# *Cipher* substitusi pengembangan dari sandi merah putih pada pramuka

Widhaprasa Ekamatra Waliprana - 13508080
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*If18080@students.if.itb.ac.id*

***Abstract*—Informasi merupakan hal yang dipertukarkan pada proses informasi. Untuk menjaga informasi dapat digunakan ilmu kriptografi. Algoritma kriptografi terbagi menjadi 2 yaitu algoritma kriptografi klasik dan algoritma kriptografi modern. Salah satu dari algoritma kriptografi klasik adalah cipher substitusi. Cipher substitusi merupakan metode mengganti karakter pada plain teks tepat menjadi karakter lainnya pada cipher teks. Oleh karena kesederhanaan dari cipher substitusi ini maka cipher ini dapat dipecahkan dengan menggunakan teknik analisis frekuensi. Teknik analisis frekuensi adalah teknik menggunakan hubungan frekuensi huruf yang sering muncul. Oleh karena itu akan dikembangkan suatu cipher substitusi yang merupakan pengembangan sandi merah putih pada pramuka. Sandi merah putih adalah sandi yang proses enkripsinya adalah mengganti satu huruf menjadi dua huruf yang merupakan perpaduan dari dua kunci yaitu merah dan putih. Sedangkan cipher substitusi pengembangannya adalah kunci yang digunakan dapat menjadi masukan sang pemilik pesan sehinngga cipher teks yang dihasilkan lebih bervariasi dan sulit untuk dipecahkan menggunakan teknik analisis frekuensi. Dengan dicetuskan ide dan diimplementasikannya cipher substitusi pengembangan dari sandi merah putih pada pramuka ini diharapkan terbentuk metode-metode kriptografi baru sehingga perkembangan ilmu kriptografi semakin meningkat pesat.**

***Index Terms*—Cipher Substitusi, Kriptografi, Sandi Merah Putih, Teknik Analisis Frekuensi.**

## I. PENDAHULUAN

Dalam proses komunikasi pasti akan ada pertukaran informasi. Informasi merupakan aspek yang penting dalam proses komunikasi baik 2 arah maupun ke banyak arah. Informasi yang dipertukarkan dalam proses komunikasi pun bermacam-macam. Jika dilihat dari segi isinya informasi dapat terbagi menjadi informasi publik ataupun prifat. Sejak zaman dahulu kala, setiap orang senantiasa berusaha untuk melindungi kerahasiaan dari informasi yang tersimpan pada suatu pesan yang dikomunikasikan melalui media tertentu kepada orang yang dimaksud agar informasi tersebut hanya diterima orang ataupun instansi yang ditujukan. Hal ini ditujukan agar informasi yang kita dimiliki tidak dimanipulasi pihak lain sehingga akan mengakibatkan kerugian yang sangat besar bagi kita. Oleh karena itu setiap orang akan selalu menjaga kerahasiaan informasinya dengan cara apapun.

Kriptografi merupakan solusi dari permasalahan tersebut. Kriptografi merupakan ilmu dan seni untuk menjaga kerahasiaan pesan yang dalam hal ini adalah informasi dengan cara mengubahnya ke bentuk yang sulit dimengerti maknanya. Hal ini bertujuan agar sebuah pesan yang disampaikan hanya akan dapat dimengerti oleh orang yang berhak untuk mengetahuinya saja, tidak ada pihak lain yang terlibat. Kriptografi ini telah digunakan sejak zaman dahulu pula, bangsa-bangsa pada peradaban kuno sudah memakainya seperti mesir pada ribuan tahun yang lalu dan juga peradaban lainnya yang telah menggunakan konsep kriptografi dalam menyamarkan pesan rahasia dalam strategi perang.

Ilmu kriptografi semakin banyak digunakan dan mulai berubah menjadi kebutuhan utama. Dengan maraknya perkembangan ilmu teknologi informasi keamanan pun menjadi penting dan keamanan tidak hanya sekedar pada media tulis saja. Banyak teknologi-teknologi baru yang menyimpan informasi penting yang kita miliki dan tidak boleh jatuh ke tangan yang tidak berhak seperti nomor PIN dan *password*. Pada bidang inilah ilmu kriptografi dibutuhkan, agar informasi tersebut tidak jatuh ke tangan yang salah.

