Penerapan Kombinatorika dan Teori Graf pada Mesin Bombe untuk Pemecahan Enigma

Mikhael Andrian Yonatan - 13524051
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: mikhael.yonatan10@gmail.com , 13524051@std.stei.itb.ac.id

Abstrak—Kriptografi telah menjadi hal yang begitu esensial dalam kehidupan manusia, baik di masa kini maupun masa lalu. Pada masa Perang Dunia II, terdapat dua mesin kriptografi yang begitu luar biasa, yaitu Enigma dan Bombe yang keduanya sangat berkaitan erat dengan teori graf dan kombinatorika. Penelitian ini bertujuan untuk menganalisis konsep graf yang dapat diterapkan pada Mesin Bombe untuk memahami cara kerjanya dalam memecahkan kode yang dihasilkan Mesin Enigma. Pada penelitian ini metode yang digunakan adalah uji coba secara langsung terhadap suatu algoritma. Hasil dari penelitian ini membuktikan bahwa konsep graf berlaku dan menjadi dasar dari cara kerja Mesin Bombe, terlepas dari ruang kemungkinan yang sangat besar dan mekanismenya yang begitu kompleks dari segi mekanis.

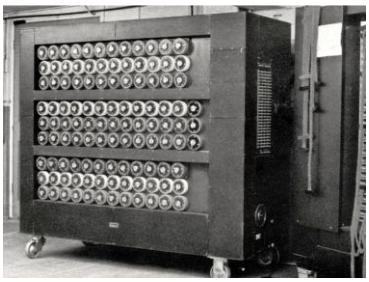
Kata Kunci—Graf, kombinatorika, Kriptografi, Mesin Enigma, Mesin Bombe, uji coba algoritma.

I. PENDAHULUAN

Pada masa Perang Dunia II, kriptografi merupakan hal yang dapat menentukan arahnya peperangan. Informasi penting seperti strategi militer, pergerakan pasukan, dan komunikasi antarkomando harus diamankan dengan metode penyandian yang kompleks sehingga tidak mudah dipecahkan dan diketahui oleh pihak musuh. Mesin Enigma adalah mesin penyandi yang sangat terkenal dan kompleks pada masa itu, yang digunakan oleh Jerman untuk mengenkripsi dan mendekripsi pesan-pesan militernya. Faktor utama yang membuat Mesin Enigma begitu kompleks adalah kemungkinan konfigurasi awal yang digunakan.

Gambar 1. Mesin Enigma, diambil dari [1]

Hal ini menciptakan kebuntuan bagi pihak Sekutu karena tidak dapat memahami begitu banyak pesan yang telah berhasil dicegat. Oleh karena itu, pihak Sekutu (terutama Inggris) membentuk tim kriptanalis yang dipimpin oleh Alan Turing di Bletchley Park untuk memecahkan kode yang dihasilkan Mesin Enigma. Maka, diciptakanlah sebuah mesin elektromekanik bernama Turing Welchman Bombe (Mesin Bombe), yang dirancang untuk mempercepat proses analisis kemungkinan konfigurasi awal Mesin Enigma dan memungkinkan Sekutu membaca pesan-pesan yang sebelumnya tidak terpecahkan.



Gambar 2. Mesin Bombe, diambil dari [2]

Mesin Enigma dan Mesin Bombe keduanya memiliki keterkaitan erat, bukan hanya secara historis, tetapi juga secara mekanis dan matematis. Kedua mesin dapat direpresentasikan dalam bentuk graf yang menggambarkan alur sinyal saat transformasi huruf pada setiap kemungkinan konfigurasi, Representasi ini membuka peluang untuk menganalisis dan memahami cara kerja keduanya secara lebih sistematis dan visual.

Meskipun telah banyak penelitian yang membahas bagaimana Mesin Enigma dan Mesin Bombe bekerja, sebagian besar penelitian tersebut berfokus pada simulasi kerja dan implementasi algoritmik, namun masih kurang menekankan pemanfaatan teori graf. Penelitian ini bertujuan untuk menekankan pendekatan teori graf dalam analisis Mesin Bombe, untuk memahami bagaimana jalur sinyal dapat dimodelkan sebagai simpul dan sisi dalam suatu graf sebagai proses dalam menemukan konfigurasi Mesin Enigma yang tepat. Dengan demikian, diharapkan pemahaman terhadap mekanisme pemecahan kode melalui Mesin Bombe dapat dilakukan secara konseptual dan visual melalui pendekatan graf.

II. LANDASAN TEORI

A. Kombinatorika

Kombinatorika adalah cabang matematika yang menghitung jumlah penyusunan objek-objek tanpa harus mengenumerasi semua kemungkinan susunannya. Pada kombinatorika terdapat dua cara berbeda untuk menghitung jumlah kemungkinan susunan dari suatu himpunan, yaitu permutasi dan kombinasi.

