Teknik Kriptografi dengan Menggunakan Algoritma Gabungan LCG-ROT

Ananta Pandu Wicaksana / 13510077
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
ananta.pandu@students.itb.ac.id

***Abstrak*— Makalah ini memiliki tema kriptografi. Makalah ini membahas algoritma enkripsi baru yang terbentuk dari 2 buah algoritma, yaitu: algoritma pembangkit bilangan acak *Linear Congruental Generator* (LCG) dan algoritma enkripsi ROT.**

***Kata Kunci*— Enkripsi, LCG, ROT.**

I. PENDAHULUAN

A. LATAR BELAKANG

Pada awalnya penulis bingung apakah yang akan dijadikan topik pada makalah kali ini, namun pada akhirnya penulis menjadikan kriptografi sebagai topik kali ini, karena menurut penulis, memberitahukan sebuah rahasia kepada seseorang tanpa diketahui orang lain merupakan hal yang memiliki keindahan tersendiri.

Selain itu penulis juga ingin mencoba membuat suatu program pengenkripsi pesan. Penulis mencoba menggabungkan dua algoritma yang berbeda untuk membuat sebuah program enkripsi, dengan harapan, kemungkinan pesan dapat dibaca oleh orang yang tak berhak adalah sangat kecil.

Penulis menemukan ide ini setelah membaca buku kuliahnya dan menemukan dua algoritma tersebut dan terpikir "Bagaimana kalau digabungkan saja?".

B. TUJUAN

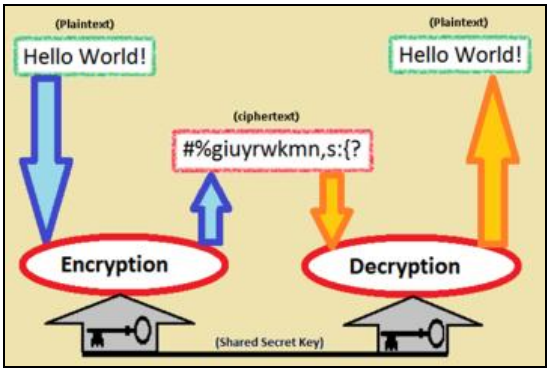
a. Menambah pemahaman tentang kriptografi beserta algoritma-algoritmanya.
b. Membuat suatu algoritma enkripsi yang lebih kompleks dan lebih susah dibongkar.
c. Membuat pengaplikasiannya dalam bentuk program.

II. DASAR TEORI

A. KRIPTOGRAFI

Kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan [SCH96]. Keamanan pesan diperoleh dengan mengubah pesan tersebut menjadi suatu bentuk yang tak bermakna agar tak diketahui orang lain, lalu mengubah kembali ke asal jika sampai di tujuan.

Pesan yang dirahasiakan dinamakan plainteks, sedangkan pesan hasil penyandian dinamakan chiperteks. Chiperteks dapat dikembalikan ke plainteks oleh orang yang berhak, biasanya orang tersebut mengetahui metode penyandian atau memiliki kunci (key) penyandian. Proses penyandian pesan disebut enkripsi dan proses pembalikan penyandiannya disebut dekripsi.



Setiap proses sandi-menyandi memiliki metode tersendiri, tidak jarang metode yang digunakan menggunakan kunci. Kunci (key) berfungsi untuk menghasilkan chiperteks yang unik (beda key beda chiperteks) dan juga mengembalikan chiperteks tersebut ke plainteks.

Gambar diatas menunjukkan metode enkripsi dengan key sama (simetri) sedang bila terdapat dua key berbeda: key1 untuk enkripsi dan key2 untuk dekripsi, metode yang digunakan menggunakan kunci asimetri.

B. Algoritma Enkripsi ROT

Salah satu algoritma enkripsi yang terkenal adalah *Caesar Chiper*, yaitu teknik kriptografi yang dahulu digunakan Julius Caesar untuk mengirim pesan kepada gubernurnya.

