

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

¹⁾Yogi Arif Widodo, ²⁾Mulyanto, S.Kom., M.Cs., dan ³⁾Bedi Suprpty, S.Kom., M.Kom.

^{1,2,3)}Program Studi, Teknik Informatika, Politeknik Negeri Samarinda

^{1,2,3)}Jl. Cipto Mangun Kusumo Sungai Keledang – Samarinda - Indonesia

E-mail : yogirenbox33@gmail.com, Penulis dua, dst...

ABSTRAK

Rivest Shamir Adleman (RSA) merupakan teknik kriptografi modern yang melewati batas paten selama 20 tahun, sehingga mudah dibaca secara bebas. Sulitnya memfaktorkan bilangan besar $n = p \cdot q$ menjadi faktor prima, serta perbedaan kunci dalam mengungkap teks maupun penyandian, membuat RSA menjadi salah satu teknik yang sulit dipecahkan. Bilangan konstanta atau orde p dan q menjadi eksperimen perhitungan menggunakan informasi peranti yaitu 24 zona waktu dengan format HH:mm:ss dan hh:mm:ss menghasilkan rentang 3000 lebih di waktu tertentu dan GCD kedua variable adalah 2. Pembangkitan tempo kelipatan 5 dalam menit selama kurang waktu 1 jam, menghasilkan entropi $p = 3.085055102756477$, $q = 3.7004397181410926$, konversi *Greenwich Mean Time Zone* (GMT) = 3.085055102756477, dan ideal acuan data uji adalah 3.7004397181410926. Kesesuaian waktu HH dan hh dipengaruhi oleh *pseudorandom*, mm konstanta dan ss adalah proses. Penerapan kunci privat RSA berhasil mendekripsi blok *cipher* (c) ke kode *American Standard Code for Information Interchange* (ASCII) bukan tunggal karakter atau null dengan *encoding* (UTF-8) dan lama prosesnya bergantung paling utama pada nilai p dan q yang dihasilkan oleh ketentuan, kemudian kondisi kecepatan baca peranti. Hasil GMT dipengaruhi oleh proses membatasi atas prima. Butuh sekitar 239.797 *miliseconds* (ms) untuk entropi $c = 4.814863028233948$ ke 242 kode ASCII dengan $n = 192989$ menjadikan teks awal (8.083 ms nya adalah ASCII ke c) dan 1 sampai 2 detik untuk pembangkitan hingga kunci privat dimana $p = 59$ dan $q = 3271$.

Kata Kunci: Bilangan Prima, Informasi Peranti Waktu, P dan Q , *Android Mobile*

ABSTRACT

Rivest Shamir Adleman (RSA) is a modern cryptographic technique that exceeds the patent limit for 20 years, making it easy to read freely.

Keyword: Prime Number, Information Time Device, P and Q, *Android Mobile*

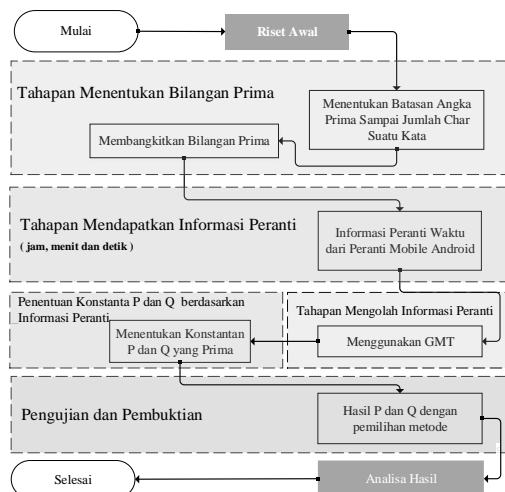
PENDAHULUAN

Bilangan prima adalah bilangan yang hanya memiliki dua faktor: 1 dan bilangan itu sendiri. Satu-satunya bilangan prima bernilai genap hanyalah 2 [1]. 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. $\forall n \in N, n > 1$, 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, maka n memiliki setidaknya 1 faktor prima yang nilainya tidak lebih dari \sqrt{n} . Bilangan prima > 3 memiliki keunikan yang selalu berbentuk antara $6k-1$

atau $6k+1$. 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 didapat mengenai bilangan prima adalah bahwa bilangan prima ada tak hingga banyaknya. Hal ini mungkin terkesan sangat jelas tapi tidak semua orang bisa membuktikan pernyataan ini. Berdasarkan sifat bilangan prima maka penelitian ini menggabungkan informasi peranti waktu pada *android mobile* menjadi teknik penentuan konstanta p dan q juga memastikan bilangan prima yang didapat adalah benar prima.