Permutasi adalah banyaknya cara menyusun himpunan objek unik (setiap objek berbeda) dengan memerhatikan urutannya. Jika terdapat n objek unik dan ingin disusun sebanyak r dengan r ≤ n, maka jumlah kemungkinan permutasi dapat dihitung dengan:

$$P(n, r) = \frac{n!}{(n-r)!}$$

Kombinasi adalah banyaknya cara menyusun himpunan objek tanpa memerhatikan urutannya (setiap objek dianggap sama). Jika terdapat n dan ingin disusun sebanyak r dengan r ≤ n, maka jumlah kemungkinan kombinasi dapat dihitung dengan:

$$C(n, r) = \frac{n!}{r!\,(n-r)!}$$

Dari dua cara perhitungan sebelumnya dapat diperluas untuk berbagai macam kasus, salah satunya adalah apabila terdapat n objek unik yang ingin disusun menjadi x pasangan maka jumlah kemungkinan susunannya dapat dihitung dengan:

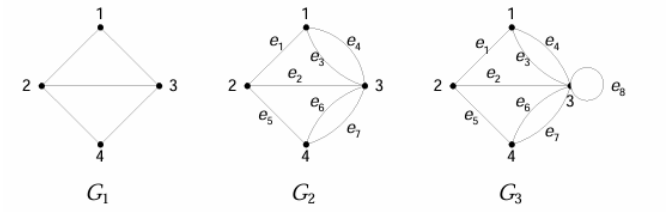
$$K = \frac{P(n,n)}{(n-2x)!\,x!\,2^x}$$

$$K = \frac{n!}{(n-2x)!\,x!\,2^x}$$

B. Graf

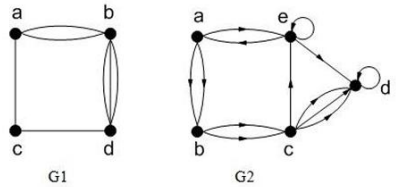
Graf adalah salah satu metode yang dapat digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Graf G didefinisikan sebagai G = (V, E), dengan V adalah himpunan tidak kosong dari simpul-simpul dan E adalah himpunan dari sisi yang menghubungkan sepasang simpul. Terdapat dua kriteria dasar dalam mengklasifikasikan graf, yaitu berdasarkan ada tidaknya gelang dan orientasi arah pada sisi.

Berdasarkan ada tidaknya gelang, graf digolongkan menjadi dua macam yaitu graf sederhana dan graf tak-sederhana. Graf sederhana artinya pada graf tidak terdapat sisi gelang maupun sisi ganda. Graf tak-sederhana dibagi lagi menjadi dua, apabila mengandung sisi gelang maka disebut graf semu, apabila hanya mengandung sisi ganda maka disebut graf ganda.



Gambar 3. G1 graf sederhana, G2 graf ganda, G3 graf semu, diambil dari [8]

Berdasarkan orientasi arah pada sisi, graf digolongkan menjadi dua macam yaitu graf berarah dan graf tak-berarah. Graf berarah artinya setiap sisi memiliki orientasi arah. Graf tak-berarah artinya setiap sisi tidak memiliki orientasi arah.

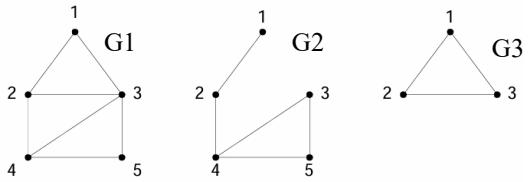


Gambar 4. G1 graf tak-berarah, G2 graf berarah, diambil dari [5]

Dalam teori graf terdapat banyak terminologi, dalam penelitian ini perlu memahami beberapa di antaranya yaitu derajat, upagraf, lintasan, dan sirkuit.

Derajat adalah nilai yang dimiliki oleh setiap simpul, nilai yang dimaksud adalah jumlah sisi yang berhubungan dengan simpul tersebut. Secara praktis, derajat suatu simpul dapat diketahui dengan jumlah simpul yang berhubungan (oleh sisi) dengan simpul yang bersangkutan.

Dari suatu graf G = (V, E) dapat dibuat suatu upagraf (subgraf) G1 = (V_1, E_1) dengan $V_1 \subseteq V$ dan $E_1 \subseteq E$. Apabila upagraf G1 mengandung seluruh simpul pada graf G, maka G1 disebut sebagai upagraf merentang.

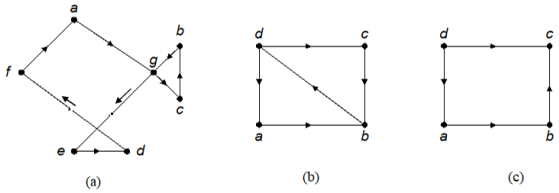


Gambar 6. G1 graf , G2 upagraf merentang dari G1, G3 upagraf dari G1 diambil dari [8]

Lintasan adalah barisan selang-seling simpul dan sisi pada suatu graf. Misalnya, lintasan dengan panjang n dari simpul v_0 ke simpul v_n dalam graf G adalah v_0, e_1, v_1, e_2, v_2, ..., e_n, v_n sehingga diperoleh $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2, ..., e_n = (v_{n-1}, v_n)$. Sirkuit adalah lintasan tertutup yang dimulai dari simpul v_n dan berakhir kembali di simpul v_n. Maka, hal utama yang membedakan lintasan dengan sirkuit adalah simpul awal dan simpul akhir. Pada sirkuit, simpul awal dan akhir haruslah sama sehingga terbentuk lintasan tertutup. Pada sirkuit maupun lintasan tidak boleh terdapat pengulangan sisi atau simpul, kecuali simpul awal dan akhir pada sirkuit (lintasan tertutup).

