Analisis Implementasi Teori Kombinatorika dan Teknik Kriptografi pada Mesin SIGABA

Amalia Putri - 13522042
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13522042@std.stei.itb.ac.id

***Abstract*—Mesin SIGABA merupakan mesin *cipher* milik Amerika Serikat yang digunakan untuk mengenkripsi dan mendekripsi suatu pesan agar pesan yang berisi informasi penting dapat tersembunyi dan tidak diketahui oleh pihak musuh serta hanya dapat tersampaikan kepada pihak sekutu pada Perang Dunia II. Tingkat kompleksitas mesin SIGABA terbukti sangat tinggi karena hingga Perang Dunia II berakhir, belum ada pihak musuh, yakni Jerman, Jepang, dan Italia, yang berhasil memecahkan *cipher* milik Amerika Serikat. Namun sebaliknya, pihak sekutu, Amerika Serikat dan Inggris berhasil memecahkan *cipher* milik Jerman pada 1940, yakni mesin Enigma. Di balik keberlangsungan komunikasi dan penyampaian informasi dengan mesin SIGABA, terdapat teori Kombinatorika yang digunakan untuk mengembangkan metode pengacakan kunci yang sulit dipecahkan, sementara teknik Kriptografi digunakan untuk mengenkripsi dan mendekripsi pesan dengan tingkat keamanan yang tinggi.**

***Keywords*—*cipher*, dekripsi, enkripsi, Kombinatorika, Kriptografi**

I. Pendahuluan

Selama masa Perang Dunia II, penyebaran informasi dan komunikasi melalui radio nirkabel sangatlah penting untuk militer. Akan tetapi,informasi rahasia tersebut dapat disadap dan disebar ke pihak lawan sehingga komunikasi dan informasi harus dijaga dalam kode rahasia. Hal ini dilakukan dengan mengonversikan huruf-huruf dalam teks biasa menjadi suatu kode yang dirahasiakan (enkripsi) dan hanya dapat diakses serta diketahui oleh pihak sekutu.

Negara-negara seperti Jerman, Amerika Serikat, dan Inggris mempunyai mesin *cipher* yang masing-masing terdapat ciri khas dan perbedaan dari segi mekaniknya. Seiring perperangan berlangsung, pihak lawan dan pihak sekutu saling berusaha untuk memecahkan kode mesin *cipher* dengan ahli matematika yang mereka miliki. Akhirnya, pada tahun 1930, seorang ahli kritografi dari Polandia dan Inggris bersama seorang penghianat Jerman berhasil memecahkan (dekripsi) pola pengacakannya.

Sebaliknya, pihak lawan, yakni Jerman juga berusaha untuk memecahkan kode mesin SIGABA, namun hal itu terlalu sulit bagi Jerman dan hingga berakhirnya Perang Dunia II, mesin SIGABA tidak berhasil mereka pecahkan kode atau pola pengacakannya. Hal ini disebabkan oleh kemampuan mesin SIGABA yang dari segi mekanik dan kriptografi lebih kompleks dibandingkan dari mesin Enigma yang dimiliki oleh Jerman.

II. Kerangka Teori

2.1 Mesin SIGABA

Mesin SIGABA atau yang disebut sebagai ECM (*Electric Cipher Machine*) Mark II adalah mesin dengan rotor yang berotasi untuk mengenkripsi dan mendekripsi pesan. Rotor ini dapat disesuaikan dengan cara melepas dan menggantinya dengan rotor yang lain. Cara kerja mesin SIGABA mirip dengan mesin Enigma, namun mesin SIGABA memiliki keunggulan, yakni penggunaan 15 rotor untuk mengenkripsi pesan, sedangkan mesin Enigma hanya memiliki tiga rotor [1]. Dari 15 rotor tersebut, dibagi menjadi tiga jenis rotor, yakni lima rotor *cipher*, lima rotor kontrol, dan lima indeks rotor.

Gambar 2.1.1: Komponen Mesin SIGABA [1]

Secara teoretis, mesin SIGABA memiliki kombinasi pada ke-15 rotornya dan masing-masing jenisnya, yakni pada rotor *cipher*, rotor kontrol, dan rotor indeks memiliki 5 rotor. Pada rotor *cipher* dan kontrol memutasikan 26 huruf alfabet, yang artinya tiap rotor *cipher* dan kontrol memiliki 26! kemungkinan permutasinya. Namun, pada rotor indeks memutasikan suatu digit ke sejumlah digit lainnya, artinya tiap rotor indeks memiliki 10! kemungkinan permutasinya. Hal ini menunjukkan bahwa kombinasi permutasi ini bernilai $(26!)^5 * (26!)^5 * (10!)^5 \approx 2^{993}$ kunci berbeda pada mesin SIGABA [2]. Terdapat asumsi bahwa posisi awal dari 15 rotor pada kalkulasi ini perlu diabaikan karena mempertimbangkan semua kemungkinan pada kawat rotor sehingga suatu perbedaan posisi awal dianggap senilai dengan kawat rotor lainnya. Oleh karena itu, semua kemungkinan pada kawat rotor diatur pada posisi awal yang telah menjadi standar sehingga menyebabkan rotor indeks tidak melakukan semua tahapan permutasi dalam mesin,

namun dikurangi operasinya hanya menjadi 10! permutasi yang unik. Sehingga nilai kombinasi permutasinya berkurang menjadi $(26!)^{10} * 10! \approx 2^{906}$ [2].