Pada *Caesar Chiper*, tiap huruf disubstitusi dengan tiga huruf berikutnya dalam susunan alphabet. Dalam hal ini, kunci (key) yang digunakan adalah jumlah pergesran huruf (yaitu 3). Berikut adalah tabel yang menggambarkan.

plainteks	A	B	C	D	E	F	G	H	I
chiperteks	D	E	F	G	H	I	J	K	L

Jadi A disubstitusi dengan D, B dengan E, dst. Dengan memisalkan setiap huruf dengan bilangan bulat, A=0, B=1, C=2, … , Z=25, maka dapat diperoleh rumus enkripsi :

$$C = E(p) = (p+3) \bmod 26$$

Demikian metode pengenkripsian *Caesar Chiper*. Rumus ini memiliki perode maskimal 25, artinya nilai yang dihasilkan adalah dari 0 sampai dengan 25.

Selanjutnya metode ini disebut ROT3, yaitu pengenkripsian dengan menggeser huruf sebanyak tiga. Ada pula ROT1 (geser 1) dan ROT13 (geser 13),. Selanjutnya dapat kita rumuskan fungsi ROT sendiri, yaitu:

$$ROT(m, x, a) = (x+a) \bmod m$$

Dengan m adalah modulus, x adalah bilangan yang akan dienkripsi, dan a sebagai key. Fungsi ROT ini memiliki periode nilai m-1.

C. Pembangkit Bilangan Acak *Linear Congruental Generator* (LCG)

Adakalanya kita membutuhkan bilangan acak dalam program yang kita buat. Contohnya bilangan acak kerap digunakan untuk simulasi kehidupan nyata pada program.

Tidak ada perhitungan yang benar-benar menghasilkan bilangan yang acak sempurna. Bilangan yang dihasilkan dengan rumus matematika adalah bilangan acak yang semu (pseudo), karena pembangkitannya dapat menghasilkan nilai yang sama untuk input yang sama. Pembangkit bilangan acak semacam itu disebut pembangkit bilangan acak semu (*pseudo-random number generator* atau *PRNG*).

Metode yang paling umum digunakan untuk membangkitkan bilangan acak adalah dengan pembangkit acak kongruen lanjar (*linear congruental generator* atau *LCG*) yang berbentuk:

$$x_n = (ax_{n-1} + b) \bmod m$$

dengan :
x_n = bilangan acak suku ke-n
x_{n-1} = bilangan acak suku sebelumnya
a = faktor pengali
b = penambah
m = modulus
(a,b, dan m konstanta)

Kunci pembangkit adalah x_0 (disebut umpan/seed).

LCG memiliki periode yang tak lebih besar dari modulusnya (m). LCG memiliki periode maksimal m-1.

Disini, penulis membuat fungsi LCG dengan lima input (m, a, b, x_0, dan n) dan dengan cara rekursif.

III. Algoritma Gabungan LCG-ROT

A. Penggabungan Kedua Algoritma

Pada algoritma ROT biasa, jumlah karakter yang digunakan adalah 26, yaitu jumlah huruf alfabet, namun pada algoritma ini, jumlah karakter adalah 73, yaitu: 26 huruf kecil, 26 huruf kapital, 9 angka, dan 12 karakter lain (spasi dan ganti baris tidak termasuk). Karena itu ada 73 macam pergeseran yang mungkin digunakan.

Karena dalam fungsi ROT yang digunakan terdapat m yang merupakan junlah karakter (73) dan x adalah karakter yang dienkripsi, maka a adalah yang menjadi pusat perhatian kita. Sebagai key, a inilah yang akan kita ubah sendiri untuk menghasilkan chiperteks.

Sedangkan dalam algoritma ini, fungsi LCG yang telah dirumuskan sebelumnya hanya akan membutuhkan 4 input yang bisa kita ubah, yaitu a, x_0, b, dan n (sedangkan m sudah pasti 73).

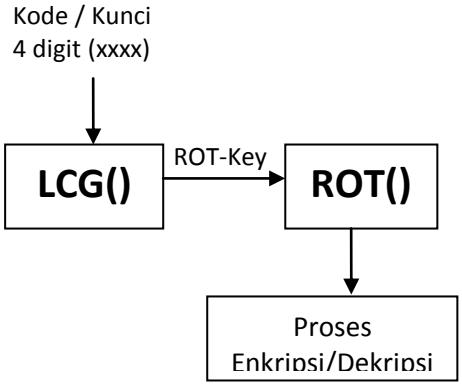
Lalu apakah yang dimaksud algoritma

gabungan? Yang dimaksud adalah, metode ini menggunakan kedua algoritma (LCG dan ROT) sekaligus, secara berurutan untuk mengenkripsi. Caranya adalah menghasilkan LCG dengan 4 input untuk menghasilkan sebuah bilangan acak, lalu bilangan acak inilah yang merupakan ROT-key, yaitu key yang menentukan pergeseran karakter.

