RABIT : Jurnal Teknologi dan Sistem Informasi Univrab
Volume xx No. xx | Januari xxx : Hal :xx-xx
ISSN CETAK : 2477-2062
ISSN ONLINE : 2502-891X

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
Jika difaktorkan hanya habis dibagi oleh angka 1 dan dengan dirinya sendiri disebut Bilangan Prima. Keunikanya selalu berbentuk antara 6k-1 atau 6k +1. Salah satu konsep atau metode yang mengolah bilangan yang prima dimiliki oleh *Rivest Shamir Adleman* (RSA) yang menetapkan 2 buah pola yaitu 2 variabel *p* dan *q* untuk membangkitkan kunci RSA. Hasil GCD $(p-1, q-1)$ tidak terlalu besar menandakan pemfaktoran memakan waktu dan rentang *p* dan *q* berjarak jauh. Bilangan konstanta atau orde *p* dan *q* menjadi eksperimen aritmatika menggunakan kombinasi informasi peranti waktu pada *android mobile* dengan bentuk jam (HH), menit (mm) dan detik (ss). *Greenwich Mean Time Zone* (GMT) merupakan bentuk zona waktu informasi yang menjadikan jam berpola determisitik menjadi probabilistik yang diolah oleh *pseudorandom* menghasilkan Zona Awal 15:17:02 GMT + 8 ke Zona Lain menjadi 10:17:03 GMT - 11. Waktu yang digunakan ketika terjadinya proses aritmatika. Kemudian dengan kombinasi peranti waktu, 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* menandakan bahwa *p* dan *q* juga memiliki hasil yang efisien walaupun penetapan bilangan yang prima tidak besar. Pengujianya dilakukan dengan teknik *Exception Handling* sebagai *monitoring* konsep, sehingga hasil ujinya tidak ditemukan pengecualian tangkapan. Hasil Prima yang besar, dapat dihasilkan dengan menaikan nilai pada konstanta *inisial* dan *n* yang ditetapkan pada rumus $P_{penentuan}$ dan $q_{penentuan}$.

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

ABSTRACT
If factored, it is only divisible by the number 1 and by itself is called a Prima Number. Its uniqueness is always in the form between 6k-1 or 6k +1. One of the concepts or methods of processing prime numbers is owned by Rivest Shamir Adleman (RSA) which sets 2 patterns, namely 2 variables p and q to generate RSA keys. GCD results (p-1, q-1) are not too large, indicating factoring takes time and the range of p and q is far apart. 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 (HH), minutes (mm) and seconds (ss). Greenwich Mean Time Zone (GMT) is a form of time zone of information that makes a determinant pattern clock into a probabilistic process that is processed by pseudorandom resulting in an Early Zone 15:17:02 GMT + 8 to Other Zones to 10:17:03 GMT - 11. The time used when arithmetic processes occur. Then with a combination of time devices, HH plays a role in the formation of p while q is influenced by mm and ss with the provisions as such an index. The initial generation is determined by the upper limit prime n = 512. With a simple naive solution technique where 2 to n - 1 produces an arrayListPrimeNumber = 2,3,5,7,9 ... n. Combination and Arithmetic managed to determine p = 179 and q = 419 indicating that p and q also have efficient results even though the determination of prime numbers is not large. The test was carried out using the Exception Handling technique as a concept monitoring, so that the test results were not found to catch exceptions. Large Prima results can be generated by raising the value of the initial constants and n determined in the formula $P_{penentuan}$ *and* $q_{penentuan}$.

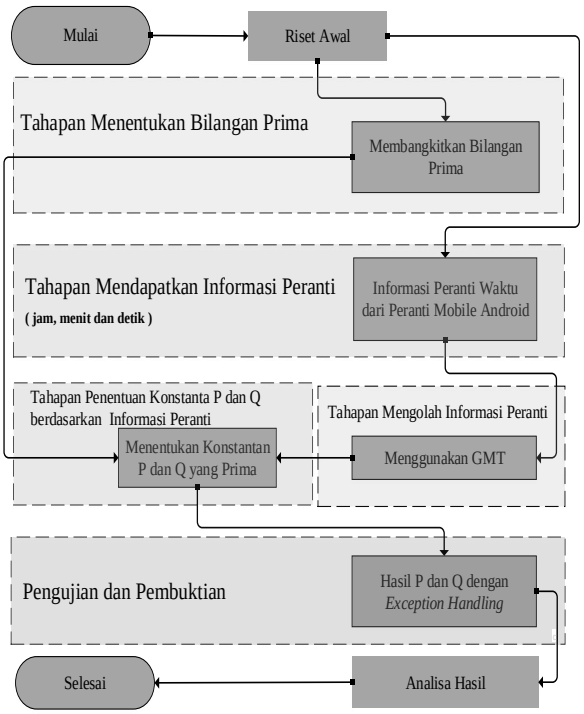
Keyword: Prime Number, Information Time Device, P and Q

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 $\sqrt{n}$. 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]. 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 [8] 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 [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 [9], 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 [10]. Pada 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. 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.