Kriptografi dapat terbagi ke dalam dua bagian yaitu algoritma kriptografi klasik dan algoritma kriptografi modern. Algoritma kriptografi klasik adalah algoritma pengamanan teks yang bersifat sederhana, berbasis pada pemrosesan per-karakter pada suatu teks dan dapat dilakukan tanpa menggunakan komputer. Sedangkan algoritma kriptografi modern adalah algoritma kriptografi

yang menggunakan algoritma kompleks dan menggunakan pengolahan berbasis bit dalam proses enkripsi pesannya.

Meskipun sekarang algoritma klasik sudah tidak mulai dipakai namun algoritma klasik adalah awal mula ditemukan algoritma-algoritma modern lainnya. Dari sini pulalah terbentuk metode keamanan dan kerahasiaan data yang digunakan pada setiap teknologi informasi yang kita gunakan sekarang-sekarang ini. Algoritma kriptografi klasik digunakan dan kemudian dikembangkan serta disesuaikan dengan kebutuhan dan perkembangan zaman.

Kriptografi klasik ini penting untuk dipelajari karena mudah dipahami dan dikembangkan menjadi algoritma yang memiliki tingkat keamanan dan kerahasiaan lebih baik. Seperti cipher substitusi yang dapat dikembangkan menjadi cipher substitusi pengembangan dari sandi merah putih yang terdapat dalam makalah ini.

## II. DASAR TEORI

Pengimplementasian cipher substitusi ini dilakukan berdasarkan atau berlandaskan beberapa teori yang telah ada sebelumnya. Pada bagian ini akan dijelaskan kembali teori-teori yang melandasi pengimplementasian cipher substitusi ini.

### 2.1 Cipher Substitusi

Cipher substitusi adalah metode enkripsi dimana plain teks digantikan dengan cipher teks dengan sistem yang sederhana. Bagian dari plain teks yang disubstitusi adalah setiap karakter (pada umumnya) yang terdapat pada plain teks tersebut. Kemudian prosesnya adalah mengganti setiap huruf dengan huruf lainnya sehingga menghasilkan cipher teks yang memiliki panjang sama dengan plain teksnya.

Cipher subtitusi merupakan algoritma yang termsuk dalam pencil and paper cipher yaitu saking sederhananya tidak perlu menggunakan komputer dalam penerapannya sehingga hanya menggunakan selembar kertas dan sebuah pensil untuk melakukan proses enkripsi ataupun menuliskan pola enkripsi dari algoritma tersebut.

Penerapan cipher substitusi yang sangat dikenal dan yang dianggap menjadi pelopor dari cipher substitusi ini adalah Caesar cipher. Caesar cipher adalah algoritma kriptografi yang diciptakan oleh kaisar romawi terkenal yang bernama Julius Caesar. Julius hendak mengirimkan pesan kepada gubernur-gubernurnya. Namun agar kerahasiaan dan keamanan terjaga, maka Julius membuat sandi yang sampai saat ini dinamakan Caesar cipher. Proses sandi yang dilakukan adalah dengan menggesaer abjad yang ada dengan abjad lain yang letaknya berbeda sesuai dengan abjad lain yang letaknya berbeda sesuai dengan jumlah pergeseran yang digunakan

**Gambar 1: Caesar Cipher**

Caesar cipher adalah cipher yang paling sederhana yang pernah digunakan. Bahkan pada zaman tersebut banyak pula pihak-pihak lain yang menggunakan metode pengenkripsian ini selain Julius Caesar. Caesar cipher ini terus berkembang agar dapat menjadi metoda keamanan dan kerahasiaan pesan yang baik.

Cara pemecahan Caesar cipher ini cukup dengan menggeser abjad-abjad yang ada sebanyak 26 kali sehingga dapat menemukan plain teks yang diinginkan.

Penerapan cipher substitusi lain yang terkenal adalah Atbash cipher yang digunakan oleh bangsa Yahud Ibrani untuk menyembunyikan pesan yang ada pada zaman tersebut seperti dalam kitab, buku ajaran, mistisme, dan lain sebagainya. Atbash cipher ini bekerja dengan cara menukarkan huruf pertama dengan huruf pertama terakhir dan seterusnya.

