Cipher Subtitusi Abjad Tunggal dengan Penyamarataan Frekuensi Hasil Enkripsi

Tugas Makalah I – IF 3058 Kriptografi, Semester II Tahun 2009 / 2010

Kevin Tirtawinata – 135 07 097

Teknik Informatika ITB

Email : kevin.tirtawinata@gmail.com

Abstract *– Makalah ini akan membahas mengenai salah satu algoritma kriptografi klasik yang tergolong pencil dan paper cipher, yaitu Cipher Subtitusi Abjad Tunggal. Cipher subtitusi abjad tunggal termasuk dalam kriptografi yang sederhana dan mudah dipahami. Dalam makalah ini akan dibahas mengenai sejarah ,metode dasar , teknik pengolahan abjad , penggunaan cipher subtitusi abjad tunggal dan bagaimana analisi frekuensi dapat memecahakan cipher subtitusi abjad tunggal ini. Di dalam makalah ini juga akan dibahas mengenai salah satu teknik yang mungkin dapat meningkatkan keamanan dari cipher subtitusi abjad tunggal dengan melindungi teknik kriptografi ini dari serangan analisi frekuensi.*

1.PENDAHULUAN

Sudah sejak lama informasi menjadi salah satu aspek penting dalam kehidupan ini. Walaupun teknik komunikasi berkembang, namun tidak ada salah satu alat komunikasi jarak jauh yang aman untuk mengirimkan suatu informasi yang pentin atau dirahasiakan. Segala pihak dari berbagai lingkup kehidupan sangat membutuhkan keamanan informasi, di saat informasi tersebut adalah informasi yang menjadi sebuah rahasia, yang tidak boleh diketahui oleh pihak lain. Untuk mengamankan informasi yang sifatnya rahasia, dikembangkan salah satu cabang ilmu pengetahuan yang bernama kriptografi.

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya, begitulah dahulu kriptografi muncul pada saat informasi yang perlu dijaga hanyalah sebatas pesan. Seiiring berkembangnya teknologi informasi kripografi berkembang sehingga ia tidak lagi sebatas mengenkripsi pesan tetapi juga memberikan aspek keamanan untuk berbagai jenis informasi yang ada.

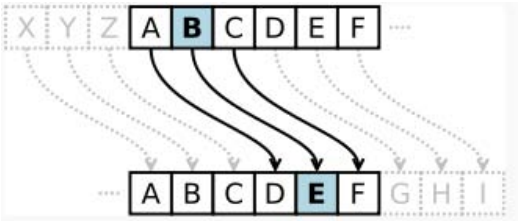
Di sisi lain, dikembangkan juga salah satu ilmu lain, yaitu kriptanalisis, yang merupakan ilmu dan seni untuk memecahkan cipherteks menjadi plainteks tanpa mengetahui kunci yang diberikan, pelaku dari kriptanalisis, disebut kriptanalis. Dengan adanya kriptanalisis, maka kebutuhan akan amannya pesan meningkat dan kriptografi pun harus menggunakan algoritma-algoritma baru yang tidak mudah dipecahkan.

2. LANDASAN TEORI

2.1 Cipher Subtitusi

Algoritma kriptografi subtitusi atau cipher subtitusi adalah algoritma yang tergolong dalam pencil and paper cipher, dan tidak memutuhkan komputer dalam penerapannya, karena proses enkripsinya sangat sederhana, tidak seperti algoritma enkripsi modern. Penerapan cipher subtitusi ini hanya memerlukan kertas dan pinsil saja dalam pengunaannya. Dalam penggunaannya, ciphere subtitusi memiliki banyak jenis penerapan dalam mengenkripsi pesan, yaitu cipher subtitusi abjad tunggal, cipher subtitusi homofonik, cipher abjad – majemuk dan cipher subtitusi poligram.

Algoritma kriptografi yang bernama caesar cipher, algoritma kriptografi yang diguakan oleh salah seorang kaisar romawi yang bernama Julius Caesar untuk menyandikan pesan yang dikirim kepada gubenur-gubenurnya. Caesar cipher Penggunaan caesar cipher adalah menggeser abjad yang ada dengan abjad lain yang letaknya berbeda sesuai dengan jumlah pergeseran yang digunakan. Caesar cipher adalah cipher yang paling sederhana yang pernah digunakan. Pada jaman romawi kuno, banyak pihak-pihak lain yang menggunakan pengenkripsian pesan selain Julius Caesar. Dan caesar cipher juga ikut berkembang agar dapat tetap digunakan untuk mengamankan pesan.



