Transposisi Pesan dengan Metode Pemetaan Bresenham

Rubiano Adityas - 13510041
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510041@std.stei.itb.ac.id

***Abstract*—Makalah ini membahas tentang transposisi pesan dengan menggunakan fungsi pembentuk garis bresenham, yang kerap digunakan untuk membentuk garis pada bidang ilmu visualisasi komputer. Selain pembahasan mengenai metode penggunaan fungsi bresenham tersebut, juga dipaparkan hasil analisis mengenai kekuatan transposisi pesan tersebut.**

***Index Terms*—Bresenham, fungsi pemetaan, kriptografi.**

I. PENDAHULUAN

Kriptografi merupakan cabang ilmu yang esensial dalam bidang keamanan informasi, di zaman dimana penggunaan teknologi informasi sangat tinggi, dan jumlah transmisi data tentu sangat banyak. Suatu pesan yang dikirim bisa diserang oleh pihak ketiga (*adversaries*), sehingga pihak ketiga dapat mengetahui isi pesan, atau mengubah isi pesan tersebut. Apabila data yang ditransmisikan memiliki nilai kerahasiaan yang tinggi (misal: rencana bisnis perusahaan dagang), maka tentu penyerangan terhadap pesan akan sangat merugikan. Oleh karena itu, dikembangkanlah berbagai macam metode kriptografi untuk mengamankan pesan/informasi yang ditransmisikan. Kriptografi secara umum bisa dibagi menjadi dua, klasik dan modern. Usia ilmu kriptografi modern relatif muda, namun kriptografi klasik sudah ada sejak zaman yunani kuno.

Perbedaan utama antara kriptografi klasik dan modern, ialah kriptografi modern menggunakan bit sebagai satuan terkecil komponen pesan, sementara kriptografi klasik menggunakan karakter. Pada hakikatnya, baik kriptografi klasik maupun modern, tersusun atas dua bagian, yakni substitusi dan transposisi. Desain dua bagian tersebutlah yang membedakan metode kriptografi yang satu dengan yang lainnya. Berbagai desain dikemukakan dan diusulkan oleh para matematikawan dan kriptograf, dan masing-masing desain memiliki kelebihan dan kekurangan tersendiri.

Perkembangan ilmu komputer meningkat secara eksponensial, sejak ditemukannya mesin komputasi pertama, hingga saat ini. Perkembangan ilmu komputer menghasilkan banyak sekali cabang ilmu komputer, yang mempelajari berbagai aspek yang spesifik. Kriptografi merupakan salah satunya, dan contoh lainnya adalah visualisasi komputer (grafika). Visualisasi komputer mempelajari cara untuk merepresentasikan data secara visual ke layar, dengan memanfaatkan rumusan matematika dan bahasa pemrograman tingkat rendah. Salah satu fungsi pada ilmu visualisasi komputer adalah fungsi bresenham, yang digunakan untuk membentuk sebuah garis lurus yang tidak terpotong.

Pada kesempatan kali ini penulis ingin memaparkan sebuah gagasan desain algoritma kriptografi kunci simetri yang baru, berfokus pada bagian transposisinya. Fungsi transposisi yang penulis desain, memanfaatkan fungsi bresenham pada visualisasi komputer untuk segmentasi pesan.

II. TEORI DASAR

A. Kriptografi

Kriptografi merupakan ilmu untuk menjaga kerahasiaan suatu pesan. Ada empat tujuan mendasar dari ilmu kriptografi, yang pertama adalah kerahasiaan, yakni usaha untuk menjaga isi dan informasi dari siapapun yang tidak berhak untuk mengetahuinya. Kemudian yang kedia adalah integritas data, yakni berhubungan dengan pengubahan data yang dilakukan secara ilegal. Tujuan ketiga adalah autentikasi, yakni berhubungan dengan identifikasi dari sesuatu sistem atau informasi itu sendiri. Tujuan yang terakhir adalah don-repudasi, usaha untuk mencegah adanya penyangkalan terhadap suatu informasi.