Kunci yang terdapat nampaknya setara dengan kunci *cipher* saat ini, memiliki panjang 906-bit, sejalan dengan *keyspace* teoretis. Panjang ini lebih dari tiga setengah kali lipat dari kunci enkripsi terbesar yang tersedia saat ini, yang hanya sepanjang 256-bit. Jika asumsi ini benar, dapat dijelaskan mengapa kekuatan lawan tidak diketahui pernah berhasil meretas SIGABA selama konflik berlangsung. Namun, perlu dicatat bahwa keberlakuan *keyspace* ini masih memerlukan konfirmasi lebih lanjut. *Keyspace* SIGABA sebenarnya, yakni dengan panjang 906-bit ternyata tidak ditemukan. *Keyspace* sejati mesin ini ternyata dibatasi oleh berbagai variabel selama masa operasionalnya [2].

Oleh sebab itu, pada masa Perang Dunia II, angka rotor indeks yang digunakan harus dirancang menjadi lebih rendah, namun kemampuan enkripsi-dekripsinya tetap optimal sehingga kombinasi yang digunakan, yakni $10! \, . \, 2^5 . 10^5 \approx 2^{48,4}$. Sebuah *keyspace* dengan ukuran $2^{48,4}$ saat ini masih tergolong kecil, sehingga dapat menjadi rentan terhadap pencarian kunci secara menyeluruh. Namun, ukuran *keyspace* sebesar ini pada zaman 1940-an seharusnya tidak dapat ditembus dengan menggunakan teknologi yang tersedia pada masa itu, asalkan tidak ada serangan pintas yang dapat dieksploitasi [3].

2.2 Teori Kombinatorika

Kombinatorial adalah cabang Matematika untuk menghitung jumlah penyusunan objek-objek tanpa harus mengenumerasi semua kemungkinan susunannya, contohnya persoalan jumlah PIN yang dapat dibuat untuk PIN kartu ATM dengan enam digit angka, jumlah buku yang dapat dikodekan dari perpustakaan dengan enam digit angka, dan berapa banyak cara untuk membuat sebuah komisi dengan sejumlah anggota. Dalam Kombinatorial, terdapat perluasan kaidah dasar menghitung, yakni kaidah perkalian $(p_1 \times p_2 \times ... \times p_n)$ dan kaidah penjumlahan $(p_1 + p_2 + \cdots + p_n)$ [4]. Selain itu, dalam pengombinasian suatu kemungkinan dalam konteks permutasi, yakni sebuah konsep penyusunan sekumpulan objek/angka menjadi beberapa urutan berbeda tanpa mengalami pengulangan. Dalam permutasi urutan diperhatikan. Setiap objek yang dihasilkan harus berbeda antara satu dengan yang lain. Sebagai contoh, urutan huruf {ABC} berbeda dengan {CAB} begitu juga dengan {BAC} dan {ACB}. Banyaknya permutasi r unsur dinyatakan P_r^n dengan persamaan (1) berikut [5].

$$P_r^n = \frac{\mathrm{n!}}{\mathrm{(n-r)!}} \qquad r \leq n \qquad (1)$$

$$P_n^n = n! \qquad (2)$$

Permutasi merupakan pengembangan dari aturan perkalian, yakni dengan cara menyusun suatu unsur secara urut dengan objek yang berbeda dari kelompok unsur. Permutasi sekumpulan n dengan yang berlainan diambil secara bersama-sama. Jika n unsur yang tersedia terdapat n_1, n_2, dan n_3 merupakan unsur yang sama, maka banyaknya permutasi yang berlainan dari n adalah $\frac{n!}{n_1!n_2!n_3!}$ dengan $\mathrm{n_1 + n_2 + n_3 \leq n}$ [5].

2.3 Teknik Kriptografi Mesin SIGABA

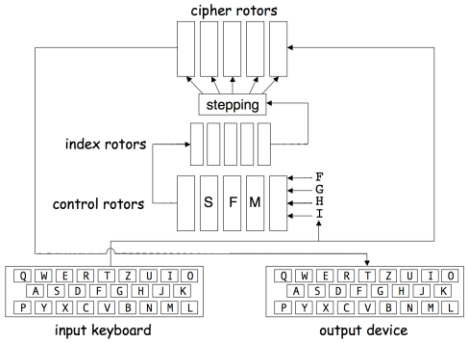
Kriptografi berasal dari Bahasa Yunani, terdiri dari dua suku kata, yaitu *kripto* dan *graphia*. *Kripto* artinya menyembunyikan, sedangkan *graphia* artinya tulisan [6]. Kriptografi adalah ilmu dan seni mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasian data, keabsahan data, integritas data, serta autentikasi data untuk menjaga keamanan pesan dengan cara menyandikannya menjadi bentuk lain yang tidak bermakna agar pesan yang bersifat rahasia tidak dapat dibaca oleh pihak yang tidak berhak [7].