METODE

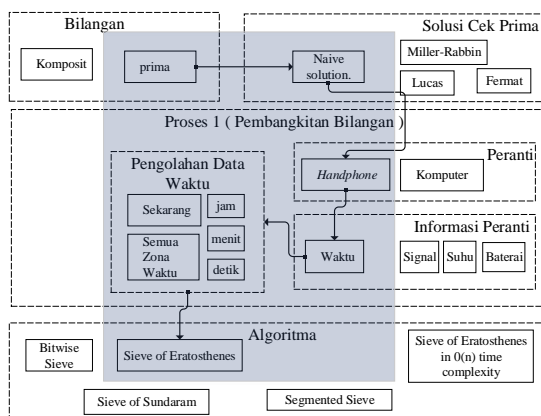
Berdasarkan pendahuluan, pembangkitan dan menentukan konstanta p dan q yang prima maka penelitian menggunakan informasi peranti dapat digambarkan dalam bentuk diagram alir metode penelitian.



Gambar 1. Diagram Alir Metode Penelitian

Kerangka Konsep Penelitian

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



Gambar 3. Kerangka Konsep Penelitian

Bilangan Prima merupakan bilangan bulat positif, sifat pembagiannya [2] 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 [3] 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.

HASIL

Hasil proses tahapan menentukan bilangan prima, mendapatkan informasi peranti, mengolah informasi peranti dan penentuan konstanta p dan q berdasarkan informasi peranti, pengujian dan pembuktian dan analisa hasil menggunakan perangkat *visual studio code*, *android studio*, dan *android mobile*.

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*, Misalnya dari kalimat "Politeknik Negeri Samarinda Tahun 2020" Diuraikan menjadi kode *ASCII* yang diperlihatkan pada Tabel 1.

Tabel 1 Hasil Karakter ke *ASCII*

<i>char</i>	P	o	...	n
<i>ASCII</i>	80	111	...	n

Kemudian dengan persamaan 1.1 didapat totalnya = 3400.

$$total = \sum_{i=1}^n U_i \dots \dots \dots (1.1)$$

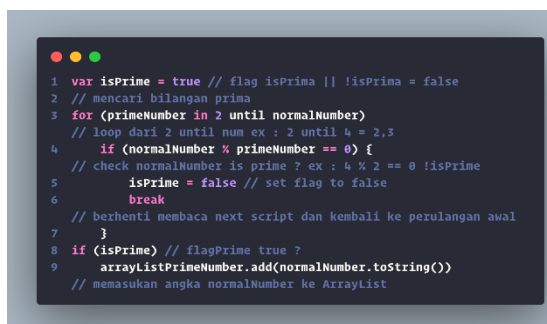
dimana :

Total = Batas Atas Prima

U_i = Nilai Karakter Pada *ASCII*

Matematikawan membuktikan bahwa bilangan prima terbesar itu tidak ada, Pada Juli 2018 bilangan prima ‘terbesar’ ditemukan, yaitu $277.232.917 - 1$ yang diketahui [4]. Proses pembatasan prima mengkonsumsi sebuah waktu yang berhubungan dengan tahapan pengolahan informasi peranti jam, menit dan detik.

- b. Membangkitkan Bilangan Prima dengan mengeliminasi angka bukan prima [5]. misalnya, jika $A = 3$ dan nilai pembaginya (sisanya bagi) $B = 2$, maka ditandai sebagai prima sebaliknya bukan prima. Hasil rentang 1 sampai 3400 membangkitkan 478 angka prima (jumlah angka yang prima) didefinisikan *arrayListPrimeNumber*.



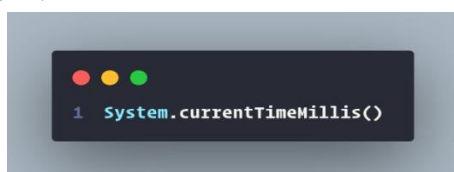
```

1 var isPrime = true // flag isPrime || !isPrime = false
2 // mencari bilangan prima
3 for (primeNumber in 2 until normalNumber)
4   // loop dari 2 until num ex : 2 until 4 = 2,3
5   if (normalNumber % primeNumber == 0) {
6     // check normalNumber is prime ? ex : 4 % 2 == 0 !isPrime
7     isPrime = false // set flag to false
8     break
9   }
10  // berhenti membaca next script dan kembali ke perulangan awal
11 }
12 if (isPrime) // flagPrime true ?
13   arrayListPrimeNumber.add(normalNumber.toString())
14 // memasukan angka normalNumber ke ArrayList
  
```

Gambar 3. Potongan Kode
Membangkitkan Bilangan Prima dengan
Naive Solution

Tahapan Mendapatkan Informasi Peranti

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



```

1 System.currentTimeMillis()
  
```

Gambar 4. Potongan Kode Mendapatkan
Informasi Peranti Waktu Sekarang

Data waktu yang didapat masih berupa nilai keseluruhan waktu yaitu 1594886148236 yang kemudian diformat menjadi (HH:mm:ss) untuk menjadikannya jam, menit dan detik.