Pada teori graf, terdapat dua jenis lintasan dan sirkuit yaitu Euler dan Hamilton.

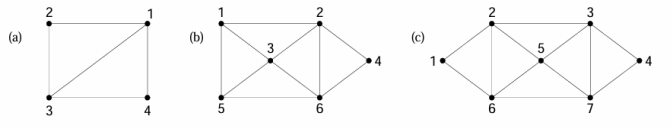
Lintasan Euler adalah lintasan yang melalui masing-masing sisi pada graf tepat satu kali, graf yang memiliki lintasan Euler disebut graf semi-Euler. Sirkuit Euler adalah sirkuit yang melalui masing-masing sisi pada graf tepat satu kali, graf yang memiliki sirkuit Euler disebut graf Euler. Suatu graf berarah memiliki lintasan Euler jika dan hanya jika graf terhubung dan setiap simpul memiliki derajat-masuk dan derajat-keluar sama keculai dua simpul, yang pertama memiliki derajat-keluar satu lebih besar derajat-masuk, dan yang kedua memiliki derajat-masuk satu lebih besar dari derajat keluar. Suatu graf berarah memiliki sirkuit Euler jikan dan hanya jika graf terhubung dan setiap simpul memiliki derajat-masuk dan derajat-keluar yang sama.



Gambar 7. a Graf berarah Euler, b Graf berarah semi-Euler, c tidak keduanya, diambil dari [9]

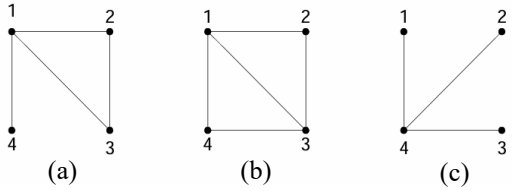
Suatu graf tak-berarah memiliki lintasan Euler jika dan hanya jika graf terhubung dan memiliki dua buah simpul berderajat ganjil atau semua simpul berderajat genap. Suatu graf

tak-berarah memiliki sirkuit Euler jika dan hanya jika graf terhubung dan setiap simpul berderajat genap.



Gambar 8. a Graf tak-berarah semi-Euler, b Graf tak-berarah semi-Euler, c Graf tak-berarah Eulder, diambil dari [9]

Lintasan Hamilton adalah lintasan yang melalui setiap simpul di dalam graf tepat satu kali, graf yang memiliki lintasan Hamilton disebut graf semi-Hamilton. Sirkuit Hamilton adalah sirkuit yang melalui setiap simpul di dalam graf tepat satu kali, kecuali simpul asal (sekaligus simpul akhir) yang akan dilalui dua kali, graf yang memiliki sirkuit Hamilton disebut graf Hamilton. Cara paling efektif dalam menentukan apakah suatu graf memiliki lintasan Hamilton atau sirkuit Hamilton adalah secara empiris.

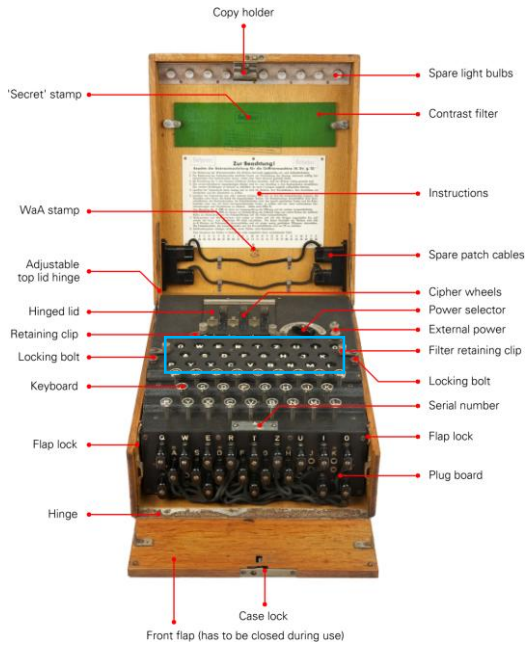


Gambar 9. a Graf tak-berarah semi-Hamilton, b Graf tak-berarah Hamilton, c tidak keduanya, diambil dari [9]

III. PEMBAHASAN

A. Mesin Enigma

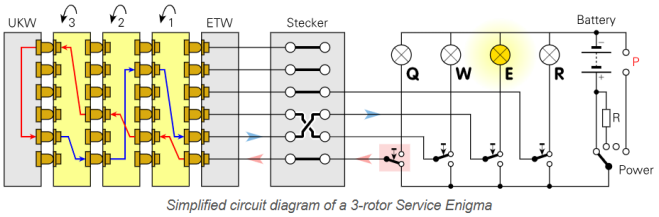
Sebelum memahami cara kerja dari Mesin Bombe, kita perlu memahami Mesin Enigma dengan baik terlebih dahulu. Nyatanya, terdapat lebih dari satu model Mesin Enigma, namun yang akan dibahas dalam penelitian ini adalah model yang paling banyak digunakan oleh militer Jerman semasa perang yaitu Enigma I. Perbedaan utama pada model ini dibandingkan model lainnya adalah hanya menggunakan 3 buah rotor saja.



