Simulator Mesin Cipher Hagelin BC-52

Tintin Chandra [1)] Wijaya Kesuma [2)]
STMIK IBBI Medan
Jl. Sei Deli No. 18 Medan, Telp. 061-4567111 Fax. 061-4527548
Email : tinuhnsbm@gmail.com

Abstrak

Simulator merupakan bentuk peniruan terhadap keadaan yang sebenarnya. Program komputer saat ini banyak dikembangkan untuk meniru cara kerja dari beberapa peralatan. Salah satu contohnya adalah Hagelin BC-52 merupakan salah satu mesin kriptografi yang terkenal pada masa perang dunia II yang dimulai dari tahun 1939 hingga 1945 dan masih digunakan setelah Perang Dunia II selesai. Bentuk atau model mesin menyerupai mesin ketik. Digunakan oleh angkatan darat dan angkatan laut Jerman untuk saling berkomunikasi dengan mengirimkan pesan yang telah diacak. BC-52 diciptakan oleh Boris Hagelin dan merupakan salah satu mesin cipher yang paling berhasil secara komersial pada masanya. Dalam perancangan simulasi mesin Hagelin BC-52 menggunakan model waterfall. Hasil dari simulasi mesin Hagelin BC-52 adalah sebuah mesin cipher Hagelin BC-52 yang meniru cara kerja dari mesin ini. Dalam program simulator ini juga dapat dilakukan pengaturan terhadap kunci, lug, pinwheel dan movement bar.

Kata Kunci: Perangkat Lunak, Simulator, Mesin Cipher, Hagelin BC-52

Abstract

Simulator represent imitation form to real situation. Computer program in this time developed many to imitate the way of activity from some equipments. One of the the example is Hagelin BC-52 represent one of the machine of cryptography famous at is second wartime world started from year 1939 till 1945 and still used after finish second wartime. Form or machine model look like typewriter. Used by ground forces and Germany navy to is to communicating with each other to deliver message which have is random. BC-52 created by Boris Hagelin and represent one of the machine of cipher most successful commercially at a period to him. In scheme of machine simulation of Hagelin BC-52 use model of waterfall. Result of from machine simulation of Hagelin BC-52 is a machine of cipher Hagelin BC-52 imitating the way of activity of this machine. In this simulator program also can be done by arrangement to key, lug, and pinwheel of movement bar.

***Keywords**: Software, Simulator, Machine of Cipher, Hagelin BC-52.*

1. Pendahuluan

Pada masa Perang Dunia II, era sebelum lahirnya komputer digital, salah satu cara untuk mengamankan komunikasi pada waktu itu adalah menggunakan mesin *cipher* mekanik. Salah satu mesin *cipher* yang terkenal pada masa itu adalah Hagelin C-52. Mesin ini mempunyai 6 buah *pin wheel* yang bergerak tidak beraturan yang dipilih dari satu set yang terdiri atas 12 buah. Beberapa situs Internet banyak membahas mengenai cara kerja dari mesin ini dan menyertakan program *applet* mini untuk mengkodekan pesan dengan mesin Hagelin BC-52 yang diberi nama berdasarkan nama penemunya yaitu Boris Hagelin yang merupakan seri ke 52 dari pengembangan mesin C-52. Kebanyakan *applet* tersebut tidak menampilkan bentuk *simulator* mesin ini secara nyata. Meskipun dibandingkan dengan teknik kriptografi pada saat ini yang telah sangat modern dan lebih aman, keberadaan mesin BC-52 masih banyak dipelajari sebagai langkah awal dalam mempelajari teknik kriptografi berdasarkan atas mesin *cipher* mekanik. *Cipher* Enigma terkenal lewat kontribusinya pada Perang Dunia II dan dipakai oleh pihak Jerman. Mereka mengembangkannya dan dikenal sebagai Mesin Enigma. Mesin ini berdasarkan atas suatu sistem yang memakai tiga rotor untuk memperoleh huruf *cipher text* dari *plain text*. Rotor dapat berputar secara konjungsi dengan yang lain, jadi mampu menghasilkan substitusi yang bervariasi seperti Caesar Shift.

Ketika suatu huruf diketik pada *keyboard* Mesin Enigma, huruf tersebut pertama dikirim melalui rotor pertama, yang akan menggeser huruf tersebut menurut *setting* yang terbaru. Huruf tersebut kemudian akan dilewatkan ke rotor kedua, dimana huruf tersebut kemudian digantikan oleh sebuah substitusi menurut setting terbaru dari rotor kedua. Huruf baru tersebut kemudian akan dilewatkan lagi ke rotor ketiga, yang mana telah diganti *setting*-nya lagi. Berikutnya, huruf baru tersebut akan diumpan balik melalui reflektor, dan dikirim balik ke ketiga rotor itu kembali dalam urutan terbalik. Trik yang membuat begitu menarik pada saat itu adalah rotor yang dapat berputar. Jika sebuah huruf *plain text* dilewatkan

pada rotor pertama, rotor tersebut akan berputar dengan satu posisi. Dua rotor yang lain akan bertahan hingga rotor pertama telah berputar 26 kali (jumlah huruf dalam alphabet). Kemudian rotor kedua akan berputar satu posisi. Setelah rotor kedua telah berputar 26 kali (26 × 26 huruf, karena rotor pertama telah berputar 26 kali untuk tiap waktu putaran pada rotor kedua), rotor ketiga baru akan berputar satu posisi jika rotor kedua telah berputar 26 kali. Siklus ini terus berlangsung hingga seluruh huruf dari pesan dienkripsi. Hasil dari rotasi tersebut menghasilkan suatu *shifting shift*. Dengan kata lain, huruf "s" dapat dikodekan menjadi huruf "b" bagian pertama dari pesan, dan kemudian sebagai huruf "m" pada pesan berikutnya. Prinsip dari *shifting rotor* ini mempunyai 26 × 26 × 26 = 17.576 kemungkinan posisi rotor.