Dengan fungsi yang sudah tersedia di Kotlin, dapat digunakan *SimpleDateFormat* sebagai berikut :

```

val dfTime
= SimpleDateFormat("HH:mm:ss")
  
```

Maka didapatkan waktu sekarang 15:55:48 dengan zona awal GMT +8 sebagai Informasi Peranti.

Tahapan Mengolah Informasi Peranti

Informasi Peranti diolah kembali untuk menghasilkan informasi peranti yang probabilistik 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.

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

Gambar 4 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 *SimpleDateFormat* sebagai berikut :

```

val sudoRandom
= (listZoneTime.indices).random()
  
```

Maka hasil nilai *sudoRandom* = 9.

Sehingga didapat *arrayTime* [*sudoRandom*] = GMT -10.

Kemudian dilakukan konversi waktu sekarang 15:55:48 GMT +8 ke GMT -10 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, tujuannya mengkonsumsi sebuah waktu ketika mendapatkan informasi waktu itu sendiri.



Gambar 5. Hasil Zona Awal dan Zona Lain

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 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 $arrayListPrimeNumber = p = q$ 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 $indexP = 36$.

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

$$hh * n = indexP$$

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) atau proses pemfaktoran yang memakan waktu lebih lama.

- b. Menentukan Konstanta Q yang Prima nilai q memiliki aturan mirip dengan nilai p , tetapi memiliki 5 keputusan perhitungan ($q_{keputusan}$) dari 6 ketentuannya ($q_{ketentuan}$).

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

$$K1 = p$$

$$K2 = \text{informasi peranti waktu menit}$$

$$K3 = \text{informasi peranti waktu detik}$$

$$K4 = K2 + K3$$

$$K5 = K1 * K2$$

$$K6 = K2 * K3$$

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

$$K[n] = \begin{cases} n = 0, & K[(jml \text{ prima} - k1)] \\ K1 < Kn \text{ dan } K1 \neq Kn, & K[n] \end{cases}$$

Dimana :

$$K[n] = arrayListPrimeNumber [n]$$

$$jml \text{ prima} = arrayListPrimeNumber.size$$

Dengan persamaan 2.1 dan 2.2 didapat $K1 > K2$ dan mengembalikan nilai $K2$ seperti yang diperlihatkan pada Tabel 2.

Tabel 2 Hasil ($q_{keputusan}$) dan ($q_{ketentuan}$)

informasi peranti waktu			15:05:31
arrayListPrimeNumber [n].size			478
K1	36	$K1 < Kn \ \&\& \ K1 \neq Kn$	RETURN value Kn as [n] as K[n]
K2	55	$K1 < K2 \ \&\& \ K1 \neq K2$	TRUE
K3	49	$K1 < K2 \ \&\& \ K1 \neq K3$	TRUE
K4	104	$K1 < K2 \ \&\& \ K1 \neq K4$	TRUE
K5	495	$K1 < K2 \ \&\& \ K1 \neq K5$	TRUE
K6	2695	$K1 < K2 \ \&\& \ K1 \neq K6$	TRUE
$q_{keputusan}$		K[n]	K[55]
k[n] =		arrayListPrimeNumber [n]	
k[31] =		263	

K2 = 55 sebagai *index* atau [n] dengan begitu $K[n] = K[55] = \text{index}Q$. Maka didapat nilai $q[\text{index}Q] = 263$.

Tahapan ini berhasil menentukan dan menghasilkan $p = 157$ dan $q = 263$ sesuai ketentuan yang ditetapkan dan telah diuji pada pengujian primalitas dengan naive solution dan pembuktian kombinasi dengan informasi peranti dengan metode pemilihan.

Pengujian dan Pembuktian

Pengujian telah dilakukan terhadap hasil p dan q dengan *naive solution*, selain untuk cek bilangan prima sederhana terhadap p dan q , juga bisa digunakan sebagai pembangkit bilangan prima.

Diketahui:

$$p = 157$$

$$q = 263$$

(NaiveSolution).....(3.1)

$$\text{prime} = \begin{cases} n = p, & k[(jml \text{ prima} - k1)] \\ n = q, & k[n] \end{cases}$$

Dimana :

$$n = p \text{ atau } q$$

Maka dengan persamaan 3.1 didapat hasilnya adalah benar atau *true* untuk p dan q , Hasilnya diperlihatkan pada Gambar x. Persamaan 3.1 sama dengan yang diperlihatkan pada Gambar 3.