Gambar 10. Enigma I / Wehrmacht Enigma, diambil dari [1]

Di balik arsitektur Mesin Enigma yang cukup kompleks (Gambar 10) terdapat cara kerja yang sederhana. Secara umum, Mesin Enigma bekerja seperti rangkaian listrik dengan *keyboard* sebagai saklarnya yang apabila *keyboard* ditekan maka saklar akan menghubungkan rangkaian sehingga salah satu lampu (kotak biru pada Gambar 10) akan teraliri listrik dan menyala. Lampu yang menyala akan menerangi suatu huruf, huruf tersebutlah yang menjadi hasil enkripsi/dekripsi dari huruf yang ditekan pada keyboard.

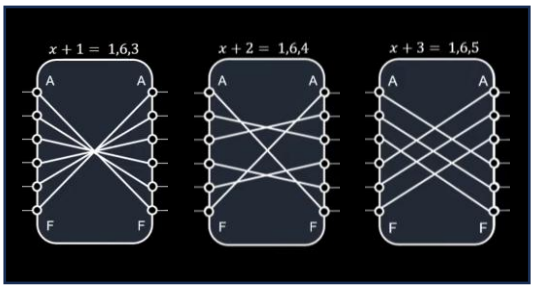
Proses transformasi suatu huruf pada Mesin Enigma terjadi sebanyak sembilan kali, dengan alur sebagai berikut *keyboard (input) – plugboard (stecker) – rotor I – rotor II – rotor III – reflector (UKW) – rotor III – rotor II – rotor I – plugboard (stecker – bulb (output)*.



Gambar 11. Proses transformasi huruf pada Enigma I, diambil dari [1]

Untuk menambah kekompleksannya, setiap kali *keyboard* ditekan, maka rotor paling kanan akan berputar satu kali dan apabila telah berputar 26 kali (jumlah huruf). Setelah rotor paling kanan berputar 26 kali, maka rotor tengah akan berputar satu kali. Setelah rotor tengah berputar 26 kali, maka rotor paling kiri akan berputar satu kali. Maka dapat disimpulkan, bahwa sangat mustahil untuk menelusuri setiap transformasinya huruf per huruf di seluruh pesan!

Kekompleksan transformasi huruf pada mekanisme rotor yang berubah-ubah dapat direpresentasikan dalam graf sehingga lebih sederhana. Simpul merepresentasikan huruf yang ada, sisi merepresentasikan hasil. Misalkan, saya memiliki simulasi 6 huruf saja dengan konfigurasi awal posisi rotor 1, 6, 2. Apabila *keyboard* ditekan maka konfigurasi rotor akan menjadi 1, 6, 3, kemudian 1, 6, 4, dan seterusnya.



Gambar 12. Representasi graf transformasi huruf, diambil dari [5]

Pada Gambar 12, akibat dari perputaran rotor apabila memasukkan ‘B’ pada iterasi pertama akan menjadi ‘E’, namun pada iterasi kedua akan menjadi ‘C’, dan pada iterasi ketiga akan menjadi ‘E’, perubahan tersebut bersifat acak.

Akan tetapi, ternyata Enigma dapat digunakan untuk mengenkripsi dan mendekripsi pesan karena hasil huruf enkripsi apabila dimasukkan kembali ke Enigma akan didekripsi menjadi huruf awal, hanya saja konfigurasi awal yang digunakan haruslah sama. Misalnya, input ‘B’ menjadi ‘E’, kemudian *reset* pengaturan ke konfigurasi awal, input ‘E’ akan menjadi ‘B’.

Oleh karena itu, apabila kita memiliki suatu kode Enigma dan konfigurasi awal Mesin Enigma ketika kode tersebut dienkripsi, kemudian kita menggunakan konfigurasi awal tersebut dan memasukkan kode Enigma yang dimiliki akan dihasilkan pesan sebenarnya (bersifat *reversible*).

Konfigurasi awal Mesin Enigma terdiri dari lima hal yaitu *walzenlage* (pemilihan jenis dan urutan rotor), *ringstellung* (*offset* setiap rotor terhadap acuan), *grundstellung* (posisi awal setiap rotor), *steckerbrett* (pasangan huruf pada *plugboard*), *umkehrwalze* (jenis *reflector*).

Terdapat lima buah jenis rotor yang dapat dipilih dan dari lima rotor tersebut akan digunakan tiga buah saja dengan memerhatikan urutannya, maka diperoleh jumlah kemungkinan pilihan sebanyak P(5,3) = 60 kemungkinan.

Untuk setiap rotor mengandung 26 huruf sehingga memiliki 26 kemungkinan pergeseran *offset* terhadap acuan, begitu juga untuk posisi awal terdapat 26 kemungkinan, maka untuk setiap terdapat 26^2 kemungkinan. Akibat pada satu Mesin Enigma digunakan tiga buah rotor yang masing-masingnya memiliki 26^2 kemungkinan, maka untuk 3 buah rotor terdapat 26^6 kemungkinan.