Cara pemecahan algoritma ini cukup dengan mencoba menggeser abjad-abjad yang ada sebanyak 26 kali sehingga dapat menemukan plainteks yang diinginkan.

0	exxegoexsrgi
1	dwwdfndwrqfh
2	cvvcemcvqpeg
3	buubdlbupodf
4	attackatonce
5	zsszbjzsnmbd
6	yrryaiyrmlac
...	
23	haahjrhavujl
24	gzzgiqgzutik
25	fyyfhpfytshj

Algortima lain yang terkanal adalah Atbash cipher yang digunakan oleh bangsa Yahudi Ibrani untuk menyembunyikan mistisme yang ada. Algoritma ini digunakan dengan cara menukar huruf pertama dengan huruf pertama terakhir ,huruf kedua dengan huruf kedua terkakhir, dan begitu seterusnya, sehingga subtitusi yang ada

```
First 13 letters: A|B|C|D|E|F|G|H|I|J|K|L|M
Last 13 Letters:  Z|Y|X|W|V|U|T|S|R|Q|P|O|N
```

Beberapa kata dalam bahasa inggirs yang kembali menjadi kata lain dalam bahasa inggris dengan menggunakan cipher atbash "hob"="sly", "hold"="slow", "holy"="slob", "horn"="slim", "zoo"="all", "irk"="rip", "low"="old", "glow"="told", and "grog"="tilt".

Cipher subtitusi homofonik adalah cipher subtitusi yang mensubtitusikan suatu huruf dengan kumpulan huruf lain dengan suatu huruf dapat menjadi beberapa kumpulan huruf. Cipher subtitusi lain adalah cipher subtitusi polialphabetic, dimana beberapa huruf diganti dengan huruf lain dengan jumlah yang sama.

Beberapa cipher subtitusi lain adalah viginere cipher dan juga one time pad.

Selain one time pad , yang tidak dapat dikriptanalisis jika pembangkitan kuncinya benar-benar diacak, cipher subtitusi lain dengan mudah dapat dipecahkan dengan analisis frekuensi. Sedangkan one time pad tidak dapat digunakan karena butuh biaya yang relatif mahal untuk mengirimkan kunci yang ada.

2.2 Cipher Subtitusi Abjad Tunggal

Cipher subtitusi abjad tunggal yang dimaksudkan sebagai inti pembahasan dalam makalah ini adalah cipher subtitusi dengan sebuah susunan kunci yang sudah diacak terlebih dahulu, ataupun merupakan pembangkitan dari suatu kunci yang mudah diingat. Tidak seperti caesar cipher ataupun atbash yang memiliki suatu aturan tertentu, cipher subtitusi abjad tunggal bersifat bebas. Yang dimaksudkan dengan susunan kata yang merupakan pembangkitan sebuah kunci adalah seperti di bawah ini :

1. Memilih sebuah kunci yang mudah untuk diingat, contoh “we hope you to enjoy this book”
2. Dibuang pengulangan huruf sehingga menjadi “wehopyunjtisbk”
3. Lalu ditambahkan huruf yang tidak ada secara berurutan menjadi “wehopyunjtisbkacdfglmqrvxz” , sebuah susudan abjad yang unik dan tidak beraturan dan dapat digunakan sebagai susunan kunci.

Susunan kunci yang terbentuk adalah sebagai berikut :

Pi:	A	B	C	D	E	F	G	H	I	J	K	L	M
Ci:	W	E	H	O	P	Y	U	N	J	T	I	S	B

Pi:	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ci:	K	A	C	D	F	G	L	M	Q	R	V	X	Z

Cipher ini juga termasuk cipher yang dapat diketahuin plainteksnya dengan sangat mudah dengan teknik analisis frekuensi.

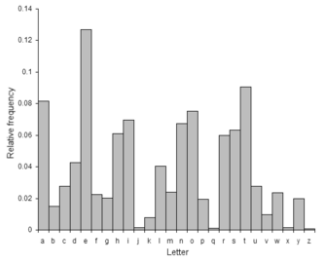
2.3 Teknik Analisis Frekuensi

Ketahanan cipher subtitusi terhadap serangan yang dilakukan oleh kriptanalis semakin rentan bila teks yang ada semakin panjang. Hal ini dikarenakan frekuensi kemunculan huruf pada cipherteks akan menujukan frekuensi dari kemunculan huruf yang bersesuaian dengan huruf cipher tersebut.