Kriptografi biasanya dilakukan pada suatu pesan berupa teks. Kemudian pesan tersebut akan diolah dengan mekanisme atau algoritma tertentu untuk mengubah pesan tersebut. Dengan mengubah pesan tersebut, orang lain akan sulit untuk mengenali dan mengerti pesan tersebut. Beberapa elemen yang berperan dalam suatu kriptografi antara lain adalah plaintext (teks pesan yang akan dienkripsi), key (nilai rahasia yang kerap dipakai sebagai parameter fungsi substitusi dan transposisi), dan ciphertext (pesan yang sudah dienkripsi).

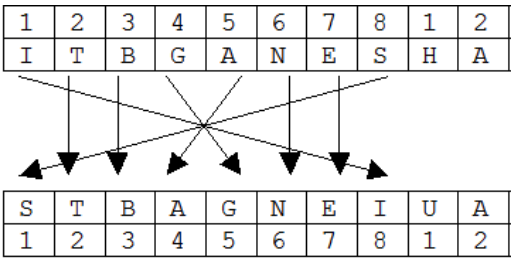
Kriptografi merupakan ilmu yang tua. Banyak metode kriptografi yang ditemukan sudah digunakan sejak dahulu. Kriptografi zaman dahulu menyamarkan pesan yang tersusun atas karakter-karakter. Ketika dunia mencapai era informasi, orang mulai berkomunikasi dengan transmisi data dalam bentuk bit-bit, maka dari itu ilmu kriptografi pun menyesuaikan. Penggolongan kriptografi menjadi kriptografi klasik dan modern didasarkan pada hal tersebut.

Baik kriptografi modern, maupun kriptografi klasik, pada dasarnya terdiri dari dua bagian, substitusi dan

transposisi. Fungsi substitusi, sesuai namanya, memetakan satu/berberapa komponen pesan (karakter/bit), menjadi satu/komponen lainnya. Fungsi transposisi menukar urutan posisi dari elemen-elemen pesan. Penggabungan kedua fungsi tersebut menghasilkan fungsi enkripsi/dekripsi yang kuat, disebut sebagai *super-encryption*.

Berdasarkan metodenya, fungsi substitusi dibagi menjadi empat jenis, antara lain: cipher abjad tunggal (monoalphabetic cipher), cipher substitusi homofonik (homophonic substitution cipher), cipher abjad majemuk (polyalphabetic substitution cipher), dan cipher substitusi poligram (polygram substitution cipher). Keempat jenis fungsi substitusi tersebut memiliki keunggulan masing-masing. Setiap jenis mempunyai cara untuk enkripsi yang berbeda, dan cara yang digunakan untuk mendekripsinya pun juga berbeda.

Untuk fungsi transposisi (atau juga kerap disebut sebagai fungsi permutasi), belum ada penggolongan metode seperti fungsi substitusi. Contoh fungsi transposisi pesan adalah sebagai berikut:

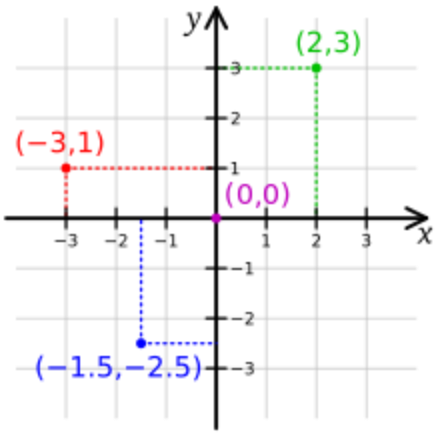


Gambar 2.1 Transposisi sederhana

Pada gambar 2.1, pesan dibagi menjadi segmen dengan panjang delapan karakter, dan kemudian dilakukan penukaran posisi karakter pada segmen tersebut, sesuai dengan arah panah yang terlihat pada gambar.