Pemasangan huruf-huruf pada *plugboard* selalu dan hanya dapat dilakukan untuk 10 pasangan (20 dari 26 huruf) yang memiliki jumlah kemungkinan sebanyak K dengan perhitungan sebagai berikut,

$$K = \frac{P(26,26)}{(26-2.10)!\ 10!\ 2^{10}}$$

Terdapat dua buah jenis reflektor yang dapat dipilih dan hanya akan digunakan satu saja, maka jumlah kemungkinan pilihan sebanyak 2 kemungkinan saja.

Dengan menggabungkan seluruh kemungkinan dari setiap hal yang terdapat pada konfigurasi Mesin Enigma maka diperoleh kemungkinan konfigurasi awal yang dapat dilakukan sebagai berikut,

$2 \cdot 60 \cdot 26^6 \cdot K = 5.58785 \times 10^{24}$ kemungkinan

Berdasarkan perhitungan yang telah dilakukan, nyatanya mengetahui konfigurasi awal dari Mesin Enigma sama sulitnya dengan menelusuri transformasi huruf per huruf karena terdapat septiliun kemungkinan. Terlebih lagi, pihak Jerman selalu mengganti konfigurasi awal Mesin Enigma setiap tengah malam sehingga untuk setiap kode Enigma hanya memiliki waktu tepat satu hari untuk dipecahkan.

Akan tetapi, mengacu pada arsitektur dari Mesin Enigma, mesin ini memiliki dua kelemahan utama yaitu huruf yang sama tidak mungkin dienkripsi/didekripsi menjadi dirinya sendiri (akibat keterbatasan arsitektur) dan apabila konfigurasi yang digunakan sama akan diperoleh sifat *reversible* antara dua huruf yang saling terhubung pada graf. Akibat kelemahan tersebut, ruang kemungkinan menjadi lebih kecil dan menjadi dasar dari cara kerja Mesin Bombe.

B. Mesin Bombe

Mesin Bombe memiliki arsitektur dan cara kerja mekanis yang jauh lebih rumit dibandingkan Mesin Enigma (arsitektur secara rinci tidak akan dibahas pada penelitian ini). Namun, secara umum Mesin Bombe terdiri dari *diagonal board* dan banyak *drum*, setiap *drum* merepresentasikan rotor yang terdapat pada Mesin Enigma sehingga dapat dikatakan bahwa Mesin Bombe seperti gabungan dari banyak Mesin Enigma. Telah disebutkan sebelumnya bahwa Mesin Bombe dapat memecahkan kode Enigma, namun hal tersebut tidak sepenuhnya tepat karena memecahkan kode Enigma secara langsung adalah hal yang mustahil. Kenyataannya, Mesin Bombe hanya mencari dua hal pada konfigurasi awal Mesin Enigma yaitu *steckerbrett* (pasangan huruf pada *plugboard*) dan *grundstellung* (posisi awal setiap rotor) untuk diterapkan pada Mesin Enigma sehingga kode Enigma yang dimiliki dapat didekripsi menjadi pesan asli.

Sebelum menggunakan Mesin Bombe harus ditemukan celah pada kode Enigma terlebih dahulu, celah yang dimaksud berupa dugaan yang berkaitan dengan kelemahan dari Mesin Enigma. Pada sebagian besar pesan yang dikirimkan oleh pihak Jerman, ternyata ada satu kata yang selalu terkandung di dalamnya yaitu "WEATHER". Memanfaatkan kelemahan Mesin Enigma yang tidak dapat mengubah suatu huruf menjadi dirinya sendiri, maka dapat dilakukan pencocokkan sebagai berikut,

Gambar 13. Pencarian celah pada kode Enigma, diambil dari [5]

Dari hasil pada Gambar 13, dapat diperoleh dugaan bahwa pada posisi rotor x+1 'W' menjadi 'O', x+2 'E' menjadi 'T', x+3 'A' menjadi 'H' dan seterusnya. Tugas Mesin Bombe adalah mencari nilai x tersebut dan pengaturan dari *plugboard* secara iteratif.

Telah disebutkan sebelumnya, bahwa Mesin Bombe seperti gabungan dari banyak Mesin Enigma, maka mula-mula Mesin Bombe akan melakukan pemetaan huruf A-Z apabila dimasukkan ke Mesin Enigma (seperti Gambar 12) pada setiap posisi rotor. Misalnya, pada posisi awal rotor 1-1-1, dilakukan pemetaan terhadap huruf A hingga Z untuk melihat huruf keluaran yang mungkin terbentuk. Langkah ini kemudian diulangi untuk seluruh kombinasi posisi rotor, seperti 1-1-2, 1-1-3, dan seterusnya, sehingga seluruh konfigurasi dapat diuji. Hal ini dilakukan karena huruf yang masuk dapat ditukar menjadi huruf apapun oleh *plugboard*.