Jadi, langkah-langkah lebih spesifiknya:

1. Membuat bilangan 4 digit sebagai key.
2. Mengonversi key ini dengan metode LCG menjadi suatu bilangan acak yang berikutnya disebut dengan ROT-key. Keempat digit bilangan akan dipecah jadi empat buah bilangan yang merupakan input dari fungsi LCG.
3. Menggunakan ROT-key untuk mengenkripsi dengan metode ROT.

Berikut ini adalah grafik yang menjelaskan.



B. Proses Enkripsi dan Dekripsi

Proses Enkripsi yang digunakan bukan memindai satu persatu karakter dalam kalimat lalu mengonversinya dengan metode gabungan, melainkan menggunakan tabel. Disini, karakter dipindai lalu dengan mencocokannya pada tabel dapat diketahui pasangan dari karakter tersebut yang merupakan chiperteksnya. Jadi algoritma gabungan LCG-ROT digunakan untuk membuat, tabel tersebut.

Berikut ini adalah contoh tabel, Karakter asli terdapat pada baris char, karakter chiper terdapat pada baris chir.

Indeks	0	..	25	26	..	34	35	..	60
Char	a	..	z	1	..	9	A	..	Z
Chir	c	..	2	3	..	B	C	..	@

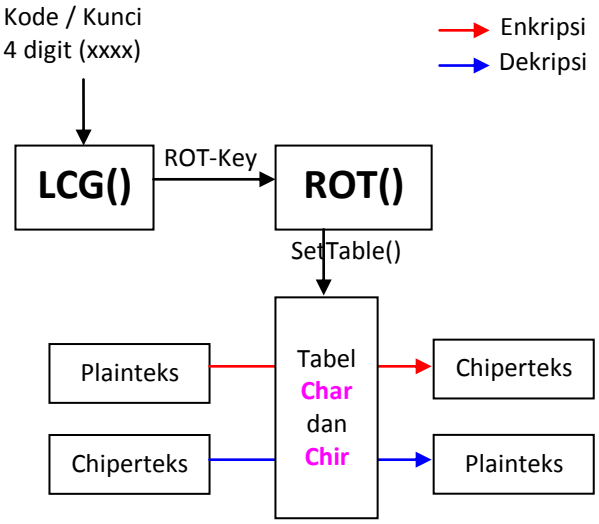
Contoh tabel dengan ROT3

Tabel diatas baru dibentuk ketika fungsi LCG-ROT dijalankan. Baris karakter chir dibuat dengan fungsi LCG-ROT.

Ketika mengenkripsi, karakter chiper dihasilkan dengan mencocokkan dengan menyusuri baris char, setelah ketemu, karakterchiper diketahui yaitu merupakan pasangannya (tabel chir).

Bagaimana dengan proses dekripsi? Proses dekripsi dilakukan dengan kebalikannya, yaitu menelusuri baris chir lalu menemukann pasangan yang merupakan baris char, sehingga tidak perlu membuat fungsi invers dari fungsi LCG-ROT.

Berikut adalah grafik yang menjelaskan.



C. Rumus dan Kode

➢ **Fungsi-fungsi :**

- Fungsi ROT :

  ROT (m,x,a) = (x+a) mod m

- Fungsi LCG :

  LCG (m,a,b,x_0,n) =
  n=0, x_0
  n>0, (a.LCG(m,a,b,x_0,n-1) +b) mod m

➢ **Kode-kode:**

- Kode ROT (JavaScript) :

```
function ROT(m,x,a) {
        return (x + a) % m;
}
```