B. Sistem Koordinat Kartesian

Dalam ilmu matematika, sistem koordinat kartesian adalah sebuah sistem koordinat yang memberikan spesifikasi unik untuk tiap titik pada bidang. Spesifikasi yang diberikan berupa sepasang angka (koordinat), yang merepresentasikan jarak titik tersebut terhadap dua buah garis sumbu yang saling tegak lurus, yang memiliki satuan panjang yang sama dengan nilai jarak pada spesifikasi titik. Koordinat terdiri dari absis, jarak titik ke garis sumbu horizontal (sumbu x), dan ordinat, jarak titik ke garis sumbi vertikal (sumbu y). Titik dimana kedua garis sumbu berpotongan adalah titik nol, yang bernilai (0,0).



Gambar 2.2 Bidang 2 Dimensi dengan Sistem Koordinat Kartesian

Sistem koordinat kartesian juga bisa digunakan untuk mengspesifikasi titik pada ruang tiga dimensi, dengan menggunakan tiga koordinat kartesian, yang memiliki aturan sama dengan sebelumnya. Apabila dilihat dar perspektif yang lebih umum, sistem koordinat kartesian bisa memuat dimensi berapapun (misal n dimensi), dengan representasi koordinat kartesian sejumlah n.

Sistem koordinat kartesian ditemukan pada abad XVII masehi oleh René Descartes (nama latin: Kartesius). Penemuannya membawa revolusi pada ilmu matematika karena sistem tersebut mampu memberikan hubungan yang sistematis antara geometri dan aljabar. Dengan menggunakan sistem koordinat kartesian, bentuk geometris bisa dideskripsikan dengan persamaan kartesian, yakni persamaan aljabar yang melibatkan koordinat titik-titik garis yang membentuk visualisasi bentuk geometris tersebut pada bidang.

C. Algoritma Garis Bresenham

Sesuai namanya, algoritma ini ditemukan oleh Jack Elton Bresenham. Algoritma garis bresenham adalah sebuah algoritma yang menentukan urutan titik pembentuk garis, untuk aproksimasi sebuah garis lurus yang melewati dua buah titik yang dispesifikasikan. Algoritma ini kerap digunakan untuk menggambar garis pada layar komputer, karena operasi yang digunakan hanyalah penambahan, pengurangan, dan bit shifting, yang mana proses-proses tersebut relatif murah (menggunakan sumber daya yang minim) dengan arsitektur komputer saat ini. Dalam ilmu visualisasi komputer, algoritma ini merupakan salah satu algoritma awal yang dikembangkan.

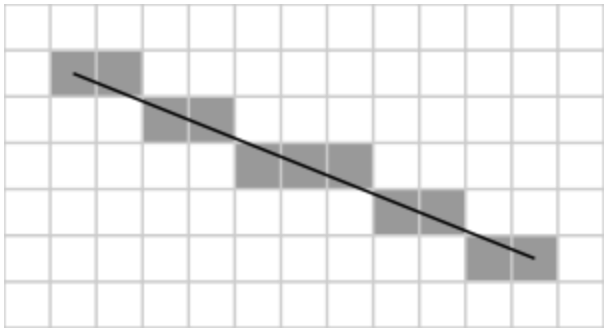
Sebuah garis pada layar yang dibentuk dengan algoritma garis bresenham tidak akan terputus, dalam artian semua pixel penyusun garis akan bertetangga dengan minimal dua pixel lain (kecuali titik ujung). Karena pixel berupa integer, dan layar merupakan bidang dua dimensi, maka layar bisa direpresentasikan sebagai sebuah sistem koordinat kartesian, dengan pixel sebagai titiknya. Algoritma garis bresenham dapat menentukan titik mana saja yang perlu

dilewati untuk membentuk garis lurus, agar garis yang terbentuk tidak putus.

