**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, yanto1294@gmal.com, dan bedirheody@gmail.com*

**ABSTRAK**

Bilangan Prima adalah bilangan yang bila difaktorkan hanya habis dibagi oleh angka 1 dan dengan dirinya sendiri dan lawan dari Bilangan Prima adalah Bilangan Komposit, yaitu bilangan habis dibagi oleh bilangan yang lain lebih dari 2. keunikanya selalu berbentuk antara 6k-1 atau 6k +1. Salah satu konsep atau metode yang menggunakan bilangan prima dimiliki oleh *Rivest Shamir Adleman* (RSA) yang menetapkan 2 buah pola bilangan yang prima untuk membangkitkan kunci RSA. Bilangan prima ditetapkan dalam 2 variabel *p* dan *q,* jika semakin jauh rentang antara *p* dan *q*, maka tidak terlalu besar atau lebih dari 10 menandakan pemfaktoran cukup memakan waktu. Bilangan konstanta atau orde *p* dan *q* menjadi eksperimen aritmatika menggunakan kombinasi informasi peranti waktu pada *android mobile* dalam bentuk jam, menit dan detik. Pembangkitan awal ditentukan dengan batas atas prima *n* = 512. Dengan teknik sederhana *naive solution* dimana menghasilkan angka prima sebanyak 97 atau . Kombinasi peranti waktu jam berperan dalam pembentukan *p* sedangkan *q* dipengaruhi oleh menit dan detik dengan ketentuan yang sedemikian rupa menjadi posisi atau *index*. Waktu yang digunakan adalah ketika terjadi proses aritmatika. *Greenwich Mean Time Zone* (GMT) atau Zona Waktu, kemudian ditentukan berdasarkan probabilistik oleh *pseudorandom* untuk mengubah zona awal 15:05:48 GMT + 8 ke zona lain menjadi 09:05:49 GMT - 10 berdasarkan 24 jenis zona waktu yang ditentukan. *Exception Handling* diterapkan sebagai *monitoring* konsep kombinasi informasi peranti waktu, sehingga didapat hasil ujinya tidak ada pengecualian tangkapan salah satunya adalah *Exception* *NumberFormatException*, maka menandakan aritmatika berhasil dalam menentukan *p = 157* dan *q = 263* dan hasil GCD= 1 sebagai tanda bahwa *p* dan *q* dalam konsep kombinasi informasi peranti waktu juga memiliki hasil yang efisien walaupun penetapan bilangan yang prima tidak besar. Untuk menghasilkan 2 prima yang besar, cukup dengan menaikan nilai konstanta inisial pada aritmatika jam, menit, detik dan batas atas prima *n* yang ditetapkan.

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

***ABSTRACT***

*Prime Numbers are numbers which, when factored, are only divisible by 1 and by themselves and the opposite of Prime Numbers are Composite Numbers, i.e., those numbers are divided by other numbers more than 2. The uniqueness is always in the form of 6k-1 or 6k +1. One concept or method that uses prime numbers is owned by Rivest Shamir Adleman (RSA) which sets 2 prime number patterns to generate RSA keys. Prime numbers are specified in 2 variables p and q, if the farther range between p and q, then GCD (p - 1, q - 1) is not too large or more than 10 indicates factoring is quite time consuming. Constant numbers or order p and q become arithmetic experiments using a combination of time device information on android mobile in the form of hours, minutes and seconds. The initial generation is determined by the upper limit prime n = 512. With a simple naive solution technique where 2 to n - 1 produces a prime number of 97 or arrayListPime = 2,3,5,7,9 ... n. The combination of time clock devices plays a role in the formation of p while q is influenced by minutes and seconds with such provisions being positions or indexes. The time used is when the arithmetic process occurs. Greenwich Mean Time Zone (GMT), then determined based on probabilistic by pseudorandom to change the initial zone of 15:05:48 GMT + 8 to another zone to 09:05:49 GMT - 10 based on 24 types of time zones specified. Exception Handling is applied as a monitoring concept of the combination of time device information, so that the test results are obtained, there are no exceptions, one of which is Exception NumberFormatException, then it indicates that arithmetic is successful in determining p = 157 and q = 263 and the results of GCD = 1 as a sign that p and q in the concept of combining time device information also has efficient results even though the determination of prime numbers is not large. To produce 2 large primes, it is enough to raise the initial constant value in the arithmetic hour, minute, second and the upper limit of the prima specified.*