2.2.1 Enkripsi

Enkripsi adalah sebuah proses penyandian yang melakukan perubahan sebuah kode (pesan) dari yang bisa dimengerti (*plaintext*) menjadi sebuah kode yang tidak bisa dimengerti (*ciphertext*). Sedangkan proses kebalikannya untuk mengubah *ciphertext* menjadi *plaintext* disebut dekripsi. Proses enkripsi dan deskripsi memerlukan suatu mekanisme dan kunci tertentu. ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi [7].

Dalam proses enkripsi mesin SIGABA, teks masukan dimasukkan menggunakan *keyboard*. Saat sebuah tombol ditekan pada *keyboard*, sinyal dihasilkan yang dikirim ke dua dari tiga bank rotor dalam kerangka rotor. Sinyal pertama dikirim ke sisi kiri bank rotor *cipher*. Kemudian, sinyal tersebut dipermutasikan melalui lima rotor *cipher* untuk menghasilkan teks sandi. Sinyal kedua dikirim ke sisi kanan bank rotor kontrol. Namun, sinyal ini diatasi dengan cara yang berbeda dari sinyal yang dikirim ke bank rotor *cipher* [2].

Pada rotor kontrol, sinyal digunakan untuk memberdayakan masukan ke dalam bank rotor kontrol. Empat masukan ke bank rotor kontrol diaktifkan saat sebuah tombol ditekan pada *keyboard*. Keempat masukan tersebut selalu 'F', 'G', 'H', dan 'I' tanpa memperhatikan huruf mana yang ditekan pada *keyboard*. Keempat sinyal ini kemudian dipermutasikan melalui rotor kontrol dari kanan ke kiri. Setelah keempat sinyal muncul dari sisi kiri bank rotor kontrol, sinyal tersebut melalui ORing untuk menentukan masukan mana dari bank rotor indeks yang diaktifkan [2].



Gambar 2.2.1.1: Enkripsi Mesin SIGABA [2]

$I_1 = B$ $I_4 = F \,||\, G \,||\, H$ $I_7 = P \,||\, Q \,||\, R \,||\, S \,||\, T$
$I_2 = C$ $I_5 = I \,||\, J \,||\, K$ $I_8 = U \,||\, V \,||\, W \,||\, X \,||\, Y \,||\, Z$
$I_3 = D \,||\, E$ $I_6 = F \,||\, G \,||\, H \,||\, O$ $I_9 = A$

Ij adalah masukan ke-j dari bank rotor indeks. Sebagai contoh, "I_7 = P || Q || R || S || T" berarti masukan ketujuh ke bank rotor indeks aktif jika salah satu dari empat *output* dari bank rotor kontrol adalah P, Q, R, S, atau T. I_0 tidak pernah diaktifkan. Input rotor indeks aktif tersebut hanya berlaku untuk versi mesin CSP-889. Versi selanjutnya, CSP-2900, beroperasi dengan cara yang berbeda. Pada versi tersebut, pemetaan huruf keluaran dari rotor kontrol ke masukan aktif bank indeks berbeda. Selain itu, bukan hanya 'F' 'G', 'H', dan 'I' yang menjadi masukan aktif untuk rotor kontrol, 'D' dan 'E' juga diaktifkan [2].

Setelah setiap huruf yang dimasukkan, satu hingga tiga dari rotor kontrol akan melangkah. Menghitung dari kiri, rotor kontrol cepat adalah rotor ketiga dalam bank rotor kontrol, rotor kontrol medium adalah rotor keempat dalam bank rotor kontrol, dan rotor kontrol lambat adalah rotor kedua dalam bank rotor kontrol. Rotor cepat melangkah sekali untuk setiap huruf yang dimasukkan melalui *keyboard*. Rotor kontrol medium melangkah sekali setiap kali rotor cepat bertransisi dari O ke huruf lain. Untuk orientasi maju, ini akan menjadi transisi dari O ke N. Untuk orientasi mundur, itu akan dari O ke P. Diklaim bahwa untuk rotor yang terbalik, transisi terjadi dari A ke B bukan dari O ke P. Rotor kontrol lambat melangkah sekali setiap kali rotor kontrol medium melakukan transisi dari O ke N dalam orientasi maju atau O ke P dalam orientasi terbalik (*inverse*). Rotor kontrol pertama dan kelima tetap tetap selama operasi dan tidak diubah oleh proses enkripsi seperti rotor kontrol cepat, medium, dan lambat [2].

Karena ORing dari *output* bank rotor kontrol, satu hingga empat masukan bank rotor indeks akan diaktifkan. Sinyal aktif dipermutasikan oleh bank rotor indeks dari kiri ke kanan. Keluaran dari bank rotor indeks kemudian di-OR kembali, meskipun dengan cara yang berbeda, untuk menentukan rotor *cipher* mana yang harus melangkah berdasarkan keluaran bank rotor indeks. Dalam format $C_j = O_x \,||\, O_y$, diperoleh nilai C_j yang berarti rotor *cipher* j akan melangkah jika keluaran dari bank rotor indeks mengandung x atau y [2].