Kode dan *cipher* telah memainkan banyak aturan-aturan yang krusial selama 3000 tahun lalu. Pada masa Perang Dunia II sejumlah bangsa menggunakan sistem kriptografik untuk mengirimkan maksud dan tujuan rahasia mereka dari mata-mata musuh dimanapun. Salah satu mesin kriptografi yang dikembangkan oleh pihak Jepang adalah mesin dengan kode nama Purple. Pada awal tahun 1930-an, Angkatan Laut Jepang membeli suatu versi komersial dari Enigma Jerman dan memodifikasinya dengan menambah fitur untuk memperkuat keamanannya. Sistem ini merupakan salah satu mesin kriptografi yang paling aman di dunia. Mesin ini diberi kode nama "Red" oleh pemerintah Amerika Serikat dan digunakan untuk mengenkripsi komunikasi politik tingkat paling tinggi antara Jepang dan agen-agennya di seluruh dunia. Setelah serangkaian usaha, American Army Signal Intelligence Service (SIS) menggunakan "*statistical analysis techniques*" yang diciptakan oleh William Friedman untuk memecahkan sistem ini pada tahun 1936 dan mampu membaca komunikasi rahasia tingkat tinggi dari Jepang. Aliran informasi terdekripsi dari Jepang ini tidak bertahan lama, karena pada awal tahun 1939, Menteri Luar Negeri Jepang memperkenalkan suatu mesin cipher yang baru yang disebut "*Purple*". Mesin ini dipertimbangkan lebih canggih dibandingkan dengan mesin "*Red*" dan tidak ada seorangpun dari pihak Amerika Serikat yang mampu memecahkannya.Mesin Purple dapat mengganti suatu huruf tunggal dengan sejumlah ratusan hingga ribuan huruf-huruf pada sejumlah panjang huruf tertentu sebelum huruf tersebut diulang dengan huruf yang diganti sama kembali. Efek ini memberikan mesin Purple mempunyai kemampuan menyembunyikan pesan *plain text* dengan sejumlah langkah-langkah unik pada mesin. Operator dari mesin juga harus mempunyai kemampuan untuk mengubah setting dari *stepping switches*, sebagai tambahan daripada *setting* pada *plugboard*. Jadi diperlukan metode dasar untuk mekanisme enkripsi setiap hari ketika mesin dioperasikan.

Mesin *cipher* jenis *pin* dan *lug* ini dibuat untuk menandingi keberhasilan mesin *cipher* mekanik edisi sebelumnya yaitu C-38/M-209. Ukuran mesin ini adalah 8 1/2 inci x 5 3/8 inci x 4 3/8 inci. Peralatan ini bersifat mekanikal, tetapi dikombinasikan dengan suatu *keyboard* elektronik yang dapat dipasangkan yaitu *keyboard* yang dikenal dengan nama B-52, sehingga perpaduan dari kedua sistem ini diistilah sebagai BC-52. Ukuran dari B-52 lebih besar yaitu berukuran 12 1/2 inci x 8 1/2 inci x 6 x 3/8 inci. Gambar 2.12 memperlihatkan bentuk dari mesin *cipher* Hagelin BC-52.

Baik model C dan CX dilengkapi dengan enam buah *pinwheel* (roda pin). Pada versi C-52, keenam roda ini dipilih dari kemungkinan 12 set roda yang ada, dimana sejumlah *pin* pada setiap roda berjumlah 25, 26, 29, 31, 34, 37, 38, 41, 42, 43, 46 dan 47. Model C telah mempunyai sistem *stepping* yang tetap dengan siklus roda yang lebih besar karena disebabkan oleh penggunaan bilangan prima pada keenam roda tersebut. Versi CX-52 mempunyai 6 *pinwheel* dengan 47 *pin* setiap rodanya dan suatu sistem pergerakan roda yang fleksibel. Kedua model mempunyai tutup yang terdiri atas batang yang dapat digerakkan: 27 dari batang tersebut digunakan untuk proses enkripsi dan sisanya 5 buah batang digunakan untuk mengontrol *stepping* dari *pin wheel*. Model CX yang pertama menggunakan batang pengontrol untuk enkripsi, tetapi karena disebabkan oleh kerumitan dalam membentuk *setting* pada *lug* maka model CX selanjutnya hanya menggunakan batang tersebut untuk mengontrol roda pada proses *stepping*. *Stepping* dari roda-roda CX dikontrol oleh suatu *lug* yang dapat diatur pada bagian batang kontrol.

C-52 dan CX-52 merupakan mesin yang sangat fleksibel yang dapat diproduksi dengan beragam cara, membentuk suatu mesin yang unik dengan karakteristik kriptografi yang unik untuk setiap pelanggan. Kedua model tersebut juga terdapat roda pergerakan yang dapat dibongkar pasang dan posisi roda-roda tersebut pada *drum* dapat diubah, saling dipertukarkan antara roda-roda pencetak, label *pinwheel* yang dapat dibentuk sendiri. Selain itu juga tersedia versi *One-Time Tape reader* selain versi roda, yaitu versi yang hanya khusus melakukan enkripsi terhadap angka-angka, dan banyak detail lainnya yang mempengaruhi proses *enciphering*.

Simulasi adalah proses mencontoh atau mempergunakan gambaran sebenarnya dari suatu sistem kehidupan nyata tanpa harus mengalaminya pada keadaan yang sesungguhnya. Simulasi merupakan satu bahasan dengan cakupan sangat luas dan bersinggungan dengan berbagai bidang ilmu.

