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 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 :

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 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 menjadikanya berpola determisitik 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 menaikan nilai *inisial* dan *n* yang ditetapkan pada rumus $P_{penentuan}$ dan $q_{penentuan}$. Seluruh proses diuji keberhasilanya dengan *monitoring* pengecualian atau *Exception Handling,* hasilnya tidak ada *Logcat Android Studio* berupa pengecualian 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 by monitoring exceptions or Exception Handling, the result is no Logcat Android Studio in the form of an exception and uses kotlin as compiler*

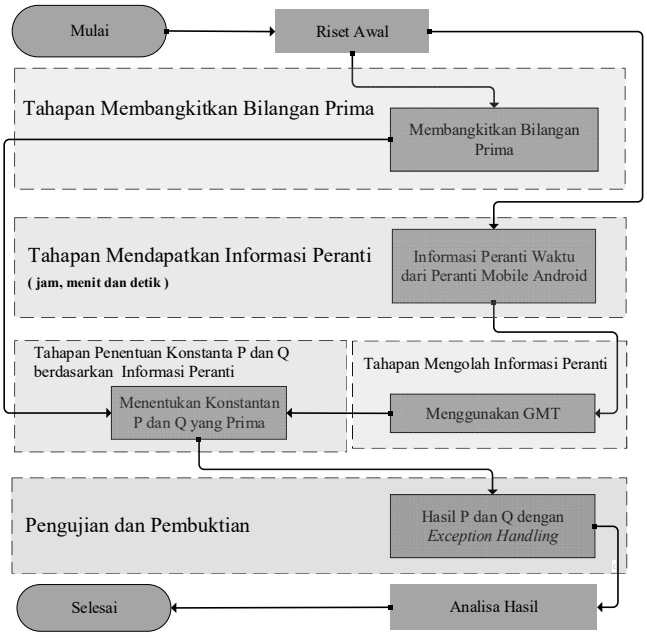
Keyword: Prime Number, Information Time Device, P and Q

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

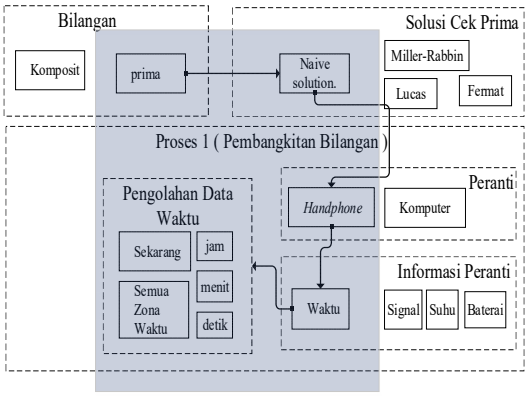
Berdasarkan pendahuluan, pembangkitan dan menentukan konstanta *p* dan *q* yang prima maka penelitian menggunakan informasi peranti dapat digambarkan dalam bentuk diagram alur metode penelitian yang diperlihatkan pada Gambar 1.



Gambar 1. Diagram Alur 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*.

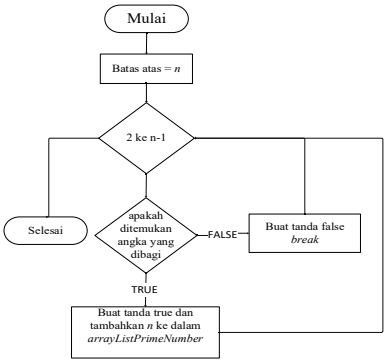
Tahapan Membangkitkan Bilangan Prima

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* 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.

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

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.

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

Gambar 4. Hasil Informasi Peranti Waktu

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

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 dengan menggabungkan syaratnya, syarat dua 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 $i = 40$.

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

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 ($q_{ketentuan}$).

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

K1 = informasi peranti waktu menit
K2 = informasi peranti waktu detik

$$(q_{keputusan}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.2)$$
$$Qi = \begin{cases} inisial * (K1 * K2)\ mod\ q.size, & K1 < K2 \\ inisial * (K1 * K2)\ mod\ q.size, & K1 > K2 \end{cases}$$

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 $Qi = 419$.

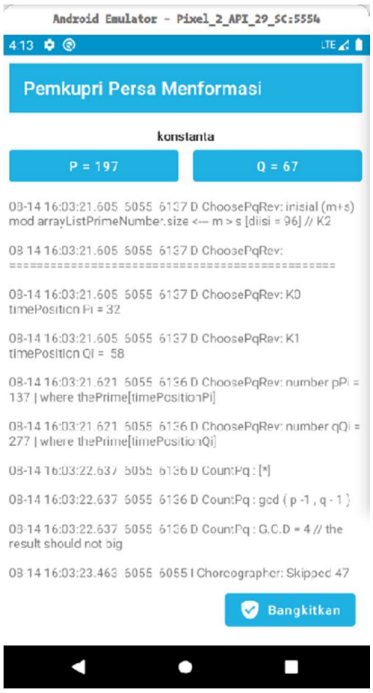
Tahapan ini berhasil menentukan dan menghasilkan $Pi = 179$ dan $Qi = 419$, sesuai ketentuan yang ditetapkan dan telah diuji pada pengujian primalitas dengan *naive solution* dan pembuktian terhadap Penentuan Konstanta P dan Q Berdasarkan Informasi Peranti dengan *Exception Handling* dan Analisa Hasil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

Pengujian dan Pembuktian

Pengujian 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 p dan q dalam tempo 300000 milidetik selama 1 jam. Hasilnya diperlihat pada Tabel 4 dan Gambar 5.

Tabel 4 Hasil Pembangkitan P dan Q

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
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 5 Aplikasi Pembangkitan P dan Q

Analisa Hasil

Berdasarkan Tabel 4 *Greatest Common Divisor* GCD(p-1 , q-1) dan Gambar 6, rata-rata bernilai 2.

Gambar 6 Hasil GCD Pembangkitan

Berdasarkan pendapat (Muchlis dkk., 2017) bilangan yang prima dan memiliki hasil GCD kecil memiliki proses pemfaktoran 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.

Berdasarkan [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.

KESIMPULAN

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 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.

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.

Hasil proses kombinasi berdasarkan infomasi peranti waktu diuji keberhasilanya dengan monitoring pengecualian atau Exception Handling yang menghasilkan tidak ada Logcat Android Studio berupa pengecualian di seluruh pemrosesan.

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.