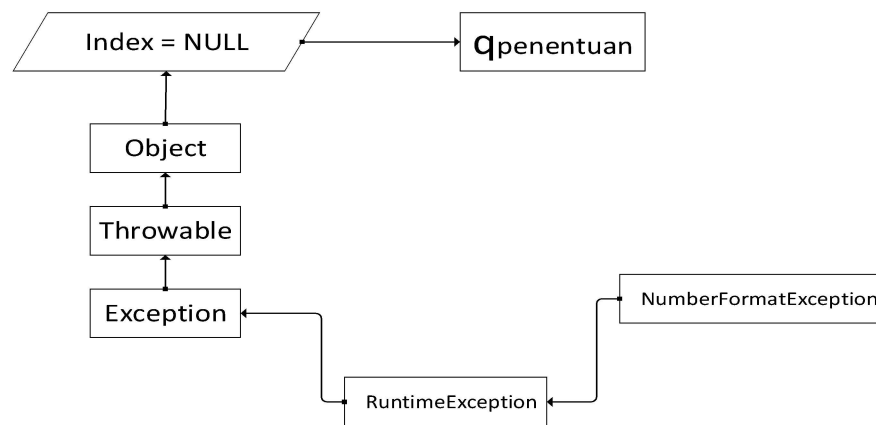
Tabel 4.4 Hasil ($q_{keputusan}$) dan ($q_{ketentuan}$)

informasi peranti waktu (HH:mm:ss)		10:17:03	
arrayListPrimeNumber .size		97	
inisial		4	
index		Kondisi	
Q_i	10	$K1 < K2$	FALSE
Q_i	80	$K2 > K1$	TRUE
Hasil			
$q_{keputusan}$		q_{Q_i}	419

Tahapan ini berhasil menentukan dan menghasilkan $P_i=179$ dan $Q_i = 419$, sesuai ketentuan yang ditetapkan pada proses A dan B, selanjutnya nilai dan konsep yang dihasilkan memasuki Tahapan Pengujian dan Pembuktian.

4.5 Hasil Pengujian dan Pembuktian

Pengujian pertama awalnya berjalan normal, tetapi di suatu pembangkitan dan percobaan berikutnya mendapati sebuah pengecualian dimana konsep awal mengalami atau menghasilkan nilai yang kosong atau *null*, sehingga mendapati *NumberFormatException* seperti yang diperlihatkan pada Gambar 4.5.1 dan Gambar 4.5.2



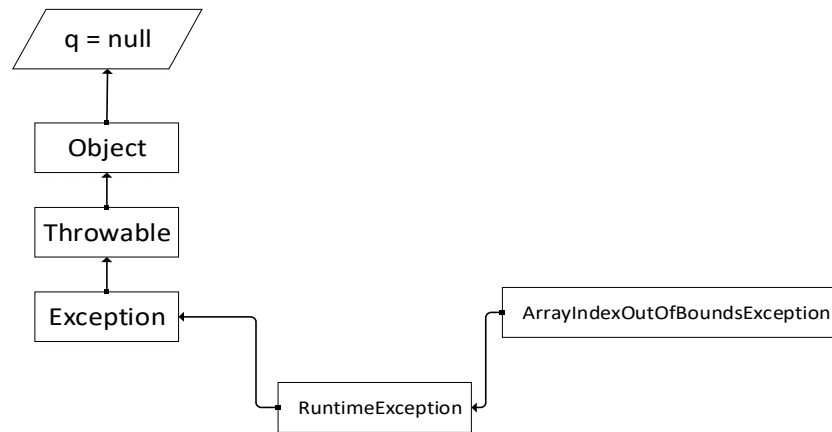
Gambar 4.5.1 Hasil Pengujian Mendapati *NumberFormatException*

```
2020-04-22 09:00:05.528 D/ChoosePq: number P : 97 | where thePrime[timePosition]
2020-04-22 09:00:05.528 D/ChoosePq: number Q : null | where thePrime[timePosition]
2020-04-22 09:00:05.528 D/MainActivity: catch err java.lang.NumberFormatException: For input string: "null"
2020-04-22 09:00:06.536 D/MainActivity: [*]
```

Gambar 4.5.2 Hasil *Logcat Android Studio* Pengecualian

NumberFormatException

Proses masih berjalan menentukan *q*, sehingga pengecualian berikutnya juga mendapati artimatika yang hasilnya melebihi batas prima yang ditentukan, sehingga tahapan penentuan *q* terjadi *ArrayIndexOutOfBoundsException* seperti yang diperlihatkan pada Gambar 4.5.3.