Simulasi komputer merupakan salah satu metode yang dapat digunakan untuk meggambarkan fenomena-fenomena fisika secara jelas atau secara *visual* sehingga mudah untuk diamati dan difahami. Demikian juga halnya dengan menjelaskan fenomena pada gerak sistem katrol dari sudut pandang fisika,

metode dan media sangat dibutuhkan untuk mengarahkan manusia berfikir secara holistik melalui hubungan antara fenomena alam dengan tinjauan fisika.

Simulasi adalah program (*software*) komputer yang berfungsi untuk menirukan perilaku sistem nyata (realitas) tertentu. Tujuan simulasi antara lain untuk pelatihan (*training*), studi perilaku sistem (*behaviour*) dan hiburan atau permainan (*game*). Beberapa contoh simulasi komputer, antara lain: simulasi terbang (*flight simulation*), simulasi sistem ekonomi makro, simulasi sistem perbankan, simulasi antrian layanan bank (*service queue*), simulasi *game* strategi pemasaran (*market game*), simulasi perang (*war game simulation*), simulasi mobil (*car simulation*), simulasi tenaga listrik (*power plan simulation*), simulasi tata kota (*sim city*). Simulasi waktu nyata (*real time*) merupakan bagian dari ilmu informatika (teknologi informasi) yang sedang berkembang sangat pesat saat ini.

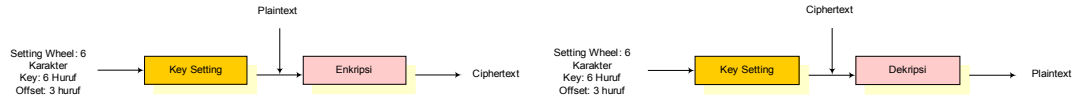
Permainan (*game*) komputer merupakan salah satu jenis simulasi komputer. Beberapa tipe game komputer antara lain: permainan strategi (*strategic game*), permainan ketrampilan tangan dan mata, permainan tantangan (*adventure game*). Permainan strategi (*strategic game*) merupakan permainan papan (*board*), kartu (*card*) atau permainan yang dimainkan pada suatu *grid* (biasanya imajiner), dimana kemenangan dihitung berdasarkan aturan tertentu. Contoh: permainan olah yudha (*war game*), catur (*chess*), *bridge, go-moku*, *command and conquer generals*.

Permainan ketrampilan tangan dan mata adalah permainan yang melibatkan kecepatan dan koordinasi antara ketrampilan tangan dan mata manusia terhadap mesin komputer, umumnya menggunakan tampilan (*screen display*) resolusi tinggi. Contoh: simulasi mobil (*driving game*), simulasi terbang (*flight simulation*), *dxball game*. Dalam permainan tantangan (*adventure game*), program komputer mentranslasikan tanggapan pemain (*player response*) terhadap suatu kejadian (*event*) baik atau buruk dalam menyelesaikan persoalan. Contoh: *puzzle*, *zork*, *delta force black hawk down*, *beach head*. Bagian-bagian *game* komputer terdiri dari: struktur data (*data structure*), metode evaluasi (*evaluation method*), dan antarmuka pengguna (*user interface*). Struktur data dalam permainan (*game*) adalah organisasi logis informasi perihal papan (*board*), potongan permainan (*playing piece*), gerakan (*move*) dan kemenangan (*winning*) serta kekalahan (*losing*). Contoh: representasi agregat (dalam simulasi olah yudha), *variabel record* (dalam permainan catur).

Metode fungsi evaluasi dalam permainan (*game*) akan menguji gerakan (*move*) yang mungkin, memberi nilai (*score*) gerakan tersebut. Kemampuan melihat ke depan (*search*) merupakan putusan kritis dalam permainan strategi komputer. Beberapa metode melihat ke depan (*looking ahead*): *minimax search algorithm*, *alpha beta search algorithm*. Antar muka pengguna (*user interface*) dengan komputer (*machine*) dirancang sedemikian rupa sehingga pemain (*player*) hanya akan berkonsentrasi pada permainan dan tidak dibebani perihal cara operasi program komputer. Antarmuka pengguna saat ini melibatkan *multimedia* (suara, gambar dan animasi).

2. Metode Penelitian

Adapun bagan dan cara kerja dari mesin kripto BC-52 dimana mesin diatur dari *wheel* dan *drum* yang berbeda untuk menghasilkan bentuk konfigurasi yang berbeda. Pada gambar 1. memperlihatkan bentuk blok diagram untuk proses enkripsi dan dekripsi dari mesin kripto BC-52.



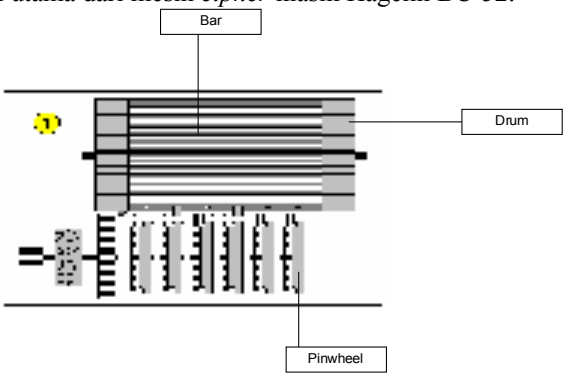
Gambar 1. Blok Diagram Proses Enkripsi dan Dekripsi

Pada diagram blok pada gambar 1., adapun cara kerjanya adalah *input* dilakukan mencakup *setting wheel* (6 karakter), *key* (6 huruf), dan offset (3 huruf). Selanjutnya dilakukan proses *key setting* terhadap *pinwheel* ataupun generasi kunci. Kemudian dengan proses enkripsi dengan melakukan perputaran pada *pinwheel* dan *lug* pada mesin *cipher* BC-52 antara *plaintext* dan untaian karakter *plaintext* selesai. Proses dekripsi hanya berjalan secara reversibel dari proses enkripsi.