METODE

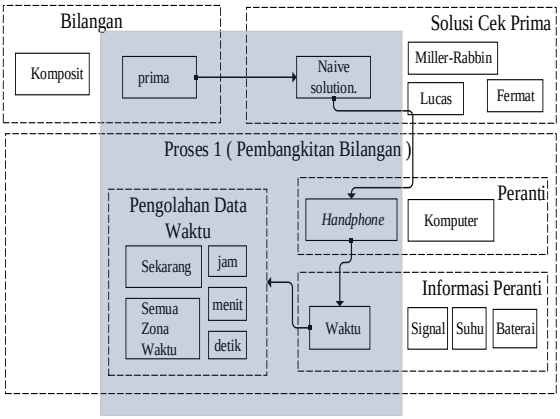
Berdasarkan pendahuluan, pembangkitan dan menentukan konstanta *p* dan *q* yang prima maka penelitian menggunakan informasi peranti dapat digambarkan dalam bentuk diagram alir metode penelitian yang diperlihatkan pada Gambar 1.



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

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*.

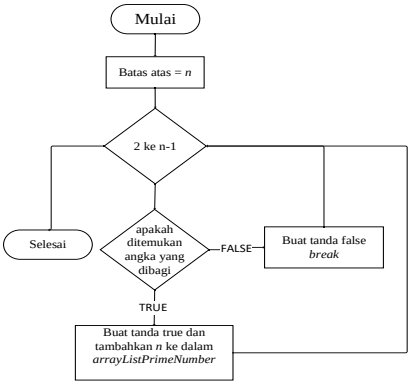
Tahapan Membangkitkan Bilangan Prima

Menentukan atau Membangkitkan Bilangan Prima dilakukan dengan mengeliminasi angka bukan prima [12]. 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 Pembangkitan Bilangan Prima rentang 1 sampai *n* diperlihatkan pada Tabel 1.



Gambar 3. *FlowChart Naive Solution*

Tabel 1 Hasil Pembangkitan Bilangan Prima

arrayListPrimeNumber	prima	2	3	5	..	509
	size	1	2	3	..	97

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.

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 :

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

Maka didapatkan waktu sekarang 15:17:02 sebagai Informasi Peranti dengan Zona Awal (ZA) yang didapat GMT +8.

Tahapan Mengolah Informasi Peranti

Informasi Peranti diolah kembali untuk menghasilkan informasi yang probabilstik dengan menggunakan *pseudorandom* dalam menentukan Zona Waktu *Greenwich Mean Time Zone* (GMT).

Seluruh zona waktu telah didefinisikan sebelumnya ke dalam *arrayTime* sebagai Zona Lain (ZL) yang diperlihatkan pada Tabel 2.

Tabel 2 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	

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

$$val\ sudoRandom = (listZoneTime.indices).random()$$

Maka hasil nilai Zona *Index* (*Zi*) atau *sudoRandom* = 10. Sehingga didapat *arrayTime* [*sudoRandom*] = GMT -11.

Kemudian dilakukan konversi waktu sekarang 15:17:02 GMT +8 ke GMT -11 yang diperlihatkan pada Gambar 5 dan Gambar 6.

```
@SuppressLint("SimpleDateFormat")
fun toGMTFormat(date: String, time: String, zoneTime: String): Date? {
    val formatter = SimpleDateFormat("yy-MM-dd HH:mm:ss")
    formatter.timeZone = TimeZone.getTimeZone(zoneTime)
    val dateTime = "$date $time"
    return formatter.parse(dateTime)
}
```

Gambar 5. Potongan Kode *Kotlin* Konversi Zona Waktu

```
// Zona Awal
15:17:02 ---HH:mm:ss----> // GMT + 8
// Zona Lain
10:17:03 ---HH:mm:ss----> // GMT - 11
```

Gambar 6. Hasil Zona Awal dan Zona Lain

Informasi yang digunakan adalah zona lain, perubahan zona sendiri merupakan proses, tujuanya mengkonsumsi sebuah waktu ketika mendapatkan informasi waktu itu sendiri, selain itu membuat Informasi Peranti Waktu zona awal terlihat deterministik menjadi probabilistik ketika menjadi 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 Menentukan Bilangan Prima.
2. Informasi Peranti telah didapatkan dalam bentuk jam:menit:detik (HH:mm:ss) seperti yang diperlihatkan pada Gambar 6.