Untuk memahami cara kerja algoritma garis bresenham secara umum, tinjau permasalahan berikut. Misal diberikan sebuah fungsi pembentuk garis, bilamana diketahui dua titik pembentuknya:

$$\frac{y - y_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}.$$

Apabila dua koordinat titik tersebut dimasukkan ke dalam rumusan diatas, maka akan didapat persamaan garis, yang memetakan nilai satu sumbu ke sumbu lain (umumnya memetakan x ke y, y = f(x)). Kemudian tinjau ilustrasi berikut ini:



Gambar 2.3 Penerapan Bresenham

Gambar diatas merupakan garis dengan titik ujung (1,1), dan (11,5). Titik (0,0) berada di pojok kiri atas. Rumusan sebelumnya digunakan untuk menentukan pemetaan titik pembentuk garis, bila diberikan sebuah nilai x, maka bisa diketahui nilai y yang berkesesuaian. Namun dalam kasus ini, terlihat bahwa apabila diberikan nilai x berupa integer, pemetaan ke garis tersebut akan menghasilkan nilai y yang real (posisi y tidak pas pada tengah kotak). Hal ini menjadi permasalahan karena pixel pada layar tidaklah real, melainkan integer. Oleh karena itu, dilakukan aproksimasi terhadap nilai y yang memungkinkan.

$$y = \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + y_0.$$

Persamaan tersebut didapat dari persamaan sebelumnya, berfungsi untuk menentukan nilai y. Nilai y yang didapat, kemudian akan dibulatkan ke nilai integer terdekat. Perlu diperhatikan juga bahwa gradien garis tersebut dibawah 1, dan diatas -1, maka dari itu untuk setiap nilai (kolom) x, hanya memiliki satu nilai y. Suksesor dari suatu x, akan memiliki nilai y yang sama atau lebih satu dari sebelumnya. Sementara untuk tiap baris (nilai y), diperbolehkan untuk memiliki berberapa nilai. Namun apabila misal gradien dari garis diatas 1 atau dibawah -1, aturan tersebut berlaku kebalikannya, untuk y dan x..

Dalam implementasi pembulatan nilai y, algoritma garis bresenham mencatat error value dari nilai pemetaan x ke y, yang nilainya antara -0.5 dan 0.5, yang merepresentasikan jarak vertikal antara nilai y sebenarnya, dan hasil pembulatan. Ketika memproses suksesor sebuah x, nilai error akan bertambah akibat gradien, apabila melebihi 0.5, maka nilai y bertambah satu dari nilai sebelumnya, dan error dikurangi 1.0. Berikut adalah pseudocode dari algoritma bresenham yang umum:

```
  function line(x0, x1, y0, y1)
       int deltax := x1 - x0
       int deltay := y1 - y0
       real error := 0
       real     deltaerr     :=     abs
(deltay/deltax)
       int y := y0
       for x from x0 to x1
           plot(x,y)
           error := error + deltaerr
           if error ≥ 0.5 then
               y := y + 1
               error := error - 1.0
```

Optimisasi terhadap pseudocode tersebut bisa dilakukan dengan mengeliminasi penggunaan data tipe real, karena menggunakan sumber daya yang relatif lebih banyak ketimbang integer. Maka dari itu, nilai-nilai pecahan seperti error dan deltaerr dimodifikasi, diubah menjadi integer, dengan mengalikannya dengan deltax.

```
  function line(x0, y0, x1, y1)
       boolean steep := abs(y1 - y0) >
abs(x1 - x0)
       if steep then
           swap(x0, y0)
           swap(x1, y1)
       if x0 > x1 then
           swap(x0, x1)
           swap(y0, y1)
       int deltax := x1 - x0
       int deltay := abs(y1 - y0)
       int error := deltax / 2
       int ystep
       int y := y0
       if y0 < y1 then ystep := 1 else
ystep := -1
       for x from x0 to x1
           if steep then plot(y,x) else
plot(x,y)
           error := error - deltay
           if error < 0 then
               y := y + ystep
               error := error + deltax
```