Pada penggunaan Atbash cipher ini terdapat kata yang kembali menjadi kata dalam bahasai inggris baik sebelum dienkripsi, maupun sesudah dienkripsi. Kata-kata tersebut diantaranya adalah “hob” = “sly”, “hold” = “slow”, “holy”=slob”, “hom”=”slim”, “zoo”=”all”, “irk”=”rip”, “low”=”old”, “glow”=”told”, dan “grog”=”tilt”.

Dalam penggunaannya, cipher substitusi memiliki banyak jenis penerapan dalam mengenkripsi pesan, diantaranya adalah cipher substitusi abjad tunggal, cipher substitusi homofonik, cipher substitusi abjad majemuk, dan cipher substitusi poligram.

#### 2.1.1 Cipher Abjad Tunggal

Cipher abjad tunggal adalah salah satu jenis cipher substitusi yang mengganti dengan satu huruf

lainnya yang bersesuaian. Jumlah kemungkinan susunan huruf-huruf cipher teks yang dapat dibuat adalah sebesar:

26! = 403.291.461.126.605.635.584.000.000

Contoh dari cipher abjad tunggal adalah Caesar Cipher yang sudah dijelaskan pada bagian sebelumnya.

Cara penyusunan substitusi dari plain teks ke cipher teksnya adalah dengan memasukkan kunci yang mudah diingat sebagai pembangkit. Mekanisme pembangkitan pola menggunakan kunci yang dimasukkan adalah dengan cara sebagai berikut:

1. Memilih sebuah kunci yang mudah diingat. Sebagai contoh, kunci yang mudah diingat adalah: "into the new world".
2. Buang semua spasi yang ada hingga menjadi "intothenewworld".
3. Kemudian buang huruf yang berulang dan sisakan satu huruf di tempat yang paling depan sehingga menjadi "intohewrld" yang berisi tanpa pengulangan huruf.
4. Kemudian ditambahkan dengan huruf alfabet sisa yang belum ada pada kata tersebut ke bagian belakang kata tersebut hingga menjadi "intohewrldabcfgjkmpqsuvxyz". Kata ini harus memiliki 26 variasi huruf yang terdapat pada alfabet. Kata ini menjadi kumpulan kata yang tidak beraturan.
5. Pasangkan huruf dalam alfabet pada plain teks secara beraturan (A,B,C,...,Z) dengan kata tadi secara berurutan.

Dengan cara di atas, maka kita dapatkan pasangan plain teks dengan cipher teksnya sebagai berikut:

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

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

Cara ini sebenarnya cukup baik, namun masih dapat dipecahkan dengan cara teknik analisis frekuensi yang akan dijelaskan pada bagian nanti.

### 2.1.2 Cipher Homofonik

Cipher substitusi homofonik adalah cipher substitusi yang mensubstitusikan suatu huruf dengan kumpulan juruf lain dengan suatu huruf dapat menjadi beberapa kumpulan huruf lain. Tujuannya adalah Cipher subtitusi homofonik adalah menyembunyikan hubungan langsung plain teks dengan cipher teks agar tidak mudah dipecahkan dengan cara teknik analisis frekuensi.

Unit cipherteks mana yang dipilih diantara semua homofon ditentukan secara acak. Contoh cipher homofonik adalah setiap huruf Adapat disubstitusi dengan BU, CP, AV, huruf B dengan AT, huruf C dengan DL, BK, AU, dan lain-lain.

Dengan cipher homofonik ini akan sulit ditebak apakah huruf yang sering keluar sesuai dengan teknik analisis frekuensi atau tidak sehingga tingkat keamanan dan kerahasiaan data ini semakin terjaga.

### 2.1.3 Cipher Abjad Majemuk

*Cipher* abjad-majemuk dibuat dari sejumlah *cipher* abjad-tunggal, masing-masing dengan kunci yang berbeda. Kebanyakan *cipher* abjad-majemuk adalah *cipher* substitusi periodik yang didasarkan pada periode *m*. Skemanya adalah sebagai berikut:

Plainteks:

$P = p_1 p_2 \ldots p_m p_{m+1} \ldots p_{2m} \ldots$

Cipherteks:

$E_k(P) = f_1(p_1)\ f_2(p_2) \ \ldots\ f_m(p_m)\ f_{m+1}(p_{m+1}) \ \ldots\ f_{2m}(p_{2m}) \ \ldots$

Untuk $m = 1$, *cipher*-nya ekivalen dengan *cipher* abjad-tunggal. Contoh *cipher* substitusi periodik adalah *vigenere cipher.*

Kunci: $K = k_1 k_2 \ldots k_m$

$k_i$ untuk $1 \le i \le m$ menyatakan jumlah pergeseran pada huruf ke-*i*.