Pembuktian terhadap persamaan 2.1 dan 2.2 telah dilakukan dengan

NaiveHasilnya kedua konstanta p dan q benar merupakan bilangan prima. Naive solution merupakan hal yang sederhana yang cocok digunakan untuk p dan q yang bernilai

berbagai tahapan metode pemilihan untuk membuktikan pernyataan pembangkitan prima dan kombinasi informasi peranti waktu jam, menit dan detik.

proses pembuktian dilakukan sebagai berikut [1]:

1. Mengumpulkan semua fakta yang ada dari permasalahan.
2. Mengaitkan semua fakta-fakta yang terkumpul dan melihat hal menarik apa yang bisa di dapat. Biasanya dalam proses ini kita cuman mengaitkan satu dua fakta, dan fakta-fakta lain digunakan selanjutnya.
3. Menentukan tujuan. Tentukan apa yang harus dicapai(syarat cukup untuk mengatakan terbukti), untuk membuktikan nilai kebenaran permasalahan.
4. Pemilihan strategi.
5. Eksekusi.
6. Penarikan kesimpulan.

signifikan kemungkinan terjadi ketika terjadi proses yang berlebihan atau kondisi baca peranti itu sendiri. Pada tahapan selanjutnya dua variabel ini menghasilkan sesuatu yang berbeda. // analisis hasil later will move

Hasil *Greatest Common Divisor* (GCD) = 2, menunjukan waktu pemfaktoran semakin lama [6]. //masuk analisa hasil later edit
array – hh).

Ketika Ketentuan (K) tidak terpenuhi mengakibatkan q *null* dan melemparkan

sebuah *NumberFormatException*, pembangkitan kunci tidak berjalan semestinya saat menit (mm) adalah 0 dan detik (ss) berada di bawah nilai P. Sehingga ketentuan *null* ditambahkan

untuk menghindari hal tersebut dan nilainya adalah posisi ukuran *array* – hh, seperti ketika menghindari *index out of bound*

Analisa Hasil

Hasil p dan q yang dibangkitkan berdasarkan informasi peranti waktu jam, menit dan detik merupakan bilangan prima yang rata-rata menghasilkan panjang p dan q sebanyak 7 bit sampai 14 bit selama uji pembangkitan sebanyak 12 kali dalam tempo waktu setiap 5 menit dalam 1 jam dan benar p dan q adalah bagian dari bilangan prima berdasarkan uji primalitas sederhana dengan naive solution yang diperlihatkan pada Gambar X.

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 , rumusnya telah berfungsi untuk setiap bilangan yang dibangkitkan atau ditentukan.



KESIMPULAN

Penelitian dan percobaan terhadap rancangan dan pengujian yang telah dilakukan menghasilkan kesimpulan sebagai berikut:

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, mempengaruhi data waktu.

melakukan perhitungan dan proses lebih cepat (berbeda). Sehingga data waktu dan perhitungan membuat hasil p dan q lebih efisien dengan melihat hasil GCD ($p - 1, q - 1$) tidak terlalu besar dan rentang dua variabel itu sendiri.

DAFTAR PUSTAKA

- [1] Cahyo Dhea Arokhman Yusufi, *Heuristic - For Mathematical Olympiad Approach*. Jakarta: Math Heuristic, 2020.
- [2] F. F. Firmansyah, "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 Ibrahim Malang, 2015.
- [3] R. H. Sari, "Apakah Integrasi Islam dapat Membudayakan Literasi Matematika?," *Semin. Mat. dan Pendidik. Mat. UNY*, hal. 655–662, 2017.
- [4] "Untuk Apa Mencari Bilangan Prima Terbesar? - Anak Bertanya." [Daring]. Tersedia pada: <https://anakbertanya.com/untuk-apa-mencari-bilangan-prima-terbesar/>. [Diakses: 18-Jun-2020].
- [5] A. TH dan B. MB, "The Unique Natural Number Set and Distributed Prime Numbers," *J. Appl. Comput. Math.*, vol. 06, no. 04, 2017, doi: 10.4172/2168-9679.1000368.
- [6] B. S. Muchlis, M. A. Budiman, dan D. Rachmawati, "Teknik Pemecahan Kunci Algoritma Rivest Shamir Adleman (RSA) dengan Metode Kraitchik," *Sinkron*, vol. 2, no. 2, hal. 49–64, 2017.