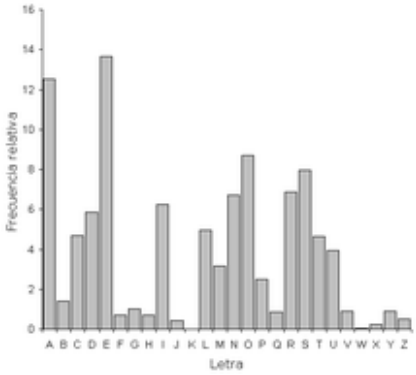
Teknik analisis frekuensi ditemukan pertama kali oleh Al-Kindi, seorang polymath arab di dalam buku *A Manuscript on Deciphering*

Cryptographic Messages. Pada tahun 1474 Cicco Simonetta menulis manual untuk mendekripsi dan enkripsi bahasa latin dan juga italia.

Frekuensi kemunculan huruf pada suatu bahasa tertentu dapat diketahui dengan mengambil sample mencapai 300.000 karakter di dalam sejumlah novel dan juga surat kabar, contoh dari persebaran huruf beberapa bahasa :



Persebarang huruf dalam bahasa Inggris



Persebaran huruf pada bahasa spanyol

Huruf / karakter yang memiliki frekuensi tinggi pada cipherteks akan menunjukkan bahwa huruf itu adalah huruf yang bersesuaian dengan huruf yang sering muncul pada suatu bahasa tertentu. Untuk kepastian susunan huruf yang digunakan, diperlukan juga trial and error, pengetahuan tentang bahasa, konteks dari plainteks dan juga intuisi dari kriptanalais.

Dengan munculnya sebuah teknik yang dapa memecahkan suatu cipher, maka muncul juga sebuah tantangan kepada semua ahli kriptografi untuk menciptakan teknik kriptografi yang tidak dapat dipecahkan.

3.HASIL DAN PEMBAHASAN

3.1 Kelebihan dan Kekurangan Cipher Subtitusi Abjad Tunggal

Kelebihan subtitusi abjad tunggal adalah panjang kunci yang relatif pendek, 26 huruf dan juga dapat merupakan pembangkitan susunan dari suatu kata / kalimat. Subtitusi abjad tunggal juga mudah digunakan dengan menggunakan pensil dan pulpen. Cipher subtitusi cukup kuat untuk ukuran teks yang relatif kecil, namun sangat rentan untuk ukuran teks yang relatif besar.

Kekurangan dari cipher subtitusi abjad tunggal adalah sangat mudah dipecahkan dengan metode analisis frekuensi untuk jumlah karakter yang relatif besar.

3.2 Pengembangan Cipher Subtitusi Abjad Tunggal

Dalam makalah ini, penulis ingin menjelaskan mengenai penambahan pergantian subtitusi pada cipher subtitusi abjad tunggal untuk meningkatkan keamanan pada cipher subtitusi abjad tunggal, terutama untuk menggagalkan teknik analisis frekuensi untuk cipher subtitusi ini. Agar hal itu dapat dilakukan maka hasil enkripsi yang ada lebih baik memiliki frekuensi setiap hurufnya relatif sama sehingga pada saat pengecekan frekuensi persebaran huruf-huruf pada plainteks dapat disamarkan dan tidak dapat diketahui oleh kriptanalis.

Contoh : Plainteks memiliki frekuensi sebagai berikut

a = 250 b = 68 c = 133 d = 96 e = 430 f = 72
g = 56 h = 159 i = 232 j = 0 k = 14 l = 158
m = 90 n = 233 o = 239 p = 99 q = 14 r = 223
s = 241 t = 316 u = 89 v = 28 w = 45 x = 19
y = 86 z = 3

Lalu dengan cipher subtitusi abjad tungga biasa dengan kunci “wehopyunjtisbkacdfglmqrvxz”, hasil enkripsinya akan menjadi

a = 239 b = 90 c = 99 d = 14 e = 68 f = 223
g = 241 h = 133 i = 14 j = 232 k = 233 l = 316
m = 89 n = 159 o = 96 p = 430 q = 28 r = 45
s = 158 t = 0 u = 56 v = 19 w = 250 x = 86
y = 72 z = 3

dan dengan cepat kriptanalis dapat mengetahi bahwa p dengan frekuensi terbesar merupakan abjad dengan frekuensi terbesar dan kunci langsung dapat ditebak dengan mudah.