Karakter cipherteks: $c_i(p) = (p + k_i) \bmod 26$ *(*)*

Misalkan periode $m = 20$, maka 20 karakter pertama dienkripsi dengan persamaan (*), setiap karakter ke-*i* menggunakan kunci $k_i$. Untuk 20 karakter berikutnya, kembali menggunakan pola enkripsi yang sama.

Contoh:

P:
KRIPTOGRAFIKLASIKDENGANCIPHERALFABETMAJEMUK
K:
LAMPIONLAMPIONLAMPIONLAMPIONLAMPIONLAMPIONL
C:
**VR...**

Perhitungan:
(K + L) mod 26 = (10 + 11) mod 26 = 21 = **V**
(R + A) mod 26 = (17 + 0) mod 26 = 17 = **R**
Dst.

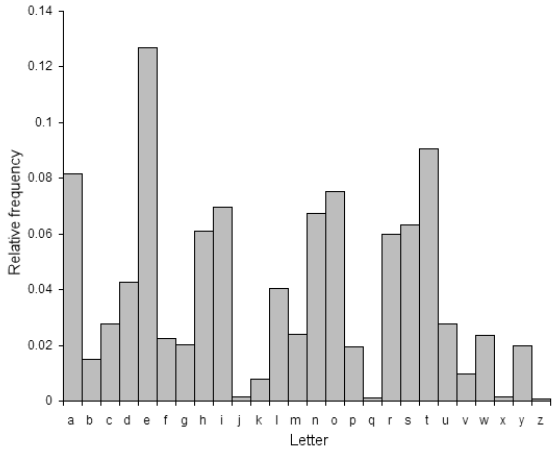
Contoh 2:
P:
SHE SELLS SEA SHELLS BY THE SEASHORE
K:
KEY KEYKE YKE YKEYKE YK EYK EYKEYKEY
C:
CLC CIJVW QOE QRIJVW ZI XFO WCKWFYVC

### 2.1.4 Cipher substitusi poligram

- Blok huruf plainteks disubstitusi dengan blok cipherteks.
- Misalnya AS diganti dengan RT, BY diganti dengan SL
- Jika unit huruf plainteks/cipherteks panjangnya 2 huruf, maka ia disebut digram (*biigram*), jika 3 huruf disebut ternari-gram, dst
- Tujuannya: distribusi kemunculan poligram menjadi *flat* (datar), dan hal ini menyulitkan analisis frekuensi

### 2.2 Teknik Analisis Frekuensi

Setiap cipher substitusi dapa dipecahkan dengan menggunakan teknik yang dinamakan teknik analisis frekuensi. Teknik analisis frekuensi adalah menggunakan hubungan kemunculan frekuensi huruf pada plain teks dan cipher teks yang tidak mengalami perubahan. Maksudnya adalah jika menggunakan cipher substitusi, maka kemunculan huruf e pada plain teks dengan kemunculan huruf hasil terenkripsi dari e akan sama. Hal inilah yang dimanfaatkan para penyerang untuk menemukan kunci ataupun plain teks yang dimaksud. Di bawah ini adalah gambar diagram distribusi huruf yang sering keluar pada kata-kata bahasa inggris.




**Gambar 2: Diagram distribusi huruf yang sering keluar pada teks bahasa inggris**

Di bawah ini adalah tabel frekuensi huruf pada bahasa inggris dengan peluangnya.

| Huruf | Peluang |
|---|---|
| A | 0.082 |
| B | 0.015 |
| C | 0.028 |
| D | 0.043 |
| E | 0.127 |
| F | 0.022 |
| G | 0.020 |
| H | 0.061 |
| I | 0.070 |
| J | 0.002 |
| K | 0.008 |
| L | 0.040 |
| M | 0.024 |
| N | 0.067 |
| O | 0.075 |
| P | 0.019 |
| Q | 0.001 |
| R | 0.060 |
| S | 0.063 |
| T | 0.091 |
| U | 0.028 |
| V | 0.010 |
| W | 0.023 |
| X | 0.001 |
| Y | 0.020 |
| Z | 0.001 |

**Tabel 1: Peluang kemunculan huruf pada teks bahasa inggris**

Dari tabel diatas frekuensi huruf yang sering keluar dapat dibagi menjadi lima kelompok, yaitu:

1. E.mempunyai peluang 0.120
2. T, A, O, I, N, S, H, R mempunyai peluang antara 0.06 dan 0.09.
3. D, L mempunyai peluang 0.04.
4. C, U, M, W, F, G, Y, P, B mempunyai peluang antara 0.015 dan 0.023.
5. V, K, J, X, Q, Z mempunyai peluang kurang dari 0.01.