Hagelin BC-52 merupakan kombinasi dari mesin C-52 dengan sebuah *keyboard* elektronik B-52. Mesin ini dikembangkan dan dimanufaktur oleh Hagelin Cryptos (Crypto AG). BC-52 merupakan mesin *cipher* mekanikal *drum* dan *lug*. Baik hasil *plain text* dan *cipher text* akan dicetak pada sebuah pita kertas.

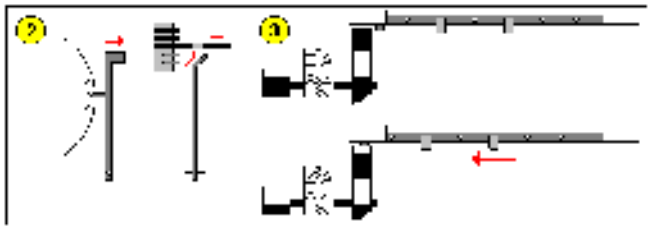
C-52 dapat dibongkar pasang dari basis *keyboard* dan dapat dioperasikan secara manual. Dalam kasus seperti ini operator menggeser *knob* huruf hingga huruf yang diinginkan muncul pada *letter dial* dan menekan *handle* pada sisi kanan bawah. Ketika di-reset, maka ini juga akan dilakukan *reset* terhadap grup *counting* dari mekanisme pencetakan.

Sebuah *wheel* pencetakan ganda mempunyai satu susunan alphabet normal dan satu lagi saling berlawanan. Enkripsi dilakukan dengan mengatur *wheel* alphabet normal dalam bentuk *plain letter* dan kemudian menambahkan sejumlah *step*. Pada posisi baru tersebut dari *wheel* pencetakan *cipher text* dicetak dengan menggunakan *print wheel* yang berlawanan tersebut. Langkah-langkah menghasilkan bilangan semu acak ditentukan oleh *setting* dari *lug* pada *drum* dan *pin* pada *wheel*. Gambar 2. memperlihatkan konstruksi utama dari mesin *cipher* klasik Hagelin BC-52.



Gambar 2. Susunan *Lug* dan *Pinwheel* BC-52

Di dalam BC-52 terdapat sebuah *drum* yang terdiri atas 32 *bar*. *Lug* yang kecil dapat ditambahkan dengan satu hingga enam posisi pada setiap bagian *bar*. Ketika *user* mengatur *handle* atau menggunakan *keyboard* maka *drum* akan membuat satu putaran penuh. Gambar 3. menunjukkan cara pengaktifkan *drum* oleh *pinwheel*.

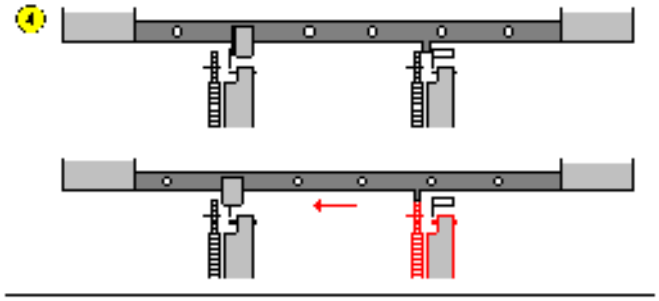


Gambar 3. Pengaktifkan *Drum* Oleh *PinWheel*

Di depan dari *drum* ini terdapat enam buah *pinwheel*, semuanya mempunyai seubah *pin* kecil untuk setiap posisi dari *wheel*. *Pin* ini dapat ditempatkan pada sisi kanan atau kiri. Setiap *wheel* mempunyai suatu lengan pengatur (*guide arm*) dengan *slope end* yang akan bergerak ke depan *drum* jika diaktifkan oleh sebuah *pin*. *Drum* akan menggerakkan gigi dari *printing wheel*. Oleh karena itu, jumlah langkah yang ditambahkan pada huruf *plain* merupakan jumlah gigi roda yang terdapat pada *drum*.

Pergerakkan dari 6 (enam) *wheel* juga bergantung pada *setting lug* pada *drum* dan *pin* pada *wheel*. Jika diberi nomor *wheel* dari kiri ke kanan, *wheel* pertama bergerak pada setiap siklus dari *drum*. Sementara 5 (lima) *wheel* digerakkan oleh 5 *bar* khusus. Setiap *bar* tersebut identik terhadap *bar* yang lain tetapi juga mempunyai *pin* kecil yang tetap. Jika suatu lug seperti bar mengenai *active guide arm*, maka ini akan mengakibatkan *lug* bergeser ke kiri, seperti halnya dengan *bar* yang normal. Namun, dalam kasus *bar-bar* khusus ini, *pin* kecil yang tetap sekarang akan ditempatkan ke kanan di depan dari roda *wheel* kecil, dimana langkah dari *pinwheel* akan satu langkah ke depan.

Pin saat ini berada pada *wheel* 2 dan merupakan posisi aktif dan pada *drum*, *advance bar* untuk *wheel* 5 mempunyai sebuah *lug* pada posisi kedua seperti pada gambar 4. Pada saat *drum* berputar, *lug* akan mengenai *guide arm* dari *wheel*, menekan *bar* pada kiri dan membiarkan *fixed pin* kecil memajukan *wheel* 5.