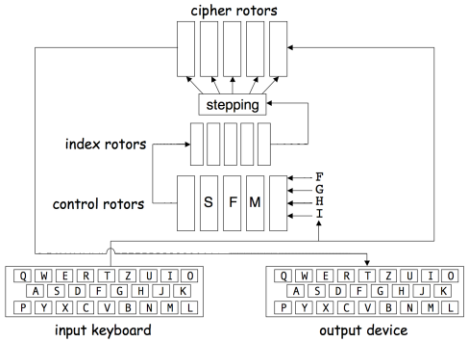
$$C_0 = O_0 \,||\, O_9 \qquad C_2 = O_5 \,||\, O_6 \qquad C_4 = O_1 \,||\, O_2$$
$$C_1 = O_7 \,||\, O_8 \qquad C_3 = O_3 \,||\, O_4$$

Sebuah kekuatan yang menarik dari algoritma enkripsi SIGABA adalah bahwa huruf Z, dan spasi kata diperlakukan sedikit berbeda dibandingkan huruf lain. Huruf lain dikirim ke bank rotor *cipher* tanpa modifikasi. Namun, huruf Z, dan spasi kata dimodifikasi sebelum dikirim ke bank rotor *cipher*. Jika huruf Z dimasukkan pada *keyboard*, itu diubah menjadi X sebelum dikirim ke bank rotor *cipher*. Jika spasi kata dimasukkan pada *keyboard*, itu diubah menjadi huruf Z sebelum dikirim ke bank rotor *cipher* [2].

2.2.2 Dekripsi

Dekripsi merupakan proses yang berlawanan dengan enkripsi, yakni proses mengembalikan *ciphertext* menjadi *plaintext*-nya. Dekripsi menggunakan kunci dekripsi mendapatkan kembali data asli [7]. Dekripsi pada mesin SIGABA mengikuti proses yang mirip dengan enkripsi, dengan beberapa perbedaan kunci. Untuk memulai dekripsi, mesin diinisialisasi menggunakan kunci yang sama seperti dalam proses enkripsi. Namun selama dekripsi, ketika sebuah tombol ditekan pada *keyboard*, sinyal diarahkan ke sisi kanan bank rotor *cipher* daripada sisi kiri, mengubah jalur proses kriptografis [2].

Fitur khas dari dekripsi SIGABA melibatkan perlakuan terhadap huruf Z dan spasi kata. Ketika *output* bank rotor *cipher* sesuai dengan huruf Z, itu mengalami transformasi menjadi spasi sebelum dikirimkan ke perangkat *output*. Aspek menarik untuk dicatat adalah bahwa teks terdekripsi akan konsisten tidak memiliki huruf Z, setiap kejadian Z dalam teks asli diartikan sebagai X selama proses dekripsi. Terdapat proses enkripsi/dekripsi, yakni sebagai berikut [2].



Gambar 2.2.2.1: Dekripsi Mesin SIGABA [2]

Plaintext : ZERO ONE TWO THREE FOUR FIVE SIX
Ciphertext : IEQDEMOKGJEYGOKWBXAIPKRHWARZODWG
Ciphertext terdekripsi : XERO ONE TWO THREE FOUR FIVE SIX

Kasus di atas memberikan gambaran komprehensif tentang transformasi khusus yang terjadi pada huruf Z dan spasi selama tahap enkripsi dan dekripsi. Interaksi rumit dari operasi kriptografis, termasuk penanganan halus terhadap karakter tertentu seperti Z, menunjukkan kompleksitas dan kedalaman mekanisme dekripsi mesin SIGABA. Dengan mengelola variasi ini dengan hati-hati, mesin SIGABA memastikan dekripsi pesan terenkripsi yang aman dan dapat diandalkan, memainkan peran penting dalam menjaga kerahasiaan dan integritas informasi sensitif selama penggunaannya yang operasional [2].

2.4 Algoritma Euclidean

Algoritma Euclidean adalah algoritma yang ditemukan oleh seorang matematikawan Yunani, bernama Euclides. Algoritma ini digunakan untuk mencari pembagi bersama terbesar (PBB) dari dua buah bilangan bulat, contohnya $PBB(a,b) = d$ di mana d merupakan PBB dari a dan b. Berdasarkan teorema Euclidean, algoritma ini menjelaskan bahwa misalkan m dan n adalah bilangan bulat tak negatif dengan $m \geq n$, dan misalkan $r_0 = m$ dan $r_1 = n$, maka dilakukan pembagian secara berturut-turut untuk memperoleh persamaan [8]

$$\begin{aligned} r_0 &= r_1 q_1 + r_2 && 0 \leq r_2 < r_1 \\ r_1 &= r_2 q_2 + r_3 && 0 \leq r_3 < r_2 \\ &\vdots \\ r_{n-2} &= r_{n-1} q_{n-1} + r_n && 0 \leq r_n < r_{n-1} \\ r_{n-1} &= r_n q_n + 0 \end{aligned}$$