Kemudian ada kumpulan huruf yang disebut sebagai bigram untuk 2 huruf, trigram untuk 3 huruf, dan n-gram untuk n huruf.

Di bawah merupakan bigram yang paling sering keluar pada tulisan bahasa inggris diantaranya adalah:

TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF.

Dan trigram yang paling sering keluar adalah trigram THE.

## III. PERANCANGAN METODE CIPHER

Cipher substitusi ini dibuat memakai ide dasar dari sandi merah putih yang diajarkan pada pramuka. Berikut akan dijelaskan secara singkat bagaimana sandi merah putih dan akan dijelaskan bagaimana skema cipher substitusi pengembangannya.

### 3.1 Sandi Merah Putih

Sandi merah putih termasuk dalam cipher substitusi yaitu mengganti satu karakter pada teks dengan karakter lainnya, hanya saja pada sandi merah putih ini karakter diganti dengan pasangan 2 karakter. Sandi merah putih ini mempunyai dua kunci yaitu kunci horizontal dengan kunci "MERAH" dan kunci vertikal dengan kunci "PUTIH". Tabel substitusi dari sandi merah putih dapat dilihat di bawah ini:

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

Z

**Gambar 3: Tabel substitusi sandi merah putih**

Dengan melihat tabel di atas kita dapat menentukan huruf 'A' sebagai pasangan huruf 'MP', huruf 'B' sebagai pasangan huruf 'EP', huruf 'C' sebagai pasangan huruf 'RP' dan seterusnya. Sedangkan huruf 'Z' tidak akan disubstitusi dikarenakan jumlah kotak hanya seluas 5 x 5 = 25 kotak.

### 3.2 Pengembangan dari sandi merah putih

Sandi merah putih yang baru dijelaskan akan dikembangkan sehingga menjadi sedikit berbeda. Pengembangannya adalah dengan cara memasukkan 2 kunci sebagai kunci horizontal dan kunci vertikalnya sebagai pengganti dari "MERAH" dan "PUTIH". Pengembangan lainnya adalah huruf 'Z' pada plain teks akan dirandom sesuai dengan key horizontal dan key vertikalnya.

Untuk karakter lain di luar ke-26 huruf maka akan diubah menjadi huruf vertikal pertama ditambah dengan huruf tersebut. Sebagai contoh, jika dimasukkan kunci horizontal = "KRIPT" dan kunci vertikal = "GRAFI" maka tabel yang akan terbentuk adalah sebagai berikut:

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

**Gambar 4: Tabel substitusi pengembangan sandi merah putih**

Dengan melihat tabel di atas kita dapat menentukan huruf 'A' sebagai pasangan huruf 'KG', huruf 'B' sebagai pasangan huruf 'RG', huruf 'C' sebagai pasangan huruf 'IG' dan seterusnya. Huruf 'Z' akan disubstitusi secara random dengan salah satu pasangan huruf di atas seperti "KG", "RG", ataupun "TI". Untuk karakter lain seperti *white space* (' ') dan angka kita ambil contoh saja 1akan menjadi "G " dan "G1".

Batasan dari cipher ini adalah panjang kunci horizontal dan vertikal harus sepanjang 5 dan tidak boleh ada huruf yang berulang di kuncinya. Sebagai contoh 'KRIPT' tidak terdapat huruf yang berulang dan 'MAKAN' terdapat huruf yang berulang yaitu 'A'.

## IV. IMPLEMENTASI DAN UJI COBA

Pada bagian ini telah diimplementasikan aplikasi pengembangan cipher substitusi ini dan juga uji coba dari aplikasi ini.