***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]. Setiap bilangan asli lebih dari 1 yang tidak prima disebut bilangan komposit [2]. Jika n adalah suatu bilangan komposit, maka n memiliki setidaknya 1 faktor prima yang nilainya tidak lebih dari . Bilangan prima yang lebih besar dari 3 memiliki keunikan yang selalu berbentuk antara [3] 6k-1 atau 6k +1 [4], 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 [5] [6]. Dengan ditemukannya bilangan prima, teori bilangan berkembang semakin jauh dan lebih mendalam. Banyak dalil dan sifat dikembangkan berdasarkan bilangan prima [7], 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 [8], 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 [9]. 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 *NumberFormatException*.

### METODE

Berdasarkan pendahuluan, pembangkitan dan menentukan konstanta *p* dan *q* yangprima maka penelitian menggunakan informasi peranti dapat digambarkan dalam bentuk diagram alir metode penelitian yang diperlihatkan pada Gambar 1.



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 2. Kerangka Konsep Penelitian

Bilangan Prima merupakan bilangan bulat positif, sifat pembagiannya [7] melahirkan konsep-konsep aritmetika modulo, dan salah satu konsep bilangan bulat yang digunakan dalam penghitungan komputer.

Pada penelitian [10] 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 [11] Pengecualian atau *Exception Handling* merupakan cara bersih memeriksa kesalahan tanpa mengacaukan kode dan mampu menangkap pengecualian sebuah aritmatika salah satunya *NumberFormat Exception*. Klausa tangkapan diikuti blok coba (*try and catch*), setiap blok tangkapan merupakan pengecualian yang menangani jenis pengecualian, Pengecualian ini cocok digunakan pada Pengujian dan Pembuktian terhadap proses Tahapan Penentuan Konstanta P dan Q berdasarkan Informasi Peranti.

### 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 Membangkitkan Bilangan Prima***

Membangkitkan Bilangan Prima dilakukan dengan mengeliminasi angka bukan prima [13]. Penerapan sederhananya 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*.

Dimana:

*n* = batas atas prima

*n* = 512

Nilai n telah ditentukan sebelumnya. Hasil rentang 1 sampai *n* diperlihatkan pada Tabel 1.



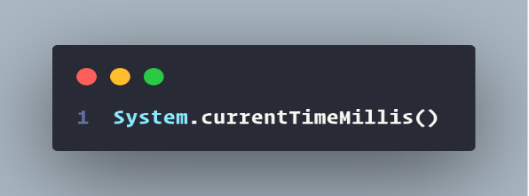
Gambar 3*. FlowChart Naive Solution*

Tabel 1 Hasil Bilangan Prima dengan Naive Solution



***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. Potongan Kode *Kotlin* Mendapatkan Informasi Peranti Waktu Sekarang

Data waktu yang didapat masih berupa nilai keseluruhan waktu yaitu 1594886148236 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 15:55:48 sebagai Informasi Peranti dengan zona awal yang didapat GMT +8.

***Tahapan Mengolah Informasi Peranti***

Informasi Peranti diolah kembali untuk menghasilkan informasi 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.

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 :

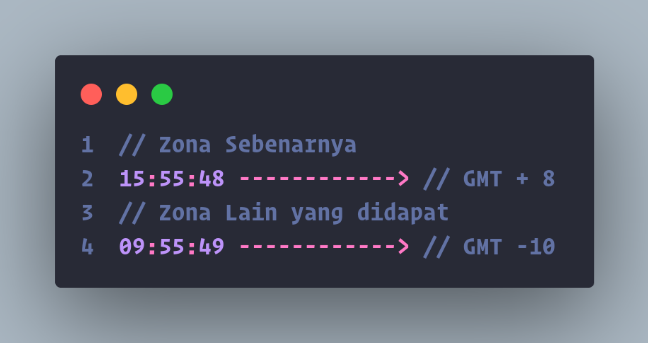
Maka hasil nilai *sudoRandom* = 9. Sehingga didapat *arrayTime* [*sudoRandom*]= GMT -10.

