Implementasi cipher dengan PHP

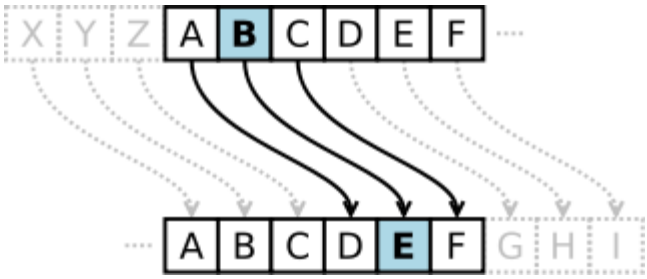
Ahmad Zafrullah
zaf@elkom08.x10.bz
http://elkom08.x10.bz

Cipher adalah suatu bentuk penyandian data (file, dll) kebentuk lain yang sulit untuk diketahui publik dengan menggunakan algoritma tertentu. Dimana disini yang akan saya bahas adalah caesar cipher dan vigenere cipher dengan implementasinya pada php.

1. CAESAR CIPHER

Dalam kriptografi, sandi Caesar, atau sandi geser, kode Caesar atau Geseran Caesar adalah salah satu teknik enkripsi paling sederhana dan paling terkenal. Sandi ini termasuk sandi substitusi dimana setiap huruf pada teks terang (*plaintext*) digantikan oleh huruf lain yang memiliki selisih posisi tertentu dalam alfabet. Misalnya, jika menggunakan geseran 3, `W` akan menjadi `Z`, `I` menjadi `L`, dan `K` menjadi `N` sehingga teks terang "wiki" akan menjadi "ZLNL" pada teks tersandi. Nama *Caesar* diambil dari Julius Caesar, jenderal, konsul, dan diktator Romawi yang menggunakan sandi ini untuk berkomunikasi dengan para panglimanya.

Langkah enkripsi oleh sandi Caesar sering dijadikan bagian dari penyandian yang lebih rumit, seperti sandi Vigenère, dan masih memiliki aplikasi modern pada sistem ROT13. Pada saat ini, seperti halnya sandi substitusi alfabet tunggal lainnya, sandi Caesar dapat dengan mudah dipecahkan dan praktis tidak memberikan kerahasiaan bagi pemakainya.



2. VIGENERE CIPHER

Sandi Vigenère sebenarnya merupakan pengembangan dari sandi Caesar. Pada sandi Caesar, setiap huruf teks terang digantikan dengan huruf lain yang memiliki perbedaan tertentu pada urutan alfabet. Misalnya pada sandi Caesar dengan geseran 3, A menjadi D, B menjadi E and dan seterusnya. Sandi Vigenère terdiri dari beberapa sandi Caesar dengan nilai geseran yang berbeda.

Untuk menyandikan suatu pesan, digunakan sebuah tabel alfabet yang disebut tabel Vigenère (gambar). Tabel Vigenère berisi alfabet yang dituliskan dalam 26 baris, masing-masing baris digeser satu urutan ke kiri dari baris sebelumnya, membentuk ke-26 kemungkinan sandi Caesar. Setiap huruf disandikan dengan menggunakan baris yang berbeda-beda, sesuai kata kunci yang diulang

Misalnya, teks terang yang hendak disandikan adalah perintah "Serbu Berlin":

```
serbuberlin
```

Sedangkan kata kunci antara pengirim dan tujuan adalah "Pizza"

"PIZZA" diulang sehingga jumlah hurufnya sama banyak dengan teks terang:

```
PIZZAPIZZAP
```