### 4.1 Implementasi Aplikasi

Di bawah ini adalah source code aplikasi yang dikembangkan menggunakan bahasa C#.

```
public class Cipher
    {
        public String HorizontalKey{get;
```

```
set;}
        public String VerticalKey{get;
set;}
        private char[,] CipherTable;
        private Dictionary<String, String>
EncryptDictionary;
        private Dictionary<String, String>
DecryptDictionary;

        //Konstruktor
        public Cipher()
        {
            HorizontalKey = "MERAH";
            VerticalKey = "PUTIH";

            CipherTable = new char[5, 5];
            EncryptDictionary = new
Dictionary<String, String>();
            DecryptDictionary = new
Dictionary<String, String>();
            GenenerateTable();
        }

        public Cipher(String hor, String
ver)
        {
            if (!ValidateKey(hor, ver))
            {
                HorizontalKey = hor;
                VerticalKey = ver;
                return;
            }

            HorizontalKey = hor;
            VerticalKey = ver;

            CipherTable = new char[5, 5];
            EncryptDictionary = new
Dictionary<String, String>();
            DecryptDictionary = new
Dictionary<String, String>();
            GenenerateTable();
        }

        public void GenenerateTable()
        {

            int Current = (int)'A';
            for (int j = 0; j <= 4; j++)
            {
                for (int i = 0; i <= 4;
i++)
                {
                    CipherTable[i, j] =
(char)Current;

EncryptDictionary.Add(CipherTable[i,
j].ToString().ToUpper() ,("" +
HorizontalKey[i] +
VerticalKey[j]).ToUpper());

DecryptDictionary.Add(("" +
HorizontalKey[i] +
VerticalKey[j]).ToUpper(), CipherTable[i,
j].ToString().ToUpper());
                    Current++;
                }
            }

        }

        public Boolean ValidateKey(String
Key)
        {
            if (Key.Length != 5)
            {
                return false;
            }

            for (int j = 0; j <= Key.Length
- 1; j++)
            {
                for (int i = j + 1; i <=
Key.Length - 1; i++)
                {
                    if (j != i && Key[j] ==
Key[i])
                    {
                        return false;
                    }
                }
            }

            return true;
        }

        public Boolean ValidateKey(String
H, String V)
        {
            return ValidateKey(H) &&
ValidateKey(V);
        }

        public Boolean IsValidChar(String
input) {
            String tempinput =
input.ToUpper();
            if(tempinput.Length > 2){
                return false;
            }

            if (tempinput.Length == 1) {
                if (tempinput[0] < 'A' ||
tempinput[0] > 'Z') {
                    return false;
                }
            }

            if (tempinput.Length == 2) {
                if (tempinput[0] < 'A' ||
tempinput[0] > 'Z' || tempinput[1] < 'A' ||
tempinput[1] > 'Z') {
                    return false;
                }
            }

            return true;
        }

        public String EncryptPerChar(String
input)
        {
            String tempinput =
input.ToUpper();
            if (tempinput.Length != 1)
            {
                return "";
            }

            if
(!EncryptDictionary.ContainsKey(tempinput))
            {
                return "";
            }

            if(input.Equals("Z")){
                Random Rand = new Random();
                int randomint =
Rand.Next(0,24);
                char tempchar =
(char)(randomint + (int) 'A');
                return
EncryptDictionary[tempchar.ToString()];
            }

            return
EncryptDictionary[tempinput];
        }

        public String DecryptPerChar(String
input)
        {
            String tempinput =
input.ToUpper();
            if (tempinput.Length != 2)
            {
```