Menurut teorema, $PBB(m,n)$ diperoleh sedemikian sehingga

$$
\begin{aligned}
PBB(m,n) &= PBB(r_0, r_1) \\
&= PBB(r_1, r_2) \\
&= \ldots \\
&= PBB(r_{n-2}, r_{n-1}) \\
&= PBB(r_{n-1}, r_n) \\
&= PBB(r_n, 0) \\
&= r_n
\end{aligned}
$$

Proses algoritma Euclidean dapat diimplementasikan pada pemrograman prosedur, ditunjukkan pada **Gambar 2.4.1**.

```
procedure Euclidean(input m, n : integer,
                    output PBB : integer)
{ Mencari PBB(m, n) dengan syarat m dan n bilangan tak-
  negatif dan m ≥ n
  Masukan: m dan n, m ≥ n dan m, n ≥ 0
  Keluaran: PBB(m, n)
}
Kamus
   r : integer

Algoritma:
   while n ≠ 0 do
      r ← m mod n
      m ← n
      n ← r
   endwhile
   { n = 0, maka PBB(m,n) = m }
   PBB ← m
```

Gambar 2.4.1: Algoritma Euclidean dalam Notasi Algoritma [8]

III. Implementasi Cara Kerja Mesin Sigaba dengan Pemrograman

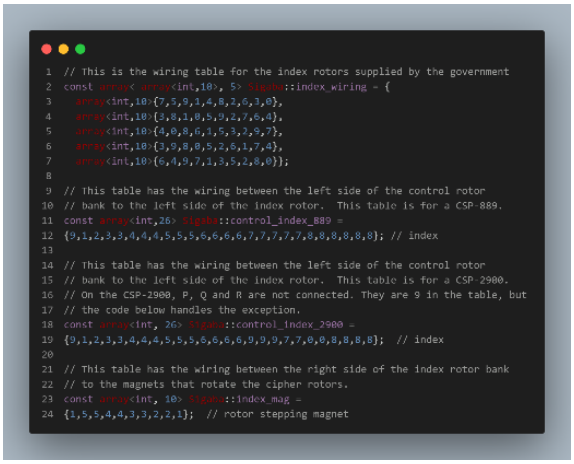
3.1 Program Mesin SIGABA pada Rotor

Pemerintah Amerika Serikat pada saat itu tidak menyediakan seluruh komponen rotor yang digunakan dalam teori sehingga implementasi mesin SIGABA pada saat itu tidak sekompleks mesin SIGABA yang berdasarkan teori. Namun, Implementasi pada program ini menggunakan komponen rotor yang lengkap sehingga akan membuat proses enkripsi-dekripsi jauh lebih kompleks. Program yang digunakan untuk mengimplementasikan atau menyimulasikan proses enkripsi dan dekripsi pada mesin SIGABA ini menggunakan bahasa pemrograman C++ dan sudah disesuaikan berdasarkan teori Kombinatorika dan teknik Kriptografi mesin SIGABA dengan kompleksitas yang maksimal, yakni $(26!)^5 * (26!)^5 * (10!)^5 \approx 2^{993}$.

NO.	*Wiring* pada Rotor *Cipher*	
	String	Hasil translasi dari bentuk *string* ke bentuk *integer*
1.	YCHLQSUGBDIXNZK ERPVJTAWFOM	24,2,7,11,16,18,20,6,1,3,8,23,13,25,10, 4,17,15,21,9,19,0,22,5,14,12
2.	INPXBWETGUYSAOC HVLDMQKZJFR	8,13,15,23,1,22,4,19,6,20,24,18,0,14,2, 7,21,11,3,12,16,10,25,9,5,17
3.	WNDRIOZPTAXHFJY QBMSVEKUCGL	22,13,3,17,8,14,25,15,19,0,23,7,5,9,24, 16,1,12,18,21,4,10,20,2,6,11
4.	TZGHOBKRVUXLQD MPNFWCJYEIAS	19,25,6,7,14,1,10,17,21,20,23,11,16,3, 12,15,13,5,22,2,9,24,4,8,0,18
5.	YWTAHRQJVLCEXU NGBIPZMSDFOK	24,22,19,0,7,17,16,9,21,11,2,4,23,20, 13,6,1,8,15,25,12,18,3,5,14,10
6.	QSLRBTEKOGAICFW YVMHJNXZUDP	16,18,11,17,1,19,4,10,14,6,0,8,2,5,22, 24,21,12,7,9,13,23,25,20,3,15
7.	CHJDQIGNBSAKVTU OXFWLEPRMZY	2,7,9,3,16,8,6,13,1,18,0,10,21,19,20,14 ,23,5,22,11,4,15,17,12,25,24
8.	CDFAJXTIMNBEQHS UGRYLWZKVPO	2,3,5,0,9,23,19,8,12,13,1,4,16,7,18,20, 6,17, 24,11,22,25,10,21,15,14
9.	XHFESZDNRBCGKQI JLTVMUOYAPW	23,7,5,4,18,25,3,13,17,1,2,6,10,16,8,9, 11,19,21,12,20,14,24,0,15,22
10.	EZJQXMOGYTCSFRI UPVNADLHWBK	4,25,9,16,23,12,14,6,24,19,2,18,5,17,8, 20,15,21,13,0,3,11,7,22,1,10