Kemudian tahapan penentuan *p* dan *q* dapat diproses lebih lanjut.

a. Menentukan Konstanta P yang Prima , Penentuan ini sederhana, dengan menghitung persamaan 1.1 didapat *i* = 40.

$$(P_{penentuan} \quad Pi = hh * inisial \quad (1.1)$$

Dimana :

Pi = *List Array* Ke-*i*
i = *Index* $arrayListPrimeNumber$
inisial = 4
hh = Informasi Peranti Waktu Jam

Maka untuk nilai *Pi* = 179.

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 - 1, q - 1*) yang hasilnya kecil atau proses pemfaktoran yang memakan waktu [13] .

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

($q_{ketentuan}$)……..…(2.1)
K1 = informasi peranti waktu menit
K2 = informasi peranti waktu detik

($q_{keputusan}$)…….(2.2)

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

Dimana :
Qi = *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 3.

Tabel 3 Hasil ($q_{keputusan}$) dan ($q_{ketentuan}$)

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

Maka untuk nilai *Qi* = 419.

Tahapan ini berhasil menentukan dan menghasilkan $Pi = 179$ dan $Qi = 419$, sesuai ketentuan yang ditetapkan dan telah diuji pada pengujian primalitas dengan *naive solution* dan pembuktian terhadap Penentuan Konstanta P dan Q Berdasarkan Informasi Peranti dengan *Exception Handling* dan Analisa Hasil terhadap pembangkitan setiap 5 menit untuk melihat performa konsep dan hasil efisien *p* dan *q*.

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 Membangkitkan Bilangan Prima tepatnya Gambar 3 diprogramkan seperti yang diperlihatkan pada Gambar 6

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

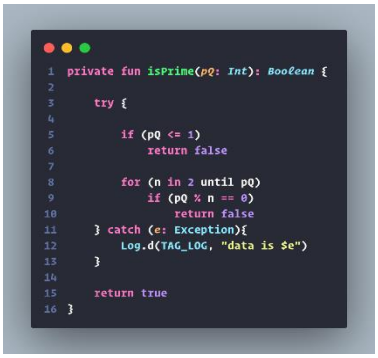
Kemudian dari hasilnya adalah benar sebuah bilangan prima yang setiap nilainya adalah *true*.

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 9 yang tidak jauh berbeda dengan Gambar 6.



```
private fun isPrime(pQ: Int): Boolean {

    try {

        if (pQ <= 1)
            return false

        for (n in 2 until pQ)
            if (pQ % n == 0)
                return false
    } catch (e: Exception){
        Log.d(TAG_LOG, "data is $e")
    }

    return true
}
```

Gambar 9. 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* yang diperlihatkan pada Gambar 7, rumusnya telah berfungsi untuk setiap bilangan yang dibangkitkan atau ditentukan.

Ujianalisis Seluruh P dan Q Rentang 5 Menit Zona Awal(ZA) Zona Lain(ZL) Zona Index(Zi)

Tabel 4 Hasil Pembangkitan P dan Q setiap 5 menit

DATA	ZA (HH:mm:ss)	ZL (hh:mm:ss)	*sudoRandom* Zi	*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

http://jurnal.polgan.ac.id/index.php/sinkron/article/view/75

Hasil pada penilitian [14] membuktikan bahwa bahasa pemrograman *kotlin* dapat mengurangi waktu kompilasi, waktu eksekusi dan dapat meningkatkan 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 pembangkitan bilangan prima mempengaruhi data waktu yang didapat dan perhitungan menghasilkan *p* dan *q* akan lebih efisien dengan melihat hasil GCD $(p-1, q-1)$ tidak terlalu besar dan rentang dua variabel itu sendiri.

Variabel penting yang berperan mempengaruhi besar kecilnya bilangan prima yang dibangkitkan ataupun ditentukannya p dan q, dipengaruhi oleh nilai variabel *inisial* dan batas atas prima *n*.

Pemanfaatan zona waktu menghasilkan 2 jenis konsep penentuan yaitu deterministik dan probabilistik dimana hasil zona awal 15:17:02 GMT +8 didapat ketika terjadi aritmatika pembangkitan bilangan prima dan zona lain berdasarkan keluaran *pseudorandom* dalam memilih daftar zona lain (*ArrayTime*) dimana hasilnya adalah 10:17:03 GMT-11.

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[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.