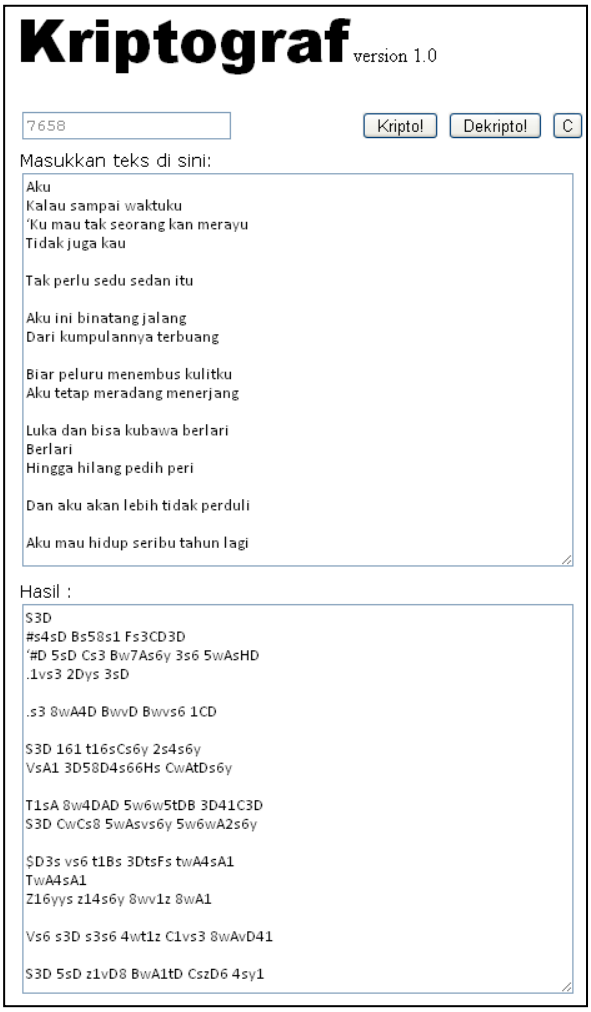
- Kode LCG (JavaScript) :

```
function LCG(m, a, b, xo, n) {
  if (n==0)
    return xo;
  else
    return (a*(LCG(m,a,b,xo,n-1))+b)%m;
}
```

D. Contoh Program

Program dibuat dengan HTML dan JavaScript.

Kode : 5345
Pesan : Saya sedang makan nasi
Hasil : eyNy H32yC5 By9yC CyH7



Kode : 7658
Pesan : "Aku" karya Chairil Anwar
Hasil : <spt yang terlihat>

IV. ANALISIS

A. Kekuatan

- Algoritma ini memiliki key yang bervariasi, yaitu dari 0000 s/d 9999.
- Metode dengan tipe kunci simetri.

B. Kelemahan

- Rot-Key yang dihasilkan hanya 73 macam, sesuai jumlah elemen pada tabel. Jadi hanya ada 73 macam perubahan.
- Rot-Key bisa saja 0, artinya chiperteks sama persis dengan plainteks. Contohnya saat kode yang dimasukkan 0000.

V. KESIMPULAN

Dengan algoritma LCG-ROT, proses enkripsi menjadi lebih susah diketahui daripada algoritma ROT biasa, karena kunci yang digunakan adalah 4 digit. Pada lain hal algoritma ini juga memiliki kelemahan, yaitu seperti metode ROT biasa, hanya ada 73 kemungkinan bentuk chiperteks, selain itu untuk kunci tertentu, chiperteks yang dihasilkan sama persis dengan plainteks.

REFERENSI

[1] Rinaldi Munir, "Matematika Diskrit", edisi ketiga. Bandung: Informatika, 2009.
[2] en.wikipedia.org/wiki/cryptograph, diakses 11-12-2011 23:00.
[3] en.wikipedia.org/wiki/Linear_congruential_generator diakses 12-12-2011 09:38
[4] en.wikipedia.org/wiki/ROT13 diakses 12-12-2011 09:30

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, 12 Desember 2012

Ananta Pandu Wicaksana
13510077

LAMPIRAN

Terlampir program dalam file kripto.zip (email), terdiri dari:

- Index.html
- Kripto.js
- Lain.js
- Style.js

File yang merupakan komponen utama adalah index.html dan kripto.js

Index. html :