3.3 Ide Dasar Pengembangan

Untuk dapat mencegah terbongkarnya pesan dengan analisis hasil ferkuensi, maka hasil enkripsi sebisa mugkin memiliki frekuensi yang rata di setiap hurufnya, ataupun memiliki persebaran yang tidak terlalu berbeda.

Untuk membuat hal itu tetap konsisten dan pesan dapat didekripsi, perubahan yang terjadi pada proses enkripsi harus memiliki standar tertentu sehingga dalam proses enkripsi dan dekripsi tidak terjadi kesalahan.

Kunci yang dipertukarkan posisinya pun harus tetap dapat menjaga keunikan dari kunci, dan

pada hasil dekripsi, perubahan kunci juga harus berdasar teks hasil enkripsi.

4. CIPHER SUBTITUSI ABJAD TUNGGAL DENGAN PENYAMARATAAN HASIL FREKUENSI (KIEVN CIPHER)

Dimakan kievn cipher agar mempermudah penulisan dan pembacaan.

4.1 Penggunaan Kievn Cipher

Cipher ini digunakan sama seperti dengan menggunakan cipher subtitusi abjad tunggal, cipher ini memerlukan masukan 26 buah kunci yang unik, tidak memiliki perulangan suatu huruf. Cipher ini juga memerlukan masukan sebuah angka yang digunakan sebagai batas untuk pergantian kunci yang ada. Untuk memudahkan ujicoba pada program java yang sudah saya buat terdapat sebuah checkbox yang mengenerate kunci otomatis yaitu kunci yang ada di bawah ini.

Pi:	A	B	C	D	E	F	G	H	I	J	K	L	M
Ci:	W	E	H	O	P	Y	U	N	J	T	I	S	B
Pi:	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ci:	K	A	C	D	F	G	L	M	Q	R	V	X	Z

Untuk menggunakan cipher ini juga dibutuhkan berapa batasan untuk pergantian perpindahan kunci dalam mevariasikan penggunaa algoritma ini.

4.2 Kievn Cipher

Setiap huruf dari cipherteks yang merupakan hasil enkripsi dihitung kemunculannya terlebih dahulu, dan bila huruf tersebut sudah muncul melebihi batas yang ada, sistem akan menukarkan huruf kunci tersebut tersebut dengan huruf lain yang minimum. Sehingga untuk enkripsi berikutnya, kemunculan huruf tersebut bukan merupakan suatu huruf yang sama. Dalam hal tersebut bisa kita perlihatkan dalam contoh kasus dengan menggunakan kunci otomatis yang tersedia.

Plainteks :

eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee

Dengan cipher subtitusi abjad tunggal biasa :

Pppppppppppppppppppppppppppppppppppppp
ppppppppppppppppppppppppppppppppppppppp
ppppppppp

Dengan Kievn Cipher n = 5 :

Ppppaaaabbbbccccddddeeeeffffgggghhhhiiiijjjj
kkkkllllmmmmnnnnooooqqqqrrrrssssttttuuuuv

Dengan Kievn Cipher n = 7 :

ppppppaaaaaabbbbbbccccccddddddeeeeeeffffff
gggggghhhhhhiiiiiijjjjjjkkkkkkllllllmmmmmmn

Proses yang membedakan kievn cipher n=x pada hal ini adalah :

Setelah suatu huruf dienkripsi menjadi sebuah huruf yang memiliki kemunculan enkripsi kx-1, maka huruf tersebut akan ditukar kuncinya dengan huruf yang memiliki huruf hasil enkripsi dengan jumlah minimum yang pertama kali ditemukan. Dalam hal ini, karena minimum dari semua enkripsi adalah 0 selain hasil enkripsi dari huruf e, maka hasil enkripsi huruf e akan berubah dari a, menjadi b, menjadi c, dan seterusnya.

Perubahan dari kunci dilakukan setiap kondisi dari huruf keluaran hasil enkripsi sesuai dengan batasan yang diberikan oleh user.

