**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 Suprapty, 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 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 kurung waktu 1 jam, menghasilkan entropi *p*, *q*, konversi *Greenwich* *Mean* *Time* *Zone* (GMT) = , dan ideal acuan data uji adalah . 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 ke 242 kode ASCII dengan 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:** Kunci Privat, RSA, Informasi Peranti, GMT, entropi.

***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:*** *Private Key, RSA, Device Information, GMT, entropy.*

### PENDAHULUAN

Bilangan prima adalah bilangan yang hanya memiliki dua faktor: satu dan bilangan itu sendiri. Satu-satunya bilangan prima yang bernilai genap hanyalah 2. Kemudian akan muncul pertanyaan mengenai apakah 1 bilangan prima? Jawabannya tentu saja tidak. 1 hanya memiliki satu factor pembagi. Kita tidak menghitung 1 sebanyak dua kali. , maka *n* selalu memiliki setidaknya satu factor prima. Setiap bilangan asli lebih dari 1 yang tidak prima disebut bilangan komposit. Jika n adalah suatu bilangan komposit, maka n memiliki setidaknya satu factor prima yang nilainya tidak lebih dari . Bilangan-bilangan prima di atas 3 memiliki keunikan bentuk. Selalu berbentuk antara 6k-1 atau 6k +1. Setiap bilangan hanya memiliki 6 bentuk berikut: 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. Hal ini mungkin terkesan sangat jelas tapi tidak semua orang

bisa membuktikan pernyataan ini ti masa sekarang, tentu memiliki aspek krusial terhadap kombinasi angka atau bilangan yang dilakukan *simple* acak informasi ataupun posisinya. Waktu merupakan sebuah

### METODE

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



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 dan ruang lingkup ilmu pengetahuan.

Gambar 3. Kerangka Konsep Penelitian

### HASIL

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

***Tahapan Menentukan Bilangan Prima***

Pada penelitian [4] bilangan prima merupakan bilangan yang istimewa dalam Al-Qur’an karena dari 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 sendiri.

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

1. 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 Karakter ke *ASCII*

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

Kemudian dengan persamaan 1.1 didapat totalnya = 3400.

..………….….…(1.1)

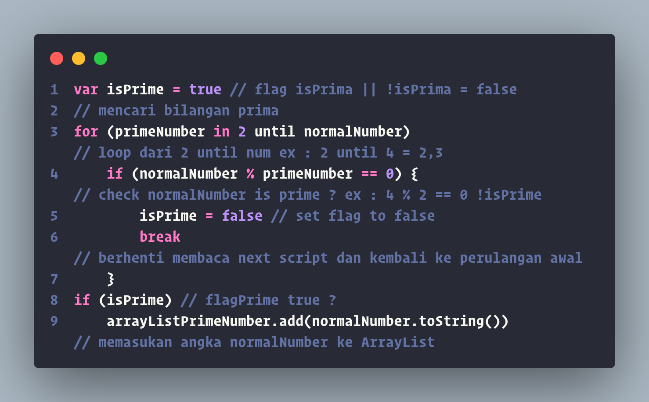
dimana :

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 [5] diatasnya masih ada. Proses pembatasan prima mengkonsumsi sebuah waktu yang berhubungan dengan tahapan pengolahan informasi peranti yaitu jam, menit dan detik.

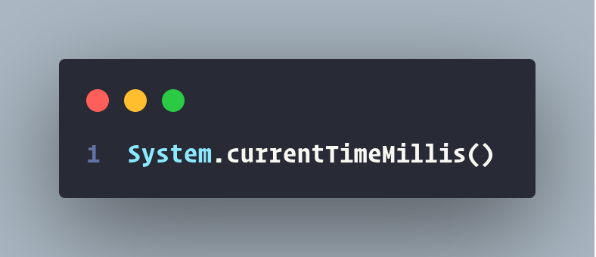
1. Membangkitkan Bilangan Prima dengan mengeliminasi angka bukan prima [6]. misalnya, jika A = 3 dan nilai pembaginya (sisa bagi) B = 2, maka ditandai sebagai prima sebaliknya bukan prima. Hasil rentang 1 sampai 3400 membangkitkan 478 angka prima (jumlah angka yang prima) didefinisikan sebagai *arrayListPrimeNumber*.



Gambar 3. Potongan Kode Membangkitkan Bilangan Prima

***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.



Gambar 4. Mendapatkan Informasi Peranti Waktu

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.

***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 Daftar Waktu Indonesia Tengah

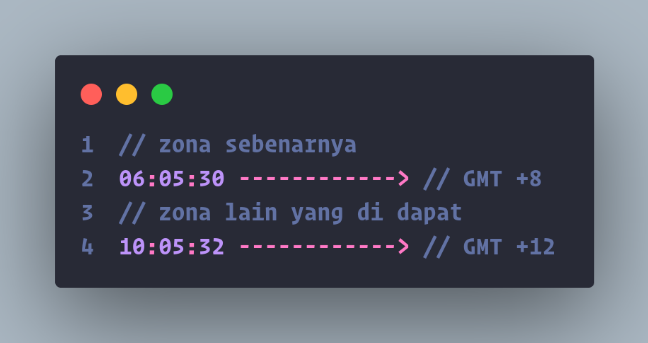
Pemilihan posisi atau *index* untuk *arrayTime* berdasarkan keluaran dari nilai *integer* oleh *sudorandom*, sebagai zona lain.