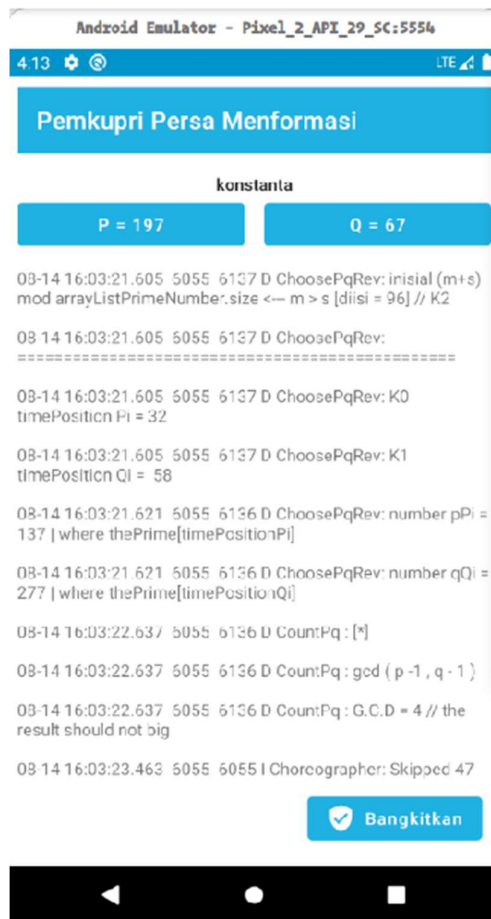
Gambar 4.5.3 Hasil Pengujian Mendapati *ArrayIndexOutOfBoundsException*

Pengujian kedua lolos dalam segala hal pengecualian, penambahan modulus menjadi solusi penentuan q dengan persamaan 2.2 dan hasil pembangkitan diperlihatkan pada Tabel 4.5.

Tabel 4.5 Hasil Pengujian dan Pembuktian Pembangkitan

DATA	Zona Awal (HH:mm:ss)	Zona Lain (hh:mm:ss)	<i>sudoRandom</i> Z_i	p	q	Selisih p dan q	GCD ($p - 1, q - 1$)
1	15:17:02 GMT + 8	10:17:03 GMT +10	10	179	419	240	2
2	16:03:20 GMT +8	08:03:21 GMT-8	7	137	277	140	4
3	16:08:18 GMT +8	18:08:19 GMT+6	16	367	103	264	6
4	16:13:18 GMT +8	11:13:19 GMT+13	23	197	67	130	2
5	16:18:18 GMT +8	08:18:19 GMT-8	7	137	31	106	2
6	16:23:18 GMT +8	08:23:19 GMT-8	7	32	71	39	2
7	16:28:18 GMT +8	17:28:19 GMT+7	17	347	479	132	2
8	16:33:18 GMT +8	08:33:19 GMT-8	7	137	47	90	2

9	16:38:18 GMT +8	21:38:19 GMT+3	13	439	149	290	2
10	16:43:18 GMT +8	18:43:19 GMT+6	16	367	257	110	2
11	16:48:18 GMT +8	10:48:19 GMT-10	9	179	379	200	2
12	16:53:18 GMT +8	22:53:19 GMT+2	12	461	499	38	2
13	16:58:18 GMT +8	02:58:19 GMT-2	1	23	61	38	2

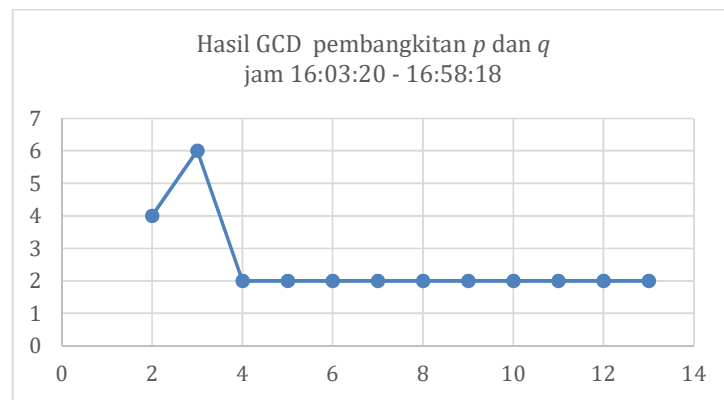


Gambar 4.5.4 Tampilan Aplikasi dan Proses Pembangkitan P dan Q

Pengujian penelitian ini menjadikanya pembuktian konsep yang benar berjalan berdasarkan sebuah aritmatika dengan memonitoring kejadian atau hitung menggunakan *Exception Handling* atau pengecualian dimana data 1 adalah hasil percobaan awal dan data ke-2 sampai data ke-13 adalah pengujian pembangkitan dalam tempo 300000 milidetik selama 1 jam.