4.3 Hasil Penerapan Kievn Cipher

Pembuktian penyamarataan frekuensi

Contoh untuk plainteks dengan ukuran sangat besar :

Plainteks

a = 6082	b = 944	c = 2517	d = 2210	e = 8375	f = 1421
g = 1319	h = 3184	i = 4961	j = 147	k = 393	l = 3082
m = 1941	n = 4524	o = 5055	p = 1755	q = 83	r = 4802
s = 4565	t = 6489	u = 1792	v = 815	w = 875	x = 222
y = 1042	z = 82				

Cipher subtitusi biasa

a = 5055	b = 1941	c = 1755	d = 83	e = 944	f = 4802
g = 4565	h = 2517	i = 393	j = 4961	k = 4524	l = 6489
m = 1792	n = 3184	o = 2210	p = 8375	q = 815	r = 875
s = 3082	t = 147	u = 1319	v = 222	w = 6082	x = 1042
y = 1421	z = 82				

Kievn Cipher n = 5

a = 2639	b = 2625	c = 2627	d = 2631	e = 2624	f = 2659
g = 2662	h = 2625	i = 2656	j = 2692	k = 2685	l = 2624
m = 2633	n = 2639	o = 2634	p = 2626	q = 2641	r = 2682
s = 2660	t = 2622	u = 2624	v = 2632	w = 2621	x = 2661
y = 2632	z = 2621				

Kievn Cipher n = 7

a = 2680	b = 2713	c = 2638	d = 2646	e = 2637	f = 2628
g = 2638	h = 2626	i = 2625	j = 2641	k = 2645	l = 2628
m = 2628	n = 2639	o = 2624	p = 2629	q = 2634	r = 2657
s = 2646	t = 2625	u = 2644	v = 2625	w = 2631	x = 2693
y = 2625	z = 2632				

Kievn Cipher n = 19

a = 2633	b = 2644	c = 2628	d = 2638	e = 2640	f = 2703
g = 2642	h = 2635	i = 2648	j = 2634	k = 2629	l = 2640
m = 2640	n = 2650	o = 2630	p = 2660	q = 2624	r = 2625
s = 2638	t = 2650	u = 2651	v = 2659	w = 2642	x = 2636
y = 2631	z = 2627				

Kievn Cipher dengan n = 400

a = 2504	b = 3039	c = 2578	d = 2858	e = 3694	f = 2492
g = 2981	h = 2810	i = 2520	j = 2469	k = 2437	l = 2448
m = 2452	n = 2799	o = 2415	p = 2568	q = 2426	r = 2557
s = 2659	t = 2798	u = 2833	v = 2446	w = 2462	x = 2517
y = 2449	z = 2466				

Contoh untuk plainteks relatif kecil :

Plainteks

a = 19 b = 2 c = 0 d = 6 e = 35 f = 7
g = 3 h = 15 i = 14 j = 0 k = 0 l = 13
m = 8 n = 12 o = 14 p = 4 q = 0 r = 20
s = 26 t = 35 u = 3 v = 1 w = 1 x = 0
y = 3 z = 0

Cipher subtitusi biasa

a = 14 b = 8 c = 4 d = 0 e = 2 f = 20
g = 26 h = 0 i = 0 j = 14 k = 12 l = 35
m = 3 n = 15 o = 6 p = 35 q = 1 r = 1
s = 13 t = 0 u = 3 v = 0 w = 19 x = 3
y = 7 z = 0

Kievn Cipher n = 19

a = 13 b = 14 c = 4 d = 20 e = 13 f = 8
g = 15 h = 6 i = 15 j = 6 k = 7 l = 15
m = 6 n = 6 o = 7 p = 13 q = 6 r = 6
s = 6 t = 7 u = 3 v = 6 w = 21 x = 5
y = 6 z = 7

a = 14 b = 8 c = 4 d = 17 e = 17 f = 18
g = 18 h = 8 i = 1 j = 14 k = 12 l = 18
m = 3 n = 15 o = 6 p = 20 q = 1 r = 1
s = 13 t = 0 u = 3 v = 0 w = 20 x = 3
y = 7 z = 0