Selanjutnya, dilakukan penyederhanaan rangkaian sirkuit rotor terhadap huruf-huruf yang berkorespondensi. Setelahnya dilakukan "pengaliran listrik" pada salah satu huruf, apabila pada rangkaian terdapat graf Hamilton dan graf Euler maka solusi adalah salah dan apabila pada rangkaian terdapat sirkuit Hamilton yang hanya melibatkan 6 simpul maka solusi adalah benar.

IV. IMPLEMENTASI PROGRAM

Contoh pada bagian sebelumnya terlalu kompleks untuk dilakukan uji coba, maka dari itu saya mencoba untuk membuat simulasi pada "dunia 6 huruf". Enigma hanya mengandung 6 huruf yaitu A-F. Mula-mula, perlu diketahui pesan asli dan hasil enkripsinya sebagai celah, di sini diperlukan konfigurasi awal dari Mesin Enigma dan konfigurasi awal ini juga lah yang akan Mesin Bombe coba cari. Berikut adalah konfigurasi awal yang digunakan,

```
ROTORS = {
    'I':   'BADCFE',
    'II':  'CBAFED',
    'III': 'DECFAB',
    'IV':  'ECFDBA',
    'V':   'FABCDE' }

REFLECTORS = {
    'B': {'A': 'F', 'F': 'A', 'B': 'E', 'E': 'B', 'C': 'D', 'D': 'C'},
    'C': {'A': 'B', 'B': 'A', 'C': 'F', 'F': 'C', 'D': 'E', 'E': 'D'} }

rotors = ['I', 'II', 'III']
start_positions = [1, 1, 3]
reflector = REFLECTORS['C']
plugboard = [('A', 'B'), ('C', 'E'), ('F', 'D')]
```

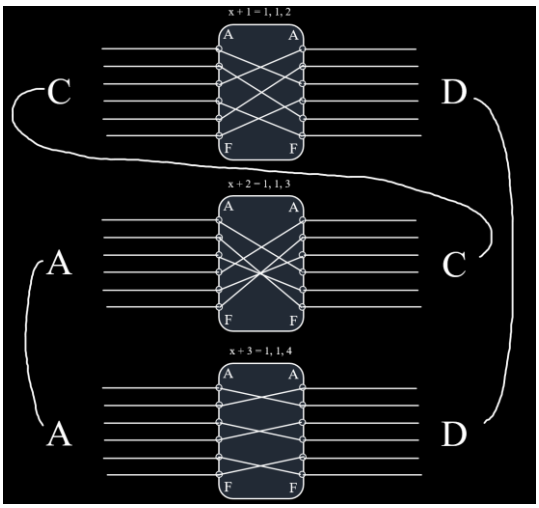
Gambar 14. Konfigurasi awal Enigma, diambil dari [10]

Pada konfigurasi awal di atas tidak menyertakan konfigurasi *ringstellung* karena dianggap tidak ada pergeseran *offset*. Rotor yang digunakan dengan urutan I, II, III dengan posisi awal masing-masing rotor adalah 1, 1, 3. *Reflector* yang digunakan adalah jenis C dan karena ada 6 huruf maka terdapat 3 pasang perubahan pada *plogboard*. Sekarang, misalkan saya mengetahui bahwa di setiap pesan Enigma mengandung pesan asli "CAA", saya juga memiliki pesan enigma "CDCDE", maka berdasarkan kelemahan Mesin Enigma dapat diketahui bahwa "CAA" berkorespondensi dengan "DCD". 'C' akan menjadi 'D' pada x+1, 'A' akan menjadi 'C' pada x+2, dan 'A' akan menjadi 'D' pada x+3. Sekarang kita coba memasukkan "CAA" kepada program simulai Enigma dengan konfigurasi awal yang telah kita buat sebelumnya yang menghasilkan luaran berikut

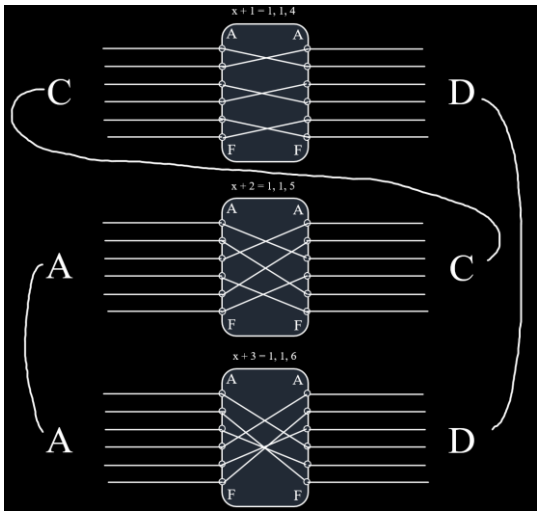
```
Encrypting: C
Rotor positions: [1, 1, 4]
Plugboard in: A
Rotor III shifted wiring: FABDEC
After rotor III forward: F
After rotor II forward: D
After rotor I forward: C
After reflector: C -> F
BADCFE
After rotor I backward: E
CBAFED
After rotor II backward: E
FABDEC
After rotor III backward: E
Plugboard out: D

Encrypting: A
Rotor positions: [1, 1, 5]
Plugboard in: C
Rotor III shifted wiring: ABDECF
After rotor III forward: D
After rotor II forward: F
After rotor I forward: E
After reflector: E -> D
BADCFE
After rotor I backward: C
CBAFED
After rotor II backward: A
ABDECF
After rotor III backward: A
Plugboard out: C

Encrypting: A
Rotor positions: [1, 1, 6]
Plugboard in: C
Rotor III shifted wiring: BDECFA
After rotor III forward: E
After rotor II forward: E
After rotor I forward: F
After reflector: F -> C
BADCFE
After rotor I backward: D
CBAFED
After rotor II backward: F
BDECFA
After rotor III backward: E
Plugboard out: D

Final Ciphertext: DCD
```