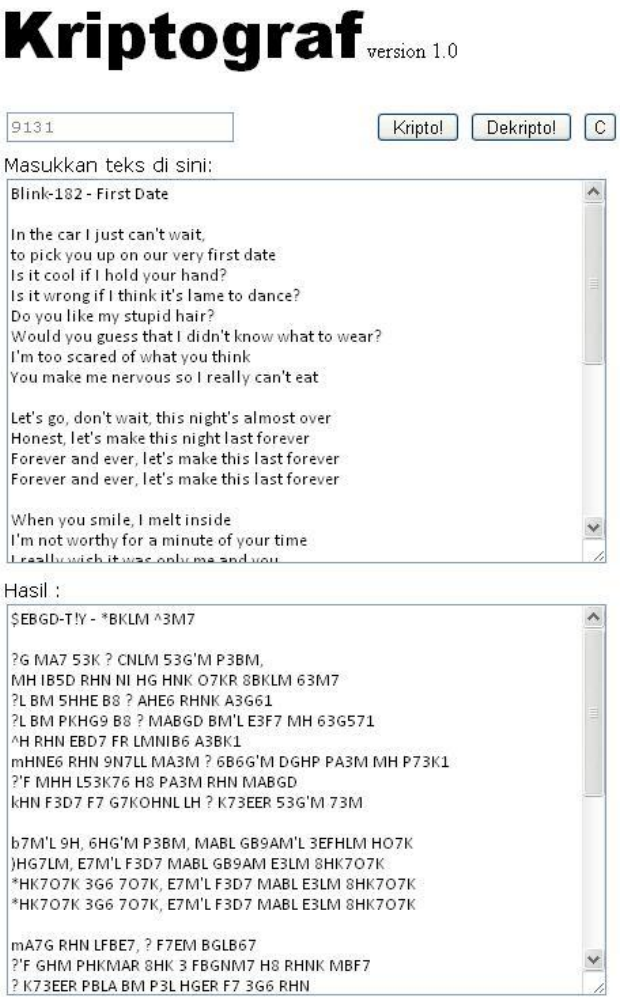
```
<html>
<head>
	<title>Kripto</title>
	<script language="javascript" src="kripto.js"></script>
	<script language="javascript" src="lain.js"></script>
	<link rel="stylesheet" href="style.css">
</head>

<body>


<form name="f">

<font size="60pt" face="arial black">Kriptograf</font>
<font>version 1.0</font>
<br><br>

<table>

	<tr>
		<td>
			<input type="text" id="kode" name="kode" value="input kode"
onMouseOver="kh_on()" onMouseOut="kh_off()">
		</td>
		<td align="right">
			<input type="button" value=" Kripto! "  onClick="kriptopressed()">
			<input type="button" value=" Dekripto! "  onClick="dekriptopressed()">
			<input type="button" value="C"  onClick="document.f.teksawal.value = '';
p.clear();">
		</td>
	<tr>
	<tr>
		<td colspan="2">
			<font class="tulisan1">Masukkan teks di sini:</font><br>
			<textarea class="TA" name="teksawal" cols="60" rows="10"></textarea>
			 
		</td>
	</tr>
	<tr>
		<td colspan="2">
			<font class="tulisan1">Hasil :</font><br>
			<textarea class="TA" name="teksakhir" cols="60" rows="10"
readonly></textarea>
		</td>
	</tr>

</table>


</form>



</body>
</html>
```