Pseudocode tersebut bisa disimplifikasi lebih lanjut, dengan cara eliminasi prosedur `swap` pada inisialisasi.

```
  function line(x0, y0, x1, y1)
     dx := abs(x1-x0)
     dy := abs(y1-y0)
     if x0 < x1 then sx := 1 else sx :=
```

```
-1
     if y0 < y1 then sy := 1 else sy :=
-1
     err := dx-dy

     loop
       setPixel(x0,y0)
       if x0 = x1 and y0 = y1 exit loop
       e2 := 2*err
       if e2 > -dy then
         err := err - dy
         x0 := x0 + sx
       end if
       if e2 <  dx then
         err := err + dx
         y0 := y0 + sy
       end if
     end loop
```

III. Rumusan Masalah

Dalam kriptografi, terdapat dua buah tahapan dalam melakukan cipher, yaitu substitusi dan transposisi. Cipher substitusi melakukan penggantian nilai karakter, dengan melakukan pemetaan satu/berberapa karakter lain menjadi sebuah/berberapa karakter yang berbeda. Sementara itu, cipher transposisi mengacak urutan nilai karakter salam suatu pesan. Banyak sekali metode untuk melakukan transposisi, namun pada umumnya kunci cipher transposisi hanyalah sebuah nilai angka, dimana angka tersebut biasa merepresentasikan panjang pembagian blok transposisi. Dengan kekuatan komputasi komputer zaman sekarang, kunci transposisi tersebut bisa dicari menggunakan metode brute force atau iterative search dengan cepat.

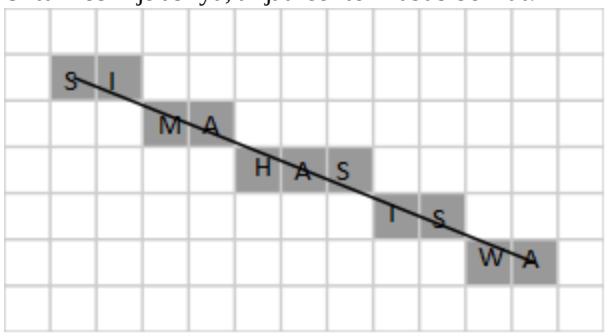
Diperlukan suatu alternatif dalam fungsi transposisi, untuk meningkatkan keamanan pesan. Gagasan penulis adalah, dibentuk sebuah fungsi baru yang memanfaatkan algoritma garis bresenham. Fungsi baru ini kemudian akan dianalisis kelebihan dan kekurangannya.

IV. Transposisi Pesan dengan Fungsi Bresenham

Desain penggunaan algoritma bresenham untuk transposisi pesan adalah sebagai berikut:

- Kunci untuk fungsi transposisi adalah dua buah titik, yang digunakan untuk membentuk persamaan garis.
- Salah satu titik yang dimasukkan tersebut, dijadikan sebagai titik pertama.
- Titik akan diibaratkan sebagai sebuah slot penampung karakter/bit, tergantung untuk kriptografi jenis apa transposisi ini digunakan (klasik/modern).
- Secara berurutan, masukkan karakter pesan, dimulau dari karakter pertama, ke titik/slot pertama, hingga seluruh pesan terpetakan pada garis.
- Tiap-tiap baris dari garis yang terbentuk, merupakan segmen-segmen pesan yang bisa kita manipulasi, dengan metode yang berbeda-beda.
- Untuk kali ini, manipulasi segmen dilakukan secara sederhana, yakni mengurutkan dari baris terakhir hingga baris pertama.