Gambar 4. Ilustrasi Pergerakan *Wheel*

Advance bar akan bertanggung jawab terhadap pergerakkan tidak teratur dari *wheel* dan ini merupakan bagian kritis dari keamanan suatu mesin kriptografik. Ketika dalam contoh berikut semua *setting* 5 bar telah melewati *guide arm*, suatu *active pin* pada *wheel* 1 akan bergerak pada *wheel* 2, 3, 4, 5 dan 6, karena terdapat suat *lug* pada posisi 1 pada setiap dari 5 bar. Suatu pin pada *wheel* 2 akan menggerakkan *wheel* 3, 4, 5 dan 5. Pin pada *wheel* 3 akan menggerakkan *wheel* 4, 5, dan 6 dan seterusnya. Dengan cara seperti ini, semua *wheel* akan bergerak dan akan berhenti dalam satu waktu, bergantung pada pin pada *wheel*. Hal ini akan membentuk suatu pergerakan *wheel* yang tidak teratur dalam urutannya. Untuk menambahkan kerumitan pada proses enkripsi maka digunakan *offset* pada *printwheel*. Ini dilakukan dengan menarik *dial knob* dari mesin, jadi memutuskan hubungan antara plain dan *cipher printwheel*, dan kemudian mengatur *dial* dengan memberikan sejumlah langkah. Ketika proses *enciphering* huruf H dengan menggunakan offset 2 (dua) huruf maka mesin sebenarnya sedang mengenkripsi huruf F.

Pemilihan kunci pesan (*message key*) dilakukan dengan membentuk 12 huruf acak. Orang-orang cenderung membentuk pola ketika melakukan pemilihan huruf-huruf acak tersebut. Cara termudah adalah dengan memilih kunci-kunci tetangga pada *keyboard*, mengulang urutan ataupun menggunakan inisial. Oleh karena itu sedikit prosedur digunakan untuk memastikan keacakan dari huruf-huruf tersebut. Sebagai posisi awal dari 6 *wheel* diatur pada posisi acak dan 12 huruf-huruf acak dimasukkan dengan menggunakan *keyboard*. Hasilnya berupa 12 huruf-huruf acak dimana ini akan dipecah menjadi 2 grup yang terdiri atas 6 huruf per grup. Grup 6 (enam) huruf pertama digunakan sebagai posisi awal (*start position*) untuk mengenkripsi grup kedua, yang merupakan *message key* sebenarnya untuk pesan yang akan diproses. Setelah kedua grup ini suatu indikator grup 3 (tiga) huruf, disebut juga sebagai trigram ditentukan. Trigram ini merujuk kepada tabel rahasia ataupun mengindikasikan untuk keperluan khusus. Terdapat banyak cara yang berbeda atas penggunaan trigram ini. Trigram dapat digunakan sebagai pengaturan *offset* dari *printwheel*, mengidentifikasikan metode enkrpsi dan/atau penggunaan kunci. Dalam contoh ini akan digunakan bentuk trigram XXH untuk mengindikasikan *printwheel offset* berada pada posisi H (A = H).

Lakukan pengaturan 6 (enam) *wheel* pada mesin BC-52 dari kiri ke kanan pada posisi berturut-turut D, K, N, W, Q, dan L (Jika huruf-huruf tersebut tidak tersedia pada *wheel labeling* maka atur *wheel* pada posisi awal) dan pastikan bahwa tidak terdapat *printwheel offset* (A = A). Dengan *key setting* seperti di atas maka akan menghasilkan karakter ODUMIQ MIB dimana hasil inilah yang akan dikirimkan kepada penerima beserta dengan pesan terenkripsi, sehingga dapat dilakukan enkrpsi terhadap grup 6 (enam) huruf kedua dengan menggunakan kunci MOXZVT, yang merupakan grup 6 huruf kedua, sebagai posisi awal (*start position*) *wheel* dan *printwheel offset* H.

Algoritma merupakan langkah-langkah maupun urutan bertahap dan spesifik dari suatu masalah. Algoritma ini kemudian diterjemahkan ke dalam program dengan menggunakan bahasa pemrograman tertentu. Algoritma digunakan untuk menganalisa serta menjelaskan urutan dan hubungan antara kegiatan-kegiatan yang akan ditempuh. Selain itu algoritma juga berfungsi untuk menyelesaikan suatu permasalahan sehingga tercapai tujuan yang diinginkan.

1. Algoritma Enkripsi dan Dekripsi BC-52

```
Untuk setiap PinWheel
  ReadPin
Set KeyOffset posisi 0
Untuk perputaran lug 1 hingga 32
  Ambil nilai AdvanceBar
  Untuk PinWheel 1 hingga 6
        Jika tidak terdapat nomor pin yang diset
              Baca setting pin
        Jika pin tidak dihold, ReadPin
```

```
                Lakukan pengecekan terhadap sliding bar
                Jika pin pada lug dan PinWheel dalam posisi aktif
                        Maka bar slided
                        Jika AdvanceBar kosong maka
                                KeyOffset bertambah 1
                        Jika modus BarStepping dalam posisi 1 atau 3
                                Maka putar wheel(AdvanceBar)
                        Jika posisi CipherBar true maka
                                KeyOffset bertambah 1
                Jika not bar slided maka
                        Jika AdvanceBar <> 0 maka
                                Jika modus BarStepping dalam posisi 2 atau 3
                                        Putar wheel(AdvanceBar)
```