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

Maka hasil nilai *sudoRandom* = 22.

Sehingga didapat *arrayTime* [*sudoRandom*]= GMT +12 dan digunakan untuk mengkonversi waktu sekarang 06:05:30 GMT +8 menjadi 10:05:31 GMT +12.

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



Gambar 5. Zona Sebenarnya 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* untuk menghasilkan prima yang deterministik dari informasi peranti yang probabilstik:

* 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 hingga ratusan dengan tambahan informasi jam peranti yang berupa nilai.

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.

***Pengujian***

Pengujian telah dilakukan dengan berbagai tahapan,

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

Hasil *Greatest Common Divisor* (GCD) = 2, menunjukan waktu pemfaktoran semakin lama [1]. //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) berapa di bawah nilai P. Sehingga ketentuan *null* ditambahkan untuk menghindari hal tersebut dan nilai nya adalah posisi ukuran *array* – hh, seperti ketika menghindari *index out of bound*.

1. Uji Pembangkitan Kunci

Uji Pembangkitan kunci dilakukan untuk melihat kunci privat yang dibangkitkan oleh *p* dan *q* memiliki ciri waktu sesuai yaitu HH:mm:ss terhadap hh:mm:ss masing-masing konstanta atau berbeda, perubahan zona waktu dipengaruhi secara probabilistik oleh *pseudorandom*. Dengan mencocokan entropi (tingkat data acak/kompresi/*encrypted*). Dapat dilihat rumus entropi sebagai berikut

(1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rentang Waktu Awal**  **Proses (RWAP) ( HH : mm )** | 02:05:31 GMT +8 | 13:57:08 GMT +8 | 14:49:07 GMT +8 | 14:54:10 GMT +8 | 14:59:09 GMT +8 |
| **PEMBANGKITAN KE -** | 1 | 2 | 3 | 4 | 5 |
| **Rentang Waktu Setelah Proses Awal  (RWSPA) ( hh : mm )** | 10:05:32 GMT + 12 | 14:57:09 GMT + 9 | 11:49:08 GMT + 5 | 09:54:11 GMT + 3 | 05:59:10 GMT - 1 |
| P | 73 | 47 | 83 | 67 | 31 |
| Q | 131 | 271 | 197 | 197 | 197 |
| **Entropi RWAP** | 2.2516291673878226 | | | | |
| **Entropi RWSPA** | 2.2516291673878226 | | | | |

Tabel 3 Uji Pembangkitan Kunci pada Hasil Pengujian Pertama Enkripsi dan Dekripsi

Uji awal memiliki acuan bervariasi HH untuk hh, mm konstanta, ss adalah proses pembangkitan dan diuji kembali pada tahap kedua, yang memiliki acuan konstanta HH, yaitu 2.2516291673878226 dan menghasilkan persis oleh ciri waktu yang berbeda untuk masing-masing data maupun keseluruhan.

***Analisa Hasil***

Analisa hasil *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. Dari 5 data menghasilkan nilai entropi P = 2.321928094887362 (semua daftar bilangan adalah berbeda) dan *q* = 1.370950594454668 (3 dari 5 bilangan adalah persis). Hasil *Greatest* *Common* *Divisor* (GCD) konstanta di angka 2. Variabel tersebut melakukan perhitungan algoritma *Rivest Shamir Adleman* (RSA) menghasilkan enkripsi berupa blok *cipher* (c) bernilai entropi 4.814863028233948 dari kode ASCII sepanjang 242 yang juga bernilai sama dengan hasil entropi c. Daftar *binary* antara c dan ASCII memiliki probabilitas berjumlah 58 diperlihatkan pada Gambar 7.

Setiap proses memiliki jalur tersendiri dan dapat diterapkan sesuai keinginan pada setiap atau sebagian proses, sebagai pemberhentian sejenak sehingga mampu menghasilkan ketidakpastian rentang waktu pembangkitan kunci.

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

### KESIMPULAN

Penlitian 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* 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.

Pengujian pertama dengan panjang kunci 2 *bit* sampai 14 *bit*, keduanya telah membangkitkan kunci privat yang mampu mendekripsi kode ASCII. Entropi Blok *CipherText* yaitu 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.

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.

### DAFTAR PUSTAKA

[1] 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.

[2] S. Nisha dan M. Farik, “RSA Public Key Cryptography Algorithm A Review,” *Int. J. Sci. Technol. Res.*, vol. 06, no. 07, hal. 187–191, 2017.

[3] F. N. Pabokory, I. F. Astuti, dan A. H. Kridalaksana, “Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard,” *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 10, no. 1, hal. 20, 2016, doi: 10.30872/jim.v10i1.23.

[4] R. H. Sari, “Apakah Integrasi Islam dapat Membudayakan Literasi Matematika ?,” *Semin. Mat. dan Pendidik. Mat. UNY*, hal. 655–662, 2017.

[5] “Untuk Apa Mencari Bilangan Prima Terbesar? - Anak Bertanya.” [Daring]. Tersedia pada: https://anakbertanya.com/untuk-apa-mencari-bilangan-prima-terbesar/. [Diakses: 18-Jun-2020].

[6] 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.