Kemudian dilakukan konversi waktu sekarang 15:55:48 GMT +8 ke GMT -10 yang diperlihatkan pada Gambar 5 dan Gambar 6.

Informasi yang digunakan adalah zona lain, perubahan zona sendiri merupakan proses, tujuanya mengkonsumsi sebuah waktu ketika mendapatkan informasi waktu itu sendiri setelah terjadi aritmatika pembangkitan bilangan prima.



Gambar 5. Potongan Kode *Kotlin* Konversi Zona Waktu



Gambar 6. Hasil Zona Awal dan Zona Lain

***Tahapan 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 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* = 36.

(………..….……………(1.1)

Dimana :

hh = informasi peranti waktu jam

inisial = 4

Maka didapat nilai *p*[*indexP*] = 157.

Jika *inisial* 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.

* 1. Menentukan Konstanta Q yang Prima nilai *q* memiliki aturan mirip dengan nilai *p*, tetapi memiliki 2 keputusan perhitungan () dari 2 ketentuanya ().

()…………………..…(2.1)

K1 = informasi peranti waktu menit

K2 = informasi peranti waktu detik

()…………….……….(2.2)

Dimana :

K[n] = *arrayListPrimeNumber* [n]

jml prima = *arrayListPrimeNumber.size*

Dengan persamaan 2.1 dan 2.2 didapat K1 < K2 dan mengembalikan nilai K2 seperti yang diperlihatkan pada Tabel 3.

Tabel 3 Hasil () dan ()



K2 = 55 sebagai *index* atau [n] dengan begitu K[n] = K[55] = *indexQ,* Maka didapat nilai *q*[*indexQ*] = 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 terhadap Tahapan Penentuan Konstanta P dan Q Berdasarkan Informasi Peranti dengan *Exception Handling*.

***Pengujian dan Pembuktian***

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

Pada Tahapan Menentukan Bilangan Prima yaitu pada Membangkitkan Bilangan Prima tepatnya Gambar 3 diprogramkan seperti yang diperlihatkan Gambar 6



Gambar 6. Potongan Kode *Kotlin* Membangkitkan Bilangan Prima dengan *Naive Solution*

Kemudian dari hasilnya adalah benar sebuah bilangan prima yang setiap nilainya adalah *true*.

Berdasarkan hasil Tahapan Penentuan Konstanta P dan Q Berdasarkan Informasi Peranti.

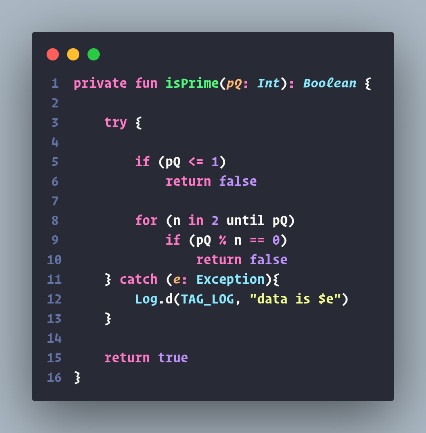
Penerapan *exception handling* berhasil tidak menangkap pengecualian dalam konsep penentuannya, maupun pengecualian secara menyeluruh.



Gambar 7 Potongan Kode *Kotlin* Pengecualian Proses

***Analisa Hasil***

Hasil *p* dan *q* yang dibangkitkan berdasarkan informasi peranti waku 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 8 yang tidak jauh berbeda dengan Gambar 6.