Kripto.js :

```
var algrtm = {
	rot : function (batas,n,np) {
		return (n+np)%batas;
	},
	LCG : function (m, a, b, xo, n) {
		if (n==0)
			return xo;
		else
			return (a*(this.LCG(m,a,b,xo,n-
1))+b)%m;
	},
};

var dbkata = {
	neff : 0,
	char : new Array(),
	chir : new Array(),
	charsearch : function (chr) {
		var i;
		for (i=0; i<=this.neff-1; i++){
			if (this.char[i]==chr)
				return i;
		}
		return -1;
	},
	chirsearch : function (chr) {
		var i;
		for (i=0; i<=this.neff-1; i++){
			if (this.chir[i]==chr)
				return i;
		}
		return -1;
	},
	setneff : function() {
		this.neff = this.char.length;
	},
	setchar : function() {
		//--- huruf kecil
		this.char[0]="a"; this.char[1]="b";
this.char[2]="c";
		this.char[3]="d"; this.char[4]="e";
this.char[5]="f";
		this.char[6]="g"; this.char[7]="h";
this.char[8]="i";
		this.char[9]="j"; this.char[10]="k";
this.char[11]="l";
		this.char[12]="m"; this.char[13]="n";
this.char[14]="o";
		this.char[15]="p"; this.char[16]="q";
this.char[17]="r";
		this.char[18]="s"; this.char[19]="t";
this.char[20]="u";
		this.char[21]="v"; this.char[22]="w";
this.char[23]="x";
		this.char[24]="y"; this.char[25]="z";
		//--- angka
		this.char[26]="1"; this.char[27]="2";
this.char[28]="3";
		this.char[29]="4"; this.char[30]="5";
this.char[31]="6";
		this.char[32]="7"; this.char[33]="8";
this.char[34]="9";
		//--- huruf kapital
		this.char[35]="A"; this.char[36]="B";
this.char[37]="C";
		this.char[38]="D"; this.char[39]="E";
this.char[40]="F";
		this.char[41]="G"; this.char[42]="H";
this.char[43]="I";
		this.char[44]="J"; this.char[45]="K";
this.char[46]="L";
		this.char[47]="M"; this.char[48]="N";
this.char[49]="O";
		this.char[50]="P"; this.char[51]="Q";
this.char[52]="R";
		this.char[53]="S"; this.char[54]="T";
this.char[55]="Y";
		this.char[56]="V"; this.char[57]="W";
this.char[58]="X";
		this.char[59]="Y"; this.char[60]="Z";
		//--- KARAKTER LAIN
		this.char[61]="!"; this.char[62]="@";
this.char[63]="#";
		this.char[64]="$"; this.char[65]="%";
this.char[66]="^";
		this.char[67]="&"; this.char[68]="*";
this.char[69]="(";
		this.char[70]=")"; this.char[71]="?";
this.char[72]=".";
	},
	setchir : function(a,b,xo,n) {
		var i;
		for (i=0; i<=this.neff-1; i++){
			LCGKey =
algrtm.LCG(this.neff,a,b,xo,n);
			idxrotted = algrtm.rot(this.neff,
i, LCGKey);
			this.chir[i] =
this.char[idxrotted];
		}
	},
	tulisAll : function() {
		var i;
		for (i=0; i<=this.neff-1; i++){
			p.tulisln(this.char[i]+" ->
"+this.chir[i]);
		}
	},
	inisiasi : function () {
		var k = new Array();
		getkode(k);
		dbkata.setchar();
		dbkata.setneff();
		dbkata.setchir(k[1],k[2],k[3],k[4]);
	},
};



var p = {
	target : "teksakhir",
	tulis : function(chr) {document.f.teksakhir.value +=
chr;},
	ln : function() {document.f.teksakhir.value += '\n';},
	tulisln : function(chr) {
		this.tulis(chr);
		this.ln();
	},
	clear : function() {document.f.teksakhir.value = "";},
};

function getkode(x) {
	var kode = document.f.kode.value;
	x[1] = eval(kode.charAt(0));
	x[2] = eval(kode.charAt(1));
	x[3] = eval(kode.charAt(2));
	x[4] = eval(kode.charAt(3));
}


function kripto() {
	//----- Definisi -----
	var tawal = document.f.teksawal.value;
	var takhir = new Array();
	//----- Inisiasi -----
	dbkata.inisiasi();
	for(i=0; i< tawal.length; i++)
		takhir[i] = tawal.charAt(i);
	for (i=tawal.length; i<2000; i++)
		takhir[i] = '';
	//----- Pengubahan -----
	for(i=0; i< tawal.length; i++) {
		if (dbkata.charsearch(takhir[i])!=(-1))
			takhir[i] =
dbkata.chir[dbkata.charsearch(takhir[i])];
	}
	//----- Pencetakan -----
	p.clear();
	document.f.teksakhir.value = takhir.join('');
}

function dekripto() {
	//----- Definisi -----
	var tawal = document.f.teksawal.value;
	var takhir = new Array();
	//----- Inisiasi -----
	dbkata.inisiasi();
	for(i=0; i< tawal.length; i++)
		takhir[i] = tawal.charAt(i);
	for (i=tawal.length; i<2000; i++)
		takhir[i] = '';
	//----- Pengubahan -----
	for(i=0; i< tawal.length; i++) {
		if (dbkata.chirsearch(takhir[i])!=(-1))
			takhir[i] =
dbkata.char[dbkata.chirsearch(takhir[i])];
	}
	//----- Pencetakan -----
	p.clear();
	document.f.teksakhir.value = takhir.join('');
}

function kriptopressed() {
	kripto();
}

function dekriptopressed() {
	dekripto();
}
```

TAMPILAN PROGRAM

Program kosong



Program dengan kunci, masukan dan keluaran