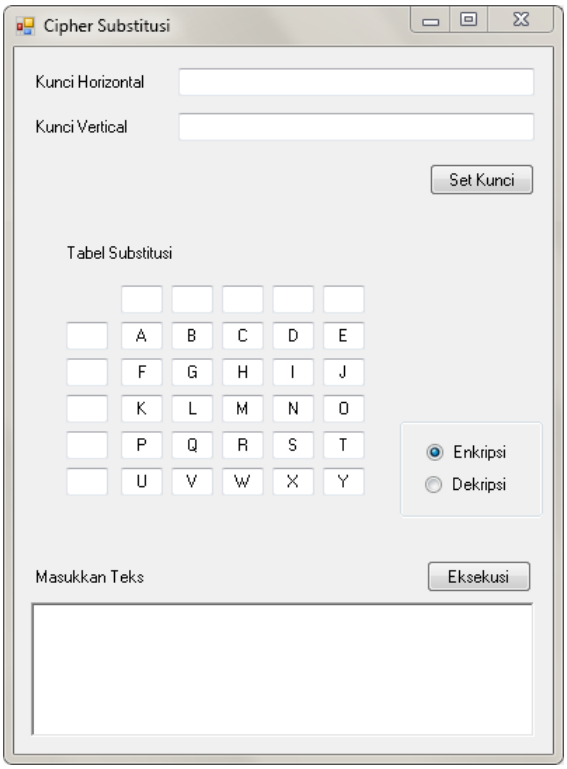
```
                return "";
            }

            if
(!DecryptDictionary.ContainsKey(tempinput))
            {
                return "";
            }

            return
DecryptDictionary[tempinput];
        }

        public String Encrypt(String input)
{
            String tempinput =
input.ToUpper();
            StringBuilder output = new
StringBuilder();
            for (int i = 0; i <=
tempinput.Length - 1; i++) {
                if
(!IsValidChar(tempinput[i].ToString()))
                {
                    output.Append("" +
VerticalKey[0].ToString().ToUpper() +
tempinput[i].ToString());
                }
                else
                {

output.Append(EncryptPerChar(tempinput[i].T
oString()));
                }
            }
            return output.ToString();
        }

        public String Decrypt(String input)
        {
            String tempinput =
input.ToUpper();
            String tempprocess;
            StringBuilder output = new
StringBuilder();
            for (int i = 0; i <=
tempinput.Length - 1; i += 2)
            {
                if (i + 1 >=
tempinput.Length) {
                    break;
                }

                tempprocess =
tempinput[i].ToString() + tempinput[i +
1].ToString();
                if
(!IsValidChar(tempprocess))
                {

output.Append(tempinput[i+1].ToString());
                }
                else
                {

output.Append(DecryptPerChar(tempprocess));
                }
            }
            return output.ToString();
        }

    }
```

Pada kelas ini disimpan kunci horizontal dan kunci vertikalnya. Struktur data yang digunakan adalah Dictionary yaitu sebuah kontainer yang memetakan setiap huruf ke pasangan huruf lainnya. Sehingga untuk mengubah suatu huruf tinggal memanggil hasil pemetaannya saja. Terdapat juga fungsi ValidateKey yang berfungsi sebagai batasan untuk kunci yang dimasukkan.

### 4.2 Uji Coba Aplikasi

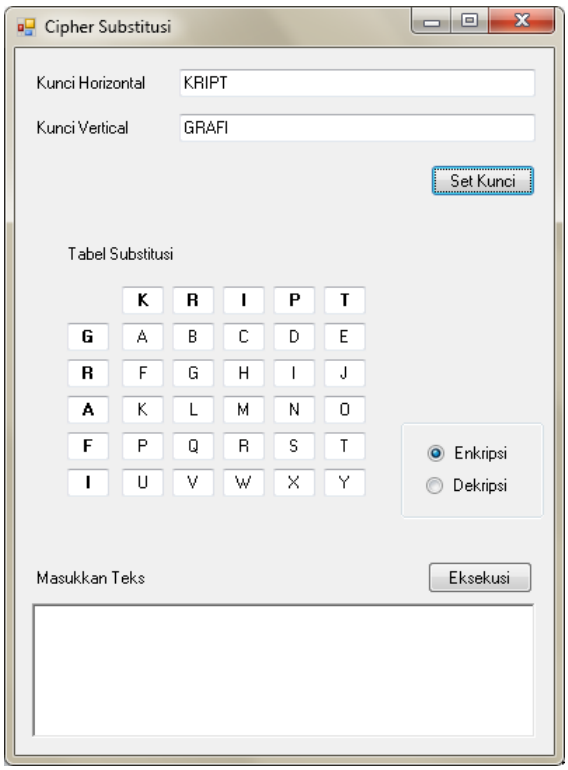
Pada bagian ini akan ditampilkan hasi *screen shoot* aplikasi cipher substitusi yang dikembangkan menggunakan bahasa C#.

Di bawah ini adalah tampilan aplikasi secara umum sebelum digunakan.