Tabel 3.1.1: *Wiring table* rotor *cipher* Mesin SIGABA

Gambar 3.1.1: Implementasi *wiring table* rotor *cipher* di C++
(Sumber: Dokumentasi Pribadi)

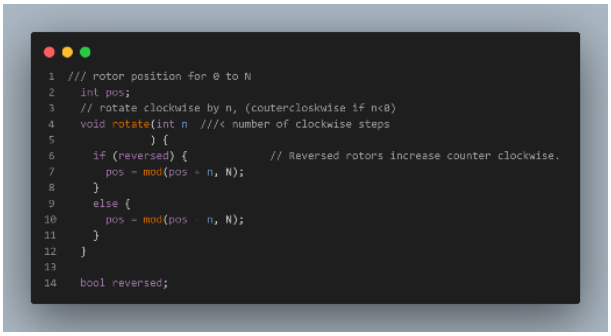


Gambar 3.1.2: Implementasi *wiring table* rotor indeks dan kontrol di C++
(Sumber: Dokumentasi Pribadi)

Berdasarkan **Tabel 3.1.1** dan **Gambar 3.1.1**, ditunjukkan bahwa terdapat translasi dari 10 rotor *cipher* yang awalnya merupakan sebuah *string*, lalu diubah menjadi *integer* serta diinisialisasi. Kemudian, ada inisialisasi kombinasi angka dari rotor indeks dan rotor kontrol ditunjukkan pada **Gambar 3.1.2**. Rotor kontrol bekerja mengatur *step* dari kiri ke kanan atau kanan ke kiri.

Kedua rotor, yakni *cipher* dan kontrol identik dalam penampilan dan fungsi sehingga dapat saling dipertukarkan. Oleh karena itu, terdapat sepuluh rotor yang tersedia untuk membentuk dua bank rotor, satu untuk masing-masing bank rotor *cipher* dan kontrol. Setiap rotor memiliki 26 kontak dengan huruf abjad A hingga Z dicetak di tepi luar. Dengan pengecualian penampilan huruf-huruf di sepanjang tepi luar pada kedua rotor *cipher* dan kontrol, bagian kiri identik dengan bagian kanan. Karena sifat ini, semua rotor *cipher* dan kontrol dapat dimasukkan secara terbalik dalam operasi *inverse*.

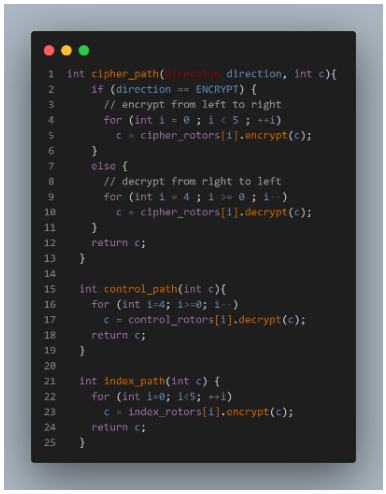
Rotor indeks memiliki desain yang mirip dengan rotor *cipher* dan kontrol, kecuali bahwa rotor indeks adalah rotor 10 kontak daripada 26. Setiap rotor indeks SIGABA mempermutasi bilangan bulat mulai dari nol hingga sembilan. Rotor indeks juga tidak melangkah seperti rotor *cipher* dan kontrol. Meskipun dapat ditempatkan secara terbalik pada mesin, pengaturan rotor indeks dalam orientasi terbalik tidak memengaruhi kriptanalisis SIGABA, sehingga fitur ini akan diabaikan.



```
/// rotor position for 0 to N
int pos;
// rotate clockwise by n, (coutercloskwise if n<0)
void rotate(int n  ///< number of clockwise steps
            ) {
  if (reversed) {             // Reversed rotors increase counter clockwise.
    pos = mod(pos + n, N);
  }
  else {
    pos = mod(pos - n, N);
  }
}

bool reversed;
```

Gambar 3.1.3: Implementasi rotasi rotor berlawanan/searah jarum jam di C++
(Sumber: Dokumentasi Pribadi)

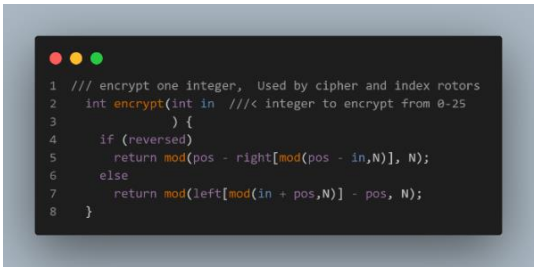


```
int cipher_path(Direction direction, int c){
    if (direction == ENCRYPT) {
      // encrypt from left to right
      for (int i = 0 ; i < 5 ; ++i)
        c = cipher_rotors[i].encrypt(c);
    }
    else {
      // decrypt from right to left
      for (int i = 4 ; i >= 0 ; i--)
        c = cipher_rotors[i].decrypt(c);
    }
    return c;
  }

  int control_path(int c){
    for (int i=4; i>=0; i--)
      c = control_rotors[i].decrypt(c);
    return c;
  }

  int index_path(int c) {
    for (int i=0; i<5; ++i)
      c = index_rotors[i].encrypt(c);
    return c;
  }
```