2. **Algoritma Mengatur *Wheel* BC-52**

```
Untuk PinWheel 1 hingga 6
Set Nilai Wpin(wheel) sama dengan  "0" hingga 47 karakter
Untuk setiap PinWheel 1 hingga 6
   Set Wmax(wheel) sama dengan  PanjangDari PinWheel
   Set Wsel(wheel) sama dengan  W_len(Len(Wpin(wheel)))
   Set Wpos(wheel) sama dengan  1
   Set PosMemo(wheel) sama dengan  1
Untuk setiap PinWheel 1 hingga 6
   Panggil Prosedur SetWheelView
```

3. **Algoritma *Move Wheel***

```
Set Nilai Wpos(wheel) sama dengan  Wpos(wheel) + 1
Jika Wpos(wheel) lebih besar Wmax(wheel) Maka Wpos(wheel)
Sama dengan Wpos(wheel) - Wmax(wheel)
Panggil Prosedur SetWheelView
```

4. **Algoritma *Reset Wheel***

```
Untuk PinWheel 1 hingga 6
   Set Wpos(k) sama dengan  1 - (LabelView(Wsel(k)) - 1)
Jika Wpos(k) < 1 Maka Wpos(k) sama dengan  Wpos(k) + Wmax(k)
   Panggil Prosedur SetWheelView
```

5. **Algoritma Menampilkan Teks Hasil**

```
Jika aLetter lebih besar 26
Maka Set aLetter sama dengan aLetter - 2
Set LastDialView sama dengan aLetter
```

6. **Algoritma Autotyping**

```
Untuk setiap karakter
   tmp sama dengan  karakter pesan
      Jika (tmp >=”A” dan tmp <= ”Z”) atau tmp = karakter spasi
Jika modus cipher And tmp = karakter spasi
  Maka tmp = ASpaceLetter
       Jika tmp <> karakter spasi
          Panggil Prosedur Enkripsi
```

3. Hasil dan Analisis

Program ini merupakan simulasi yang akurat dari Hagelin BC-52. Setelah berhasil mengembangkan peralatan *cipher* taktikal C-38 dan M-209, maka Hagelin mengembangkan suatu mesin *cipher* untuk tujuan enkripsi militer dan diplomatik. Simulator mesin *cipher* BC-52 dirancang untuk mengenkripsi pesan secara realistik dan akurat, dimana pengguna akan merasa seperti menggunakan mesin *cipher* BC-52 yang asli. Semua menu-menu juga tersedia dengan adanya *Function Key*. Semua objek seperti *button*, *wheel* dan bagian mesin lainnya yang dapat diakses akan ditampilkan dalam bentuk *icon* berupa tangan. Pada bagian *icon* berupa *speaker* maka *user* dapat mengaktifkan dan menonaktikan efek *sound* termasuk bagian *about* dan tombol untuk keluar dari program.

Pada bentuk tampilan utama *simulator* mesin *cipher* BC-52 dimana bagian ”*Encode*” dan ”*Decode*” digunakan apakah mode dari mesin *cipher* ini digunakan untuk melakukan enkripsi ataupun dekripsi. Bagian *counter* digunakan untuk menampilkan jumlah huruf yang telah diproses. Tampilan *printwheel offset* disampingnya digunakan untuk menset posisi *printwheel offset*. Sedangkan bentuk tampilan dari *pinwheel* digunakan untuk menampilkan posisi huruf pada keenam *pinwheel* dan *keyboard*

di bawahnya digunakan untuk meng-*input* teks untuk diproses (di-*encode* / di-*decode*). Pada bagian kiri *user* juga mengakses beberapa tampilan bagian ini dengan menggunakan pilihan menu. Gambar 5. memperlihatkan tampilan utama simulator mesin *cipher* BC-52.

Gambar 5. Tampilan Utama *Simulator* Mesin *Cipher* BC-52

Baik pengirim ataupun penerima harus mengatur konfigurasi BC-52 secara tepat dengan cara yang sama. *Key setting* terdiri atas 5 (lima) bagian. Pilihan dan urutan dari *pinwheel*, pengaturan *pin* pada *wheel*, lug, *printwheel offset* dan posisi awal dari *pinwheel*. Untuk mengatur key maka *user* dapat mengklik pada bagian tulisan HAGELIN BC-52 CRYPTOS di atas *pinwheel*. Pilihan dan pengaturan dari elemen variabel dari BC-52 disebut dengan *key setting*. Untuk memperoleh kualitas enkripsi yang baik dengan tingkat keamanan yang tinggi terdapat beberapa aturan yang harus diikuti. Pada bagian ini akan diberikan beberapa rekomendasi untuk membentuk *key setting* yang baik.

Pada bagian pin maupun *lug setting* Hagelin Cryptos memberikan beberapa saran sebagai berikut: direkomendasikan untuk memilih suatu pola untuk *setting pin* dan *lug*. Suatu metode yang digunakan adalah pengacakan secara statistikal. Cara yang paling sederhana adalah melakukan *flipping* pada sebuah koin dan menulis hasilnya (tampak muka = pin aktif, terbalik = pin tidak aktif). Pada setup bentuk model CX-52 untuk *movement bar wheel* khusus dimana *bar* tersebut juga dapat digunakan untuk proses enkripsi. Ketika menggunakan *movement bar* dimana juga digunakan sebagai sebagai *lug* enkripsi maka harus mengikuti aturan seperti yang telah dijelaskan pada bagian di atas, selain itu juga dipastikan untuk membentuk siklus *stepping* yang baik pada *wheel*. Oleh karena itu, disebabkan oleh komplikasi pada saat persiapan maka pola *lug* yang dapat diterima pada model CX-52 menggunakan *bar* pergerakan khusus secara eksklusif untuk proses *stepping* dan tidak digunakan untuk proses enkripsi.