Untuk lebih jelasnya, tinjau contoh kasus berikut:



Gambar 4.1 Pemetaan Pesan dengan Bresenham

Misal diberikan sebuah plaintext: "SIMAHASISWA". Dan diberikan sebuah persamaan garis yang identik dengan persamaan garis pembentuk gambar 2.3. Tiap baris dianggap sebagai sebuah segmen, dan satu per setu, segmen tersebut disatukan dari baris terakhir, dimana hasil transposisi pesan akan menjadi: "WAISHASMASI". Untuk implementasi fungsi transposisi ini, dibuat struktur data sebagai berikut:

```
public class Point {
      private int x;
      private int y;
      private byte message;

      public Point(int x, int y, int
message) }
            this.x = x;
            this.y = y;
            this.message = message;
      }

      public void setX (int x) {
            this.x = x;
      }

      public void setY (int y) {
            this.y = y;
      }

      public void setMessage (byte
message) {
            this.message = message;
      }

      public int getX () {
            return x;
      }

      public int getY () {
            return y;
      }
```

```
	public byte getMessage() {
		return message;
	}
}
```

```
public class FieldEncrypt {
	private ArrayList<Point> points;
	private Point firstPoint;

	/* secondPoint.getX() >
firstPoint.getX() */
	public Field(String plaintext,
Point firstPoint, Point secondPoint) {
		this.firstPoint =
firstPoint;
		byte[] message =
plaintext.getBytes("UTF-8");
		int x = firstPoint.getX();
		int y = firstPoint.getY();

		if (getSlope() < 0) {
			for (int i = 0; i <=
message.length; i++) {
				points.add(new
Point(x, bresenhamX(firstPoint,
secondPoint, x), message[i]));
				x++
			}
		} else {
			for (int i = 0; i <=
message.length; i++) {
				points.add(new
Point(bresenhamY(firstPoint,
secondPoint, y), y, message[i]));
				y++
			}
		}
	}

	public String getCiphertext()
{};
	public float getSlope() {};
	public void bresenhamX() {};
	public void bresenhamY() {};
}
```

```
public class FieldDecrypt {
	private ArrayList<Point> points;
	private Point firstPoint;

	/* secondPoint.getX() >
firstPoint.getX() */
	public Field(String ciphertext,
Point firstPoint, Point secondPoint) {
		this.firstPoint =
firstPoint;
		byte[] message =
ciphertext.getBytes("UTF-8");
		int x = firstPoint.getX();
		int y = firstPoint.getY();
		if (getSlope() < 0) {
			for (int i = 0; i <=
message.length; i++) {
				points.add(new
Point(x, bresenhamX(firstPoint,
secondPoint, x), null));
				x++
			}
		} else {
			for (int i = 0; i <=
message.length; i++) {
				points.add(new
Point(bresenhamY(firstPoint,
secondPoint, y), y, null));
				y++
			}
		}

		int line = getLastLine();
		for (int i = 0; i <=
message.length; i++) {
			// isi segmen baris,
dimulai dari baris terakhir
			line++;
		}
	};

	public getPlaintext() {};
	public float getSlope() {};
	public void bresenhamX() {};
	public void bresenhamY() {};
	public int getLastLine() {};
}
```

Kelas FieldEncrypt dan FieldDecrypt merupakan representasi bidang tempat dimodelkannya garis penampung pesan. FieldEncrypt berfungsi untuk melakukan enkripsi, sedangkan FieldDecrypt berfungsi untuk melakukan dekripsi. Cara kerja dari FieldEncrypt sama dengan penjelasan sebelumnya, sementara untuk FieldDecrypt adalah sebagai berikut:

- Dibentuk sebuah garis sepanjang pesan ciphertext, dimulai dari firstPoint.
- Penempatan karakter pesan dilakukan dengan cara yang berbeda dari FieldEncrypt.
- Dimulai dari baris terakhir, segmen tersebut diisi satu per satu.
- Proses dilanjutkan dengan mengakses baris predesesornya, dan kemudian dilakukan pengisian segmen kembali.
- Seusai semua karakter pesan dipetakan, pesan plaintext bisa disusun.
- Dimulai dari karakter pada firstPoint, konkat semua karakter secara mengurut urutan Point pada points.