Gambar 3.1.4: Implementasi pengacakan kunci ketiga rotor di C++
(Sumber: Dokumentasi Pribadi)



```
/// encrypt one integer,  Used by cipher and index rotors
  int encrypt(int in  ///< integer to encrypt from 0-25
              ) {
    if (reversed)
      return mod(pos - right[mod(pos - in,N)], N);
    else
      return mod(left[mod(in + pos,N)] - pos, N);
  }
```

Gambar 3.1.5: Implementasi fungsi enkripsi di C++
(Sumber: Dokumentasi Pribadi)

```
/// decrypt an integer.  (used only for cipher and control wheels
  int decrypt(int in   ///< integer to decrypt
              ) {
    if (reversed) {
      return mod(pos - left[mod(pos - in, N)], N);
    }
    else {
      return mod(right[mod(in + pos, N)] - pos, N);
    }
  }
```

Gambar 3.1.6: Implementasi fungsi dekripsi di C++
(Sumber: Dokumentasi Pribadi)

Ketika SIGABA diatur ke mode dekripsi, semua rotor kecuali rotor *cipher* diatur dan berfungsi persis sama seperti dalam mode enkripsi. Ketika SIGABA berada dalam mode enkripsi, sinyal listrik melewati rotor *cipher* dari kiri ke kanan. Saat mendekripsi, sinyal dikirim dalam arah dari kanan ke kiri, yang setara dengan menggunakan permutasi *cipher* invers.

3.2 Hasil Implementasi Program Enkripsi-Dekripsi pada Mesin SIGABA

Dari program yang telah dibuat, penulis mengimplementasikan dan menjalankan program ini dengan Command Line Interface (CLI) pada Windows Subsystem for Linux (WSL). Hasil implementasi ini terbagi menjadi tiga uji coba karena mesin SIGABA memiliki tiga rotor, yakni rotor *cipher*, kontrol, dan indeks. Berdasarkan teori, rotor *cipher* dan rotor kontrol menerima 5 huruf dari huruf [A…Z], sedangkan rotor indeks menerima 5 angka dari *range* [0 ... 9].

Perbandingan Hasil Enkripsi-Dekripsi dari Tiap Jenis Rotor
Rotor *Cipher*

Enkripsi Mesin SIGABA:
`amalia@DESKTOP-910B8JR:/mnt/d/Main Files/File Amel/Kuliah/Akademik/Semester 3/MATDIS/sigaba/build/sigaba$ ./sigaba -e --cipherPos "AJKQM" --text "Destroy the ships"`
`DAXVF PGZDT CGIJR AH`
 	
Dekripsi Mesin SIGABA:
`amalia@DESKTOP-910B8JR:/mnt/d/Main Files/File Amel/Kuliah/Akademik/Semester 3/MATDIS/sigaba/build/sigaba$ ./sigaba -d --cipherPos "AJKQM" --text "DAXVF PGZDT CGIJR AH"`
`DESTROY THE SHIPS`

Rotor Kontrol

Enkripsi Mesin SIGABA:
`amalia@DESKTOP-910B8JR:/mnt/d/Main Files/File Amel/Kuliah/Akademik/Semester 3/MATDIS/sigaba/build/sigaba$ ./sigaba -e --controlPos "AJKQM" --text "Destroy the ships"`
`SAIQF QBVLW BQPGZ QU`
 	
Dekripsi Mesin SIGABA:
`amalia@DESKTOP-910B8JR:/mnt/d/Main Files/File Amel/Kuliah/Akademik/Semester 3/MATDIS/sigaba/build/sigaba$ ./sigaba -d --controlPos "AJKQM" --text "SAIQF QBVLW BQPGZ QU"`
`DESTROY THE SHIPS`

Rotor Indeks

Enkripsi Mesin SIGABA:
`amalia@DESKTOP-910B8JR:/mnt/d/Main Files/File Amel/Kuliah/Akademik/Semester 3/MATDIS/sigaba/build/sigaba$ ./sigaba -e --indexPos "02539" --text "Destroy the ships"`
`SSDPQ HLYUL EFDRZ OF`
 	
Dekripsi Mesin SIGABA:
`amalia@DESKTOP-910B8JR:/mnt/d/Main Files/File Amel/Kuliah/Akademik/Semester 3/MATDIS/sigaba/build/sigaba$ ./sigaba -d --indexPos "02539" --text "SSDPQ HLYUL EFDRZ OF"`
`DESTROY THE SHIPS`

Tabel 3.2.1: Implementasi program enkripsi-dekripsi Mesin SIGABA di CLI
(Sumber: Dokumentasi Pribadi)