Gambar 15. Luaran dari pesan asli "CAA", diambil dari [10]

Hasilnya sesuai, itu berarti x = 3. Jawaban x dapat diketahui dengan mudah karena saya memang membuat "CAA" pada konfigurasi tersebut. Namun, pada kenyataannya, konfigurasi awal tidaklah seharusnya diketahui dan akan dicari oleh Mesin Bombe. Mula-mula dilakukan pemetaan terlebih dahulu untuk setiap posisi rotor, sambungkan juga huruf yang berkorespondensi.

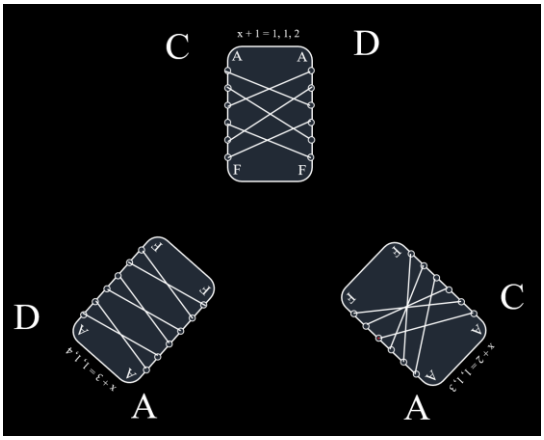


Gambar 16. Transformasi huruf dengan posisi rotor awal 1, 1, 1, diambil dari [10]

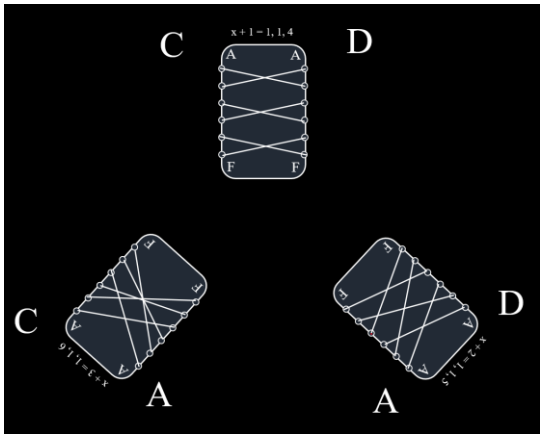


Gambar 17. Transformasi huruf dengan posisi rotor awal 1, 1, 3, diambil dari [10]

Selanjutnya sederhanakan rangkaian

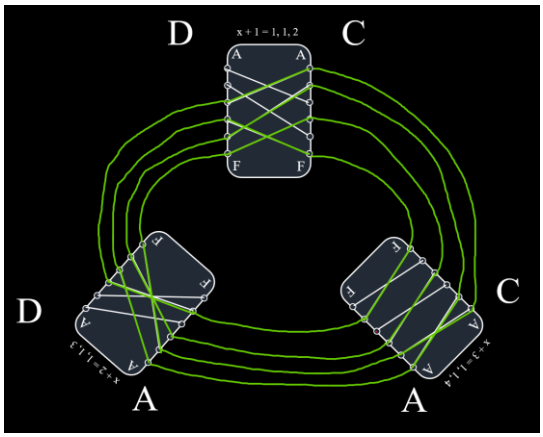


Gambar 18. Penyederhanaan Gambar 16, diambil dari [10]

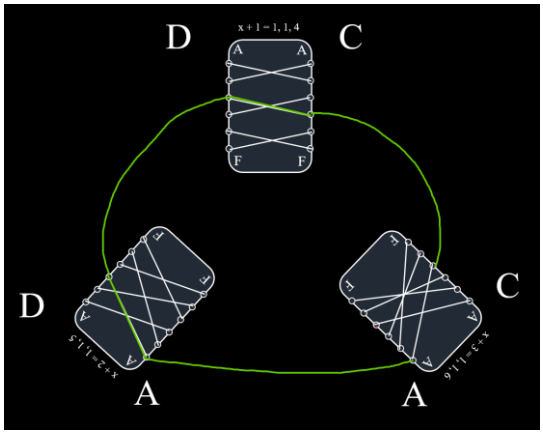


Gambar 19. Penyederhanaan Gambar 17, diambil dari [10]

Kemudian gambarkan sirkuit yang terdapat di dalamnya



Gambar 20. Sirkuit pada Gambar 18, diambil dari [10]



Gambar 21. Sirkuit pada Gambar 19, diambil dari [10]

Pada percobaan di atas, dapat dibuktikan bahwa solusi yang tepat hanya melibatkan tepat enam simpul. Namun, untuk solusi yang salah melanggar teori awal, di mana seharusnya memiliki graf Hamilton dan graf Euler (melibatkan seluruh simpul dan sisi), pada percobaan di atas sirkuit Hamilton dan sirkuit Euler dimiliki oleh upagraf (tidak melibatkan seluruh simpul dan sisi). Berdasarkan hasil di atas, hanya diperoleh satu konfigurasi *plugboard* karena hanya beberapa huruf tertentu yang terlibat.