Kievn Cipher n = 7

Kievn Cipher n = 5

a = 4 b = 13 c = 5 d = 9 e = 16 f = 15
g = 27 h = 11 i = 5 j = 4 k = 14 l = 19
m = 3 n = 15 o = 4 p = 12 q = 4 r = 4
s = 12 t = 11 u = 3 v = 14 w = 4 x = 4
y = 6 z = 3

4.4 Source Code

Fungsi cipher

```
    private String Kievn_Cipher(String
input)
          /**
         * terdiri dari 2 bagian utama, yaitu
         * inisiasi terdiri dari pernyiapan variable
perhitungan dan pemasukan
         * data kunci
         * pada bagian proses cipher terjadi
penhitungan hasil enkripsi yang akan
         * menjadi
         */
    {
        //inisiasi
        String plain = input.toLowerCase();
        String cipher = "";
        enc = new String[26];
        encinputdef();
        count = new int[26];
        for (int i = 0 ; i < 26 ; i ++ ) {
            count[i]=0;
        }
        int x;
        //proses
        for (int i = 0; i <
plain.length();i++) {
            if ((int)plain.charAt(i) > 96 &&
(int)plain.charAt(i) < 123) {
                cipher = cipher
+""+(enc[(int)plain.charAt(i) -97]);
                x =
(int)((enc[(int)plain.charAt(i) -
97]).charAt(0) - 97);
                count[x]++;

                if (count[x] % limit == limit-
1) tukar(x);
            }
        }
        return cipher;
    }
```

Fungsi Decipher

```
    private String Kievn_DeCipher(String
input)
          /**
         * terdiri dari 2 bagian utama, yaitu
         * inisiasi terdiri dari pernyiapan variable
perhitungan dan pemasukan
         * data kunci
         * pada bagian proses decipher yang
dihitung adalah teks
         * masukan yang merupakan hasil enkripsi
         */
    {
        //inisiasi
        String cipher =
input.toLowerCase();
        String plain = "";
        enc= new String[26];
        encinputdef();
        count = new int[26];
        for (int i = 0 ; i < 26 ; i ++ ) {
            count[i]=0;
        }
        int x;
        //proses
        for (int i = 0; i <
cipher.length();i++) {
            if ((int)cipher.charAt(i) > 96 &&
(int)cipher.charAt(i) < 123) {
                plain = plain
+""+(String.valueOf(returns(cipher.charA
t(i))));
                x = ((int)(cipher.charAt(i)))-
97;
                count[x]++;
                if (count[x] % limit == limit-
1) tukar(x);
            }
        }
        return plain;
    }
```

Fungsi Tukar

```
    public void tukar(int in)
        /**
        * fungsi pernukaran digunakan untuk
menukar kunci
        * fungsi ini digunakan untuk mengenkripsi
dan mendekripsi
        */
    {
        int min = count[in];
        int idx = -1;
        for (int i = 0 ; i < 26 ; i ++ )
        {
            if (count[i] < min)
            {
                min = count[i];
                idx = i;
            }
        }
        if(idx != -1)
        {
            in = returnint(in);
            idx = returnint(idx);
            String temp = enc[in];
            enc[in] = enc[idx];
            enc[idx] = temp;
        }
    }
}
```

Fungsi lain

```
    public char returns (char a)
        /**
        * mengembalikan karakter yang dienkripsi
menjadi masukan a
        */
    {
        for(int i = 0 ; i < 26 ; i++)
        {
            if
(enc[i].equals(String.valueOf(a)))
            {
                return (char)(i+97);
            }

        }
        return '0';
    }
    public int returnint (int a)
        /**
        * mengembalikan integer index dari huruf
yang memiliki hasil
        * huruf enkripsi index masukan a
        */
    {
        for(int i = 0 ; i < 26 ; i++)
        {
            if (enc[i].charAt(0) ==
(char)(a+97))
            {
                return i;
            }
        }
    }
return -1;
}
```

4.5 Interface Aplikasi Analisi Kievn Ciphere

4.6 Analisis Kievn Cipher

Berdasar hasil percobaan dengan berbagai plainteks yang ada, maka dapat disimpulkan Kievn Cipher memiliki karakteristik sebagai berikut :