Untuk melihat *bar* mana yang bereaksi terhadap pergerakan, dan apakah *bar-bar* tersebut juga digunakan untuk proses enkripsi, maka dapat dicek pada bagian BC-52 setup dengan menekan tombol F10. Pergerakan *Lug* bagian dari *setting* dari *lug* untuk proses enkripsi, maka *user* harus mengatur *lug* pada bagian pergerakan *bar* khusus dan memastikan bahwa terdapat variasi yang baik dalam *stepping* dari *wheel*. Setting dari *key* dapat dilihat seperti pada gambar 6.

Gambar 6. Tampilan *Key Setting*

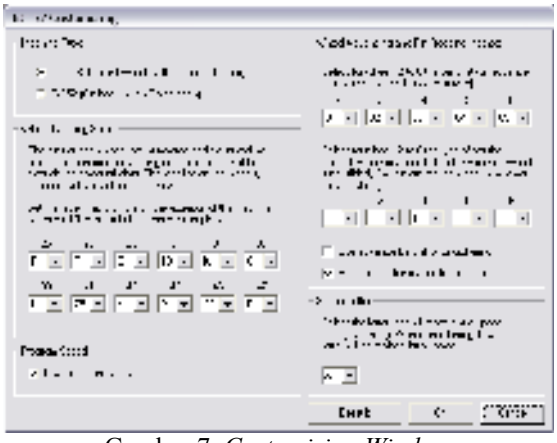
Dari hasil *key setting* yang ditentukan di atas maka dapat ditampilkan dalam bentuk *window key setting* dengan memilih pada menu **View Key** atau tombol *shortcut* **F8**. Bentuk tampilan dari *key setting* ini diperlihatkan seperti pada gambar 7.

Gambar 7. Tampilan Contoh *Key Setting*

Pilihan dari *wheel* tergantung pada model yang dipilih. C-52 menggunakan 6 buah *wheel*, dipilih dari 12 *wheel* yang berbeda dengan jumlah pin 25 hingga 47. Model CX-52 menggunakan 6 buah *wheel* yang identik dengan 47 buah *pin* pada setiap *wheel*. Model ini dapat dipilih pada bagian **Customizing Windows** (F10). Dengan mengubah model maka semua *pin* dan *lug* akan di-*reset* kembali.

User harus memilih kumpulan dari 6 *pinwheel* yang berbeda. Setiap *wheel* dirancang dengan suatu angka antara 25 dan 47 dimana ini juga merupakan jumlah *pin* dari *wheel*. Dalam *window key setting*, *user* dapat melihat 6 *wheel* yang sedang dipasangkan pada mesin *cipher* BC-52 seperti pada gambar 7. option 1. Klik pada salah satu dari 6 *wheel* tersebut untuk diekstrak ke dalam mesin *cipher* BC-52. *Wheel* sekarang ditunjukkan pada bagian tengah dari *window* dengan semua *pin setting* yang dapat dilihat. *User* dapat mengatur *pin setting* dari *wheel* dengan mengklik pada nomor *pin* untuk mengaktifkannya (merah) ataupun menonaktifkan (putih). *Wheel* yang terekstraksi tersebut dapat ditempatkan ke *wheel box* di bawah atau di tempat kosong pada mesin *cipher* BC-52. *Wheel* yang lain dapat dipilih dari *wheel box* dan ditempatkan dalam BC-52 setelah dilakukan pengaturan *pin*.

Model CX-52 mempunyai 6 buah *pinwheel* dengan 47 buah *pin* untuk setiap *pinwheel*. Klik pada salah satu dari 6 *wheel* tersebut untuk mengekstraksinya dari BC-52 seperti pada gambar 7. option 2. *Wheel* ini sekarang ditunjukkan pada bagian tengah dari *window* dan semua *setting pin*-ya akan terlihat. *User* dapat mengatur *setting pin* pada *wheel* tersebut dengan mengklik angka *pin* untuk mengaktifkan (merah) dan menonaktifkan (putih) pada bagian *pin* tersebut. Ketika semua *pin* telah diatur pada *wheel* maka tempatkan kembali pada BC-52.



Gambar 7. *Customizing Window*

User juga dapat memilih apakah 5 bar pergerakkan khusus digunakan untuk melakukan proses enkripsi atau hanya berfungsi sebagai pergerakkan saja (secara default hanya berfungsi untuk pergerakkan) dan menentukan apakah *pin pawl* disimpan dalam posisi selama siklus dari *drum* atau *pin* berubah ketika *wheel* di-*advanced* selama proses siklus.

BC-52 mempunyai 32 *sliding bar* pada *drum*. 27 dari *bar* tersebut hanya digunakan untuk proses enkripsi, dan 5 *bar* lainnya digunakan sebagai *advance wheel* 2,3,4,5 dan 6 (diberi nomor dari kiri ke kanan, dan *wheel* 1 selalu merupakan *step*). Semua dari *bar* pada drum akan melewati *pin lever* sekali pada satu siklus penuh dari *drum*.

Gunakan *scroll bar* untuk melihat 32 *bar*, dan klik pada *lug* untuk menempatkan (merah) ataupun menghilangkannya (putih) pada bagian *lug*. Secara umum 1 ataupun 2 *lug* digunakan pada *bar* yang sama, tetapi beberapa *key setting* menggunakan hingga 3 *lug*. Terdapat 5 (lima) buah bar khusus yang akan membuat *wheel* bergerak dalam bentuk yang tidak teratur. Setiap *bar* dirancang memiliki satu *wheel*, kecuali untuk *wheel* pertama dimana akan selalu bergerak. Sebagai contoh jika suatu *lug* ditempatkan pada posisi kedua dari *bar* yang bertanggung jawab untuk menggerakkan *wheel* 5, maka *bar* ini akan bergerak ke kiri dan menggerakkan *wheel* 5 satu langkah ke depan jika *pin* saat ini yang berada pada *wheel* kedua aktif.