Penyebab perbedaan dari hasil teori dan percobaan, kemungkinan disebabkan oleh pilihan konfigurasi awal dari Mesin Enigma yang tidak sesuai dengan kenyataannya, selain itu Mesin Enigma juga memiliki keterbatasan mekanis yang tidak dimiliki oleh perangkat *digital*.

V. KESIMPULAN

Penelitian ini menunjukkan bahwa proses pemecahan kode pada Mesin Enigma oleh Mesin Bombe bukan semata-mata hasil dari percobaan mekanis, melainkan didasari oleh prinsip-prinsip matematika, khususnya teori graf dan kombinatorika. Dengan memodelkan kemungkinan hubungan antarhuruf melalui graf, Mesin Bombe mampu menelusuri lintasan-lintasan kemungkinan konfigurasi awal yang sesuai dengan *crib* yang diketahui, sehingga proses pencocokan menjadi jauh lebih efisien meskipun ruang kemungkinan sangat besar.

Penerapan konsep pasangan pada plugboard dan pergeseran rotor dalam konfigurasi awal Enigma juga memperkuat pentingnya pendekatan kombinatorika dalam analisis ini. Oleh karena itu, pemahaman terhadap teori graf tidak hanya berguna dalam konteks abstrak matematika, tetapi juga memiliki kontribusi nyata dalam sejarah dunia, seperti pada keberhasilan Sekutu memecahkan sistem kriptografi Jerman dalam Perang Dunia II.

Dengan demikian, makalah ini menegaskan bahwa pendekatan matematis, khususnya teori graf dan kombinatorika, sangat relevan dan efektif dalam memahami serta menjelaskan cara kerja Mesin Bombe dalam mendekripsi pesan-pesan rahasia Enigma.

VI. UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan makalah ini dengan baik. Saya juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah memberikan dukungan, bantuan, dan motivasi selama proses penulisan makalah ini. Ucapan terima kasih saya sampaikan kepada:

1. Dr. Ir. Rinaldi, M.T. selaku dosen pengajar yang telah memberikan arahan dan referensi dalam mengerjakan makalah ini.
2. Orang tua saya yang selalu memberikan doa, semangat, dan dukungan moral.
3. Semua pihak yang tidak bisa saya sebutkan satu per satu, yang telah membantu baik secara langsung maupun tidak langsung.

Semoga makalah ini dapat memberikan manfaat dan kontribusi yang positif bagi pembaca. Saya menyadari bahwa makalah ini masih jauh dari sempurna, oleh karena itu saran dan kritik yang membangun sangat saya harapkan.

VII. LAMPIRAN

Kode program secara lengkap dapat dilihat pada link github berikut: https://github.com/MikhaelYonatan/Makalah-MATDIS.git

REFERENSI

[1] Cryptomuseum. (n.d.). *Enigma I*. 18 Juni 2025, dari https://www.cryptomuseum.com/crypto/enigma/i/

[2] Humanists UK. (n.d.). *Bombe Machine*. 18 Juni 2025, dari https://heritage.humanists.uk/object/bombe-machine/

[3] Munir, Rinaldi. (2024). *Kombinatorika (Bagian 1)*. 18 Juni 2025, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/18-Kombinatorika-Bagian1-2024.pdf

[4] YouTube. (2023). *The Enigma Machine: Encrypt and Decrypt Messages*. 18 Juni 2025, dari https://youtu.be/zCn3GCOwmeI?si=mkisKF6nTEjVcpxw

[5] YouTube. (2023). *How Did the Bombe Machine Work?* 18 Juni 2025, dari https://youtu.be/ybkkiGtJmkM?si=E4C0kUi5e9WtrfO8

[6] Brilliant.org. (n.d.). *Enigma Machine*. Diakses pada 19 Juni 2025, dari https://brilliant.org/wiki/enigma-machine/

[7] Wikipedia. (n.d.). *Enigma machine*. Diakses pada 19 Juni 2025, dari https://en.wikipedia.org/wiki/Enigma_machine

[8] Munir, Rinaldi. (2024). *Graf (Bagian 1)*. Diakses pada 20 Juni 2025, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf

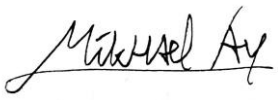
[9] Munir, Rinaldi. (2024). *Graf (Bagian 3)*. Diakses pada 20 Juni 2025, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/22-Graf-Bagian3-2024.pdf

[10] Dokumen penulis

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, 20 Juni 2025



Mikhael Andrian Yonatan
13524051