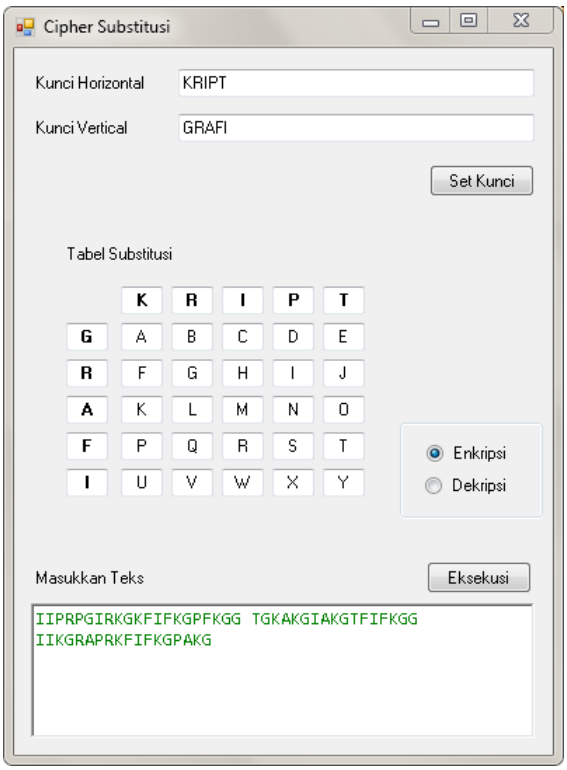

**Gambar 5: Tampilan utama aplikasi**

Di bawah ini adalah tampilan aplikasi ketika key yang valid dimasukkan. Maka gambar tabel substitusi akan muncul pada program.




**Gambar 6: Tampilan aplikasi ketika key dimasukkan**

Di bawah ini adalah tampilan aplikasi ketika mengenkripsi suatu kalimat. Pada contoh kali ini kalimat yang dicoba adalah "WIDHAPRASA EKAMATRA WALIPRANA" dan dengan kunci horizontal "KRIPT" dan kunci vertikal "GRAFI" maka akan menghasilkan "IIPRPGIRKGKFIFKGPFKGG TGKAKGIAKGTFIFKGG IIKGRAPRKFIFKGPAKG".




**Gambar 7: Tampilan aplikasi setelah proses enkripsi dilakukan**

Cipher teks yang telah terenkripsi ini sudah pasti memiliki panjang karakter yang lebih panjang dibandingkan plain teksnya. Maka diharapkan serangan dengan menggunakan teknik analisis frekuensi tidak berhasil memecahkan pesan cipher teks yang dimaksudkan. Namun cipher teks ini masih bisa dianalisis lebih lanjut seperti kemunculan huruf 'G' yang sering muncul pada cipher teks yang berarti pada himpunan huruf (A,B,C,D,E) terdapat huruf yang sering muncul pada plain teks.

## V. KESIMPULAN

Dalam pengimplimentasian cipher substitusi baru pengembangan dari sandi merah putih pada pramuka ini dapat ditarik kesimpulan sebagai berikut:

- Kriptografi adalah ilmu ataupun seni untuk menjaga keamanan suatu informasi dengan mengubahnya menjadi bentuk yang sulit dimengerti maknanya.
- Cipher substiutsi merupakan metode kriptografi yang klasik dan dianggap menjadi pelopor dari metode-metode kriptografi lainnya yang lebih modern
- Cipher substitusi dapat dipecahkan menggunakan teori analisis frekuensi.

- Untuk meningkatkan keamanannya maka suatu cipher substitusi dikembangkan seperti dengan dicetuskan ide cipher substitusi pengembangan dari sandi merah putih pada pramuka.
- Dengan dicetuskan ide dan diimplementasikannya cipher substitusi pengembangan dari sandi merah putih pada pramuka ini diharapkan terbentuk metode-metode baru sehingga perkembangan ilmu kriptografi khususnya di Indonesia semakin meningkat pesat dan kerahasiaan data suatu informasi semakin terjaga dengan baik dan juga aman.

## REFERENSI


[1] Munir, Rinaldi. 2006. "Diktat Kuliah IF5054 Kriptografi". Program Studi Teknik Informatika, Institut Teknologi Bandung.

[2] http://en.wikipedia.org/wiki/Substitution_cipher

[3] http://guslatmipaunnes.wordpress.com/2010/03/15/sandi-merah-putih/

[4] http://en.wikipedia.org/wiki/Frequency_analysis


## 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, 23 Maret 2011

Widhaprasa Ekamatra Waliprana
13508080