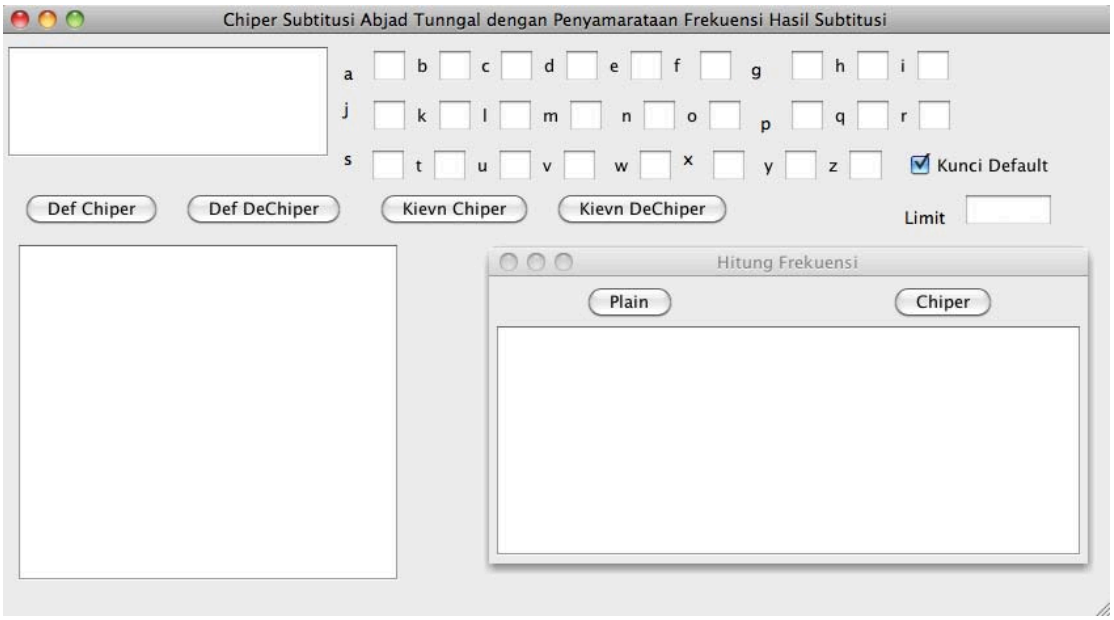
- Kievn Cipher sangat efektif digunakan untuk menyamaratakan frekuensi hasil enkripsi plainteks yang ukurannya relatif besar.
- Kievn Cipher dapat digunakan untuk mengenkripsi pesan yang relatif kecil, namun harus dengan menggunakan n yang kecil agar dapat mengacak frekuensi yang ada.
- Penambahan n pada kievn cipher mempercepat proses enkripsi, namun persebaran dari huruf tidak merata, jika kita membandingkan antara cipher dengan n=17 dengan n = 5 pada teks yang tegolong besar, maka dengan n = 5, teks yang ada semakin teracak susunan hurufnya.
- Dengan n yang relatif besar dan ukuran plainteks yang kecil perbedaan hasil enkripsi tidak terlalu berbeda dengan cipher subtitusi abjad tuggal

5.KESIMPULAN

Kievn Cipher adalah algoritma kriptografi yang mengembangkan algoritma kriptografi subtitusi abjad tunggal yang dapat melindungi hasil enkripsi dari serangan kriptanalis yang menggunakan frekuensi analisis dalam memecahka kode yang ada.

Walaupun algoritma kriptografi subtitusi abjad tunggal termasuk dalam algoritma yang dapat digunakan dengan pensil dan kertas, kerumitan dari pertukaran kunci yang ada pada kievn cipher tidak mudah untuk digunakan dengan menggunakan pensil dan kertas, selain karena algoritma ini juga dirancang bertujua untuk menjaga pesan denga ukuran yang relatif panjag, menutupi kelemahan algoritma subtitusi abjad tunggal.

Interface Aplikasi yang dibangun



DAFTAR PUSTAKA

R. Munir, "*Diktat Kuliah IF5054 Kriptografi*". Program Studi Teknik Informatika, Institut Teknologi Bandung, 2006.

http://en.wikipedia.org/wiki/Substitution_cipher , diakses Selasa 23 Maret 2010 pkl 22:53

http://en.wikipedia.org/wiki/Atbash ,, diakses Selasa 23 Maret 2010 pkl 23:29

http://en.wikipedia.org/wiki/Caesar_cipher, diakses Selasa 23 Maret 2010 pkl 11:39

http://en.wikipedia.org/wiki/Frequency_analysis, diakses Rabu 24 Maret 2010 pkl 2:24