V. ANALISIS ALGORITMA

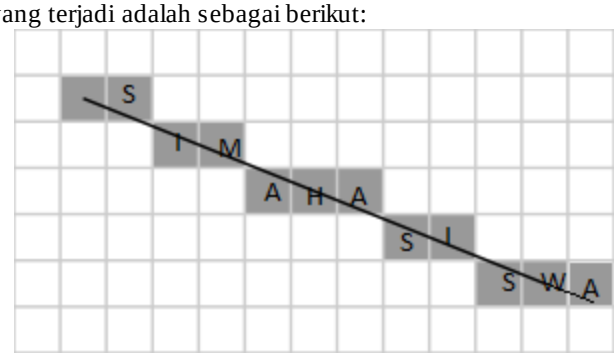
Secara teoritis, fungsi transposisi memiliki jumlah kunci yang tak hingga (nilai gradien garis yang dibentuk dari dua buah titik bisa berapapun nilainya, dari minus tak hingga s.d. tak hingga), sehingga menyulitkan kriptanalis untuk menyerang pesan dengan menggunakan brute force. Namun dalam implementasinya, gradien yang lebih dari satu tidak akan membingungkan penyerang. Hal ini dikarenakan hanya akan ada satu nilai x untuk setiap nilai y, yang menyebabkan panjang segmen hanya satu karakter.

Dengan metode pengaturan segmen yang saat ini digunakan (menggabungkan baris dari akhir ke awal), penggunaan gradien lebih dari satu hanya akan membalik urutan karakter pesan, dimulai dari akhir hingga awal, sangat mudah untuk diubah kembali menjadi plaintext. Selain gradien diatas satu, gradien dibawah satu juga akan memiliki dampak yang sama. Apabila gradien 0, maka hanya akan ada satu baris/segmen, sehingga fungsi transposisi tidak akan mengubah bentuk plaintext, sebagaimanapun rumitnya metode pengacakan segmen.

Maka dari itu, disimpulkan bahwa gradien yang ideal untuk fungsi transposisi ini adalah:

$$g = gradien$$
$$-\infty < g < 0$$
$$0 < g < \infty$$

Keamanan juga bisa ditingkatkan penggeseran penempatan karakter pertama dari posisi firstPoint, dimana fungsinya (yang seharusnya) adalah sebagai penanda tempat penyimpan karakter pertama. Walau dengan sebuah persamaan garis yang sama, apabila posisi awal penyimpanan karakter berbeda, maka segmentsi keseluruhan akan berbeda pula. Tinjau kembali gambar 4.1, misal penyimpanan karakter pertama dari "SIMAHASISWA" tidak dari (1,1), namun (2,1). Maka yang terjadi adalah sebagai berikut:



Gambar 5.1 Pengubahan Posisi Karakter Pertama

Ciphertext yang dihasilkan menjadi "SWASIAHAIMS", berbeda dengan yang sebelumnya, yakni "WAISHASMASI". Dengan pergeseran tersebut dilibatkan sebagai kunci transposisi, maka keamanan fungsi pesan akan semakin meningkat.

Mengingat bahwa kunci adalah dua buah titik, kunci juga bisa disampaikan secara implisit melalui sebuah kalimat seperti: "Bambang pergi dari Kota A ke Kota B dengan bus nomor 2". Titik pertama dan kedua pembentuk garis didapat dari koordinat Kota A dan Kota B pada peta dunia, dan penempatan karakter bergeser sebanyak nomor bus (dua).

Karena algoritma baru yang diajukan baru sebatas fungsi transposisi, ada baiknya untuk penggunaan nyata, digabungkan dengan fungsi substitusi, agar menjadi *super-encryption* yang sangat sulit untuk dipecahkan.

REFERENSI

[1] Rinaldi Munir. Presentasi Kuliah IF3058 Kriptografi
[2] Bresenham, J. E.. Algorithm for computer control of a digital plotter
[3] Abrash, Michael. Michael Abrash's graphics programming black book
[4] http://www.cs.helsinki.fi/group/goa/mallinnus/lines/bresenh.html
[5] http://lifc.univ-fcomte.fr/~dedu/projects/bresenham/index.html

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, 27 Maret 2013

Rubiano Adityas
13510041