Gambar 8. Potongan Kode *Kotlin* Cek Prima *Naive Solution*

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

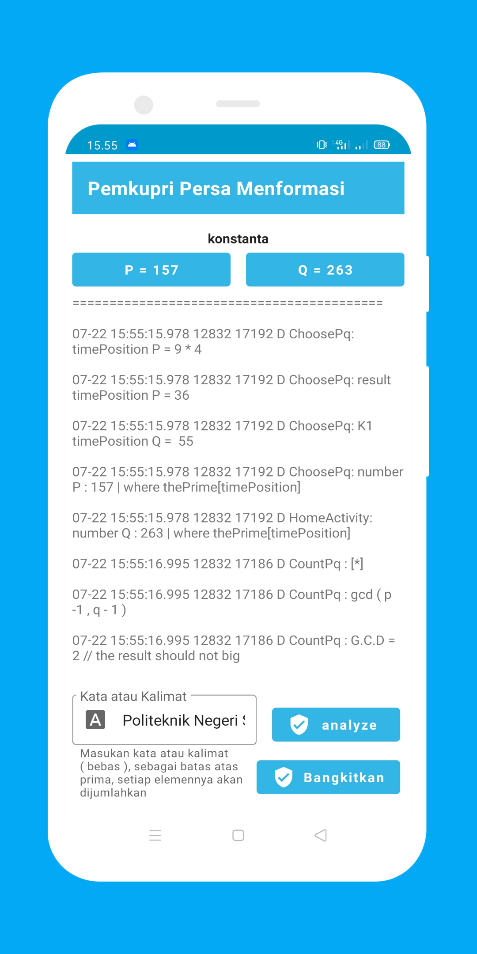
Hasil pada penilitian [14] membuktikan bahwa bahasa pemrograman *kotlin* dapat mengurangi waktu kompilasi, waktu eksekusi dan dapat meningka keringkasan. Maka berdasarkan hal tersebut kemampuan konsep sederhana ini menjadi efisien.

### KESIMPULAN

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

Proses mendapatkan waktu ketika terjadi aritmatika yang diterapkan bergantung peranti yang digunakan, ketika peranti memiliki ruang *memory* pengunaan yang besar, mempengaruhi data waktu.

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 seperti yang diperlihatkan pada Gambar 9.



Gambar 8 Hasil GCD P dan Q

Pemanfaatan zona waktu menghasilkan 2 jenis ketentuan yang terhubung yaitu bilangan prima dan ketentuannya deterministik berdasarkan batas atas dan ketentuan *p* dan *q* nya dapat dikatakan probabilistik karena bergantung pada hasil keluaran *pseudorandom* dalam memilih zona lain dimana 15:55:48 GMT +8 menjadi 09:55:49 GMT – 10.

### DAFTAR PUSTAKA

[1] Cahyo Dhea Arokhman Yusufi, *Heuristic - For Mathematical Olympiad Approach*. Jakarta: Math Heuristic, 2020.

[2] M. K. Harahap, “Membangkitkan Bilangan Prima Marsenne dengan metode Bilangan Prima Probabilistik Solovay – Strassen,” vol. 1, no. Oktober, 2019.

[3] K. Chiewchanchairat, P. Bumroongsri, dan S. Kheawhom, “Improving fermat factorization algorithm by dividing modulus into three forms,” *KKU Eng. J.*, vol. 40, no. March, hal. 131–138, 2016, doi: 10.14456/kkuenj.2015.1.

[4] J. W. P. Ferreira, “The Pattern of Prime Numbers,” *Appl. Math.*, vol. 08, no. 02, hal. 180–192, 2017, doi: 10.4236/am.2017.82015.

[5] T. Sciences, “Dirichlet ’ s Theorem Related Prime Gap,” vol. 10, hal. 305–310, 2016.

[6] R. Meštrović, *Euclid’s theorem on the infinitude of primes: a historical survey of its proofs (300 B.C.--2017) and another new proof*. 2018.

[7] 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 Ibrahuim Malang, 2015.

[8] D. Y. Sylfania, F. P. Juniawan, L. Laurentinus, dan H. A. Pradana, “SMS Security Improvement using RSA in Complaints Application on Regional Head Election’s Fraud,” *J. Teknol. dan Sist. Komput.*, vol. 7, no. 3, hal. 116–120, 2019, doi: 10.14710/jtsiskom.7.3.2019.116-120.

[9] K. Yan, “A Review of the Development and Applications of Number Theory,” *J. Phys. Conf. Ser.*, vol. 1325, no. 1, 2019, doi: 10.1088/1742-6596/1325/1/012128.

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

[11] J. Kumari, S. Singh, dan A. Saxena, “An Exception Monitoring Using Java,” vol. 3, no. 2, hal. 12–18, 2015.

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

[13] 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.

[14] M. J. Arockiajeyanthi, ] T Mrs, dan Kamaleswari, “KOTLIN-A New Programming Language for the Modern Needs,” *Int. J. Sci. Eng. Manag.*, vol. 2, no. 12, hal. 2456–1304, 2017.