4.6 Analisa Hasil

Berdasarkan Tabel 4.5 Greatest Common Divisor GCD ($p-1$, $q-1$) dan Gambar 4.6, rata-rata bernilai 2.



Gambar 4.6 Analisa Hasil GCD P dan Q

Berdasarkan pendapat (Muchlis dkk., 2017) bilangan yang prima dan memiliki hasil GCD kecil, memiliki proses pemfaktoran yang memakan waktu lebih lama, sehingga pada penelitian ini dengan bilangan prima yang kecil sudah dapat dikategorikan sebagai faktor yang cukup baik dalam menghasilkan kunci p dan q dalam penentuannya yang berdasarkan informasi peranti.

Hasil p dan q yang dibangkitkan berdasarkan informasi peranti waktu jam, menit dan detik yang merupakan bilangan prima rata-rata menghasilkan panjang p

dan q sebanyak 7 bit sampai 14 bit selama uji pembangkitan sebanyak 12 kali dalam tempo waktu setiap 300000 milidetik dalam 1 jam.

Hasil kombinasi informasi peranti waktu jam, menit dan detik, memberikan pola sedemikian rupa terhadap hasil p dan q , dengan bantuan informasi berupa nilai yang digunakan sebagai posisi atau *index* dan telah dibuktikan penentuan dalam ketentuan p dan q dengan monitoring *Exception Handling* menghasilkan konsep yang berjalan stabil atau tidak terjadi kesalahan pada konsep.

Hasil pada penelitian [14] membuktikan bahwa bahasa pemrograman kotlin dapat mengurangi waktu kompilasi, waktu eksekusi dan dapat meningkatkan keringkasan. Maka kemampuan konsep sederhana ini telah menghasilkan proses yang lebih optimal dan ditinjau dari konsepnya sendiri.

BAB V

PENUTUP

5.1 Kesimpulan

Penelitian 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* penggunaan yang besar, mampu melakukan perhitungan dan proses lebih cepat (berbeda). Sehingga data waktu dan perhitungan mempengaruhi hasil P dan Q lebih efisien dengan tetap melihat hasil GCD ($p - 1$, $q - 1$) tidak terlalu besar dan rentang dua variabel itu sendiri.
2. Dari Analisa Hasil P dan Q
 - A. Hasil p dan q yang dibangkitkan berukuran 2 *bit* sampai 14 *bit*, pola pembangkitan yang diketahui hanya waktu dalam 1 jam yaitu pada jam 16:03:20 GMT+8 sampai dengan 16:58:18 GMT+8.
 - B. Hasil proses kombinasi berdasarkan informasi peranti waktu diuji keberhasilannya dengan *monitoring* pengecualian atau *Exception Handling* yang menghasilkan tidak ada *Logcat Android Studio* berupa pengecualian di seluruh pemrosesan.

5.2 Saran

Asd

DATAR PUSTAKA

- 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+%2B1+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>
- FAQ - Kotlin Programming Language*. (n.d.). Diambil 15 Agustus 2020, dari <https://kotlinlang.org/docs/reference/faq.html>
- Fernanda, P. 2020. Keterbagian Oleh Dari Bilangan *Perfect* dan 4 - *Perfect* [skripsi]. Lampung (ID): UNIVERSITAS LAMPUNG BANDAR LAMPUNG 2020.
- Ferreira, J. W. P. (2017). The Pattern of Prime Numbers. *Applied Mathematics*, 08(02), 180–192. <https://doi.org/10.4236/am.2017.82015>
- 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.
- 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.
- Prime Numbers - GeeksforGeeks.* (n.d.). Diambil 15 Agustus 2020, dari
<https://www.geeksforgeeks.org/prime-numbers/?ref=lbp>
- 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.
- Serdano, A., Zarlis, M., Sawaluddin, & Hartama, D. (2019). Pengamanan Pesan Menggunakna Algoritma Hill Cipher Dalam Keamanan Komputer. *Jurnal*

Mahajana Informasi, 2, 1–5.

Sylfania, D. Y., Juniawan, F. P., Laurentinus, L., & Pradana, H. A. (2019). SMS Security Improvement using RSA in Complaints Application on Regional Head Election's Fraud. *Jurnal Teknologi dan Sistem Komputer*, 7(3), 116–120. <https://doi.org/10.14710/jtsiskom.7.3.2019.116-120>

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>

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.