Dalam *setup* CX-52 sebelumnya *bar-bar* khusus ini juga digunakan untuk proses enkripsi. Namun, disebabkan oleh komplikasi dalam proses persiapannya dan pola-pola *lug* yang dapat diterima, kemudian model CX-52 menggunakan *bar* pergerakkan khusus hanya untuk proses *stepping* dan tidak lagi digunakan untuk proses enkripsi. Untuk melihat *bar* yang mana yang bertanggung jawab untuk pergerakan, apakah *bar-bar* tersebut juga digunakan untuk proses enkripsi, maka dapat dicek pada bagian *setup* BC-52 dengan menggunakan tombol F10. *Lug* pada *bar-bar* ini bersifat kritikal untuk keamanan kriptografik dari mesin. Di bawah ini diberikan bentuk setup normal di mana bar 1 menggerakkan bar 2, 2 menggerakkan 3 dan seterusnya.

Pada *setup* ini, suatu *pin* pada *wheel* 1 di depan dari *lug* 1, akan menggerakkan *wheel* 2, 3, 4, 5 dan 6 ketika berturut-turut *bar* 1, 2, 3, 4 dan 5 melewati *pin* tersebut. *Pin* pada *wheel* 2 akan menggerakkan *wheel* 3, 4, 5 dan 5, pin pada *wheel* 3 akan menggerakkan *wheel* 4,5 dan 6 dan seterusnya. Dengan cara seperti ini, semua *wheel* akan bergerak pada setiap waktu dan hanya akan berhenti pada suatu waktu, tergantung pada *pin* pada *wheel*. Ini membentuk suatu urutan perputaran *wheel* yang tidak beraturan.

User dapat menyimpan *setting key* saat ini yang dibuatnya ataupun melakukan *loading* terhadap *setting key* yang telah ada sebelumnya melalui menu *simulator*. *File-file* ini kemudiaan akan disimpan dalam ekstensi .C52. Ketika memulai *simulator* maka *key setting* yang terakhir yang di-*load* secara otomatis. Pada saat keluar maka *user* akan diberi konfirmasi apakaah akan menyimpan perubahan tersebut atau tidak.

Sebelum melakukan proses enkripsi maka *user* harus melakukan pengaturan posisi awal dari 6 *wheel*, dimana proses ini disebut sebagai *message key*. Ubah posisi dari *pinwheel* dengan mengklik pada bagian atas atau bagian bawah dari posisi garis tengah *wheel*. Pilihan modus **Encode** atau **Decode** dengan mengklik pada bagian atas *label* pada mesin tersebut. Pada modus **Encode,** gunakan huruf X sebagai pengganti spasi (dapat diubah dengan menggunakan F10). Pada modus **Decode,** huruf X akan digantikan dengan spasi.

Jika *user* mengklik pada *paper ribbon* (*text box input* dan *output*) ataupun menekan tombol F5, maka **Window Smart Clipboard** akan ditampilkan. *User* dapat memilih secara lengkap *plain text* dan *cipher text*-nya serta menyimpan teks pada *clipboard* tersebut dalam bentuk teks ASCII. Adapun bentuk tampilan dari *window smart clipboard* diperlihatkan seperti pada gambar 8.

Gambar 8. *Window Smart Clipboard*

Jika *user* mempunyai sejumlah besar *plain text* yang panjang, maka *user* dapat menggunakan **Auto Typing Window**. *Window* ini akan muncul jika *user* menekan tombol F6. Pada *window* ini *user* dapat mengetik ataupun mem-*paste* potongan dari *text*, ataupun menerima *content* dari *clipboard Windows*. *User* juga dapat memiih 3 jenis kecepatan ketikan. Pilih '*Start*' untuk memulai proses terhadap teks tersebut. Pastikan bahwa semua *setting* mesin sebelum melakukan **Auto Typing**. **Auto Typing** hanya akan memproses masukan berupa huruf (dan karakter spasi pada modus enkripsi) dan akan mengabaikan semua karakter yang lain. Pilih '*Load*' untuk menempatkan *file* teks ASCII ke bagian ini. Adapun bentuk tampilan dari *window auto typing* diperlihatkan seperti pada 9.

Gambar 9. *Window Auto Typing*

4. Kesimpulan

Dari hasil hasil dan analisis pada perangkat simulasi ini dapat diambil beberapa kesimpulan antar lain simulator BC-52 ini hanya mampu memproses pesan dalam bentuk karakter dan menghasilkan cipher text berupa karakter yang teracak. Kemudian key setting yang telah diatur dapat disimpan dalam bentuk file sehingga dapat digunakan kembali. Adanya fasilitas ini, membantu user untuk membuat beberapa setting kunci yang berlainan serta adanya fasilitas autotyping sehingga membantu user untuk memasukkan pesan yang panjang hingga mencapai 64 KB.

[1] Daftar Pustaka

[1] Cormen, H., Leiserson, E., Rivest, L., 1990, Introduction to Algorithms, Mc Graw Hill Book Company.
[2] Hariyanto, B., 2003, Struktur Data, Edisi Kedua, Informatika, Bandung.
[3] Munir, R., Lidia L., 2002, Algoritma dan Pemrograman, Edisi Kedua.
[4] Roger S. Pressman, Ph.D., 2002, Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku Satu), Mc Graw-Hill Companies, Inc, Penerbit Andi.
[5] Wu, S. dan U. Manber, 1994, A Fast Algorithm for Multi-Patterns Searching.
[6] Wahana Komputer, 2010, Buku Tutorial 5 Hari: Belajar Pemograman Visual Basic, Penerbit Andi.