Berdasarkan **Tabel 3.2.1**, dapat ditunjukkan bahwa proses enkripsi-dekripsi dari masing-masing rotor berbeda. Meskipun masukan pada rotor *cipher* dan kontrol sama, hasil pesan yang terenkripsi berbeda. Dengan kemungkinan yang sangat banyak dari proses permutasinya, mesin ini akan sangat sulit diretas kunci atau pola enkripsinya. Untuk memecahkan kode dengan $(26!)^5 * (26!)^5 * (10!)^5 \approx 2^{993}$ kemungkinan akan sangat sulit ditebak dan memerlukan teknologi yang lebih modern. Dengan kemungkinan pengacakan rotor yang sangat tinggi dan kompleks, pola tersebut mustahil dipecahkan pada masa Perang Dunia II. Hal ini membuktikan dan memperkuat penyebab dari pihak musuh bahkan pihak sekutu sekalipun tidak dapat mengetahui dan memecahkan kode mesin SIGABA milik Amerika Serikat. Oleh karena itu, kelebihan mesin SIGABA ini menjadi ‘senjata’ yang sangat kuat untuk menjaga kerahasiaan informasi dan komunikasi bagi Amerika Serikat.

3.3 Kelebihan dan Kekurangan Mesin SIGABA

Berdasarkan teori dan percobaan yang telah dilakukan, SIGABA dapat dinilai sebagai mesin *cipher* yang luar biasa. Dengan cara kerja pengacakan kombinasi karakter atau angka yang bersifat *pseudorandom*, ini menjadikan SIGABA sebagai salah satu mesin *cipher* terbaik karena memiliki kompleksitas

yang sangat tinggi dan aman. Namun, di satu sisi, ini menciptakan kelemahan pada SIGABA. Hal ini dikarenakan SIGABA yang memiliki 15 rotor sehingga memerlukan ruang yang lebih banyak dan menyebabkan bentuk fisik mesin SIGABA menjadi kurang praktis, serta sulit dioperasikan.

IV. Kesimpulan

Analisis terhadap implementasi teori Kombinatorika dan teknik Kriptografi yang digunakan pada mesin SIGABA dapat disimpulkan sebagai berikut.

4.1 Terdapat tiga jenis rotor, yakni rotor *cipher*, rotor kontrol, dan rotor indeks. Masing-masing jenis rotor tersebut memiliki rotor sejumlah lima buah sehingga memiliki total 15 rotor.

4.2 Rotor *cipher* dan rotor kontrol mengombinasikan karakter [A…Z], sedangkan rotor indeks mengombinasikan angka [0…9].

4.3 Pada masa Perang Dunia II, nilai kombinasi permutasi mesin SIGABA menghampiri $2^{48,4}$. Namun secara teori, kemampuan kombinasi permutasi mesin SIGABA dapat mencapai 2^{993}.

V. Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa, atas rahmat dan karunia-Nya, penulis dapat menyelesaikan makalah ini. Tak lupa, penulis mengucapkan terima kasih kepada kedua orang tua penulis yang selalu memberikan dukungan dan doa. Penulis juga mengucapkan terima kasih kepada Ibu Dr. Nur Ulfa Maulidevi, S.T, M.Sc., Ibu Dr. Fariska Zakhralativa Ruskanda, S.T., M.T., dan Bapak Dr. Ir. Rinaldi, M.T yang telah membimbing kami pada mata kuliah IF2120 Matematika Diskrit. Terima kasih kepada semua pihak yang turut mendukung penulis dalam menyelesaikan makalah ini.

Referensi

[1] Crypto Museum. 2009. “SIGABA Electric cipher machine (ECM) Mark II” https://www.cryptomuseum.com/crypto/usa/sigaba/index.htm, diakses pada 23 Oktober 2023)

[2] Chan, Wing On. 2007. “Cryptanalysis of SIGABA”. https://core.ac.uk/download/pdf/70407877.pdf (diakses pada 25 Oktober 2023)

[3] Stamp, Mark & Wing On Chan. 2007. “SIGABA: Cryptanalysis of the full keyspace”. https://www.researchgate.net/publication/220615771_SIGABA_Cryptanalysis_of_the_full_keyspace (diakses pada 25 Oktober 2023)

[4] Munir, Rinaldi. 2023. “Kombinatorial (Bagian 1)”. https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/17-Kombinatorial-Bagian1-2023.pdf (diakses pada 3 Desember 2023).

[5] Uly, Risma. 2019. “Buku Probabilitas”. http://repository.uki.ac.id/1304/1/Buku%20Probalitas.pdf (diakses pada 3 Desember 2023)

[6] Amin, M. Miftakul. 2016. “Implementasi Kriptografi Klasik pada Komunikasi Berbasis Teks”. https://media.neliti.com/media/publications/127196-ID-implementasi-kriptografi-klasik-pada-kom.pdf (diakses pada 3 Desember 2023).

[7] Munir, Rinaldi. 2023. “Teori Bilangan (Bagian 3)”. https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/16-Teori-Bilangan-Bagian3-2023.pdf (diakses pada 3 Desember 2023).

[8] Munir, Rinaldi. 2023. “Teori Bilangan (Bagian 1)”. https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/14-Teori-Bilangan-Bagian1-2023.pdf (diakses pada 3 Desember 2023).

[9] Dunn, Joseph. 2019. “SIGABA”. https://github.com/JoeDunnStable/sigaba.git (diakses pada 23 Oktober 2023)

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2023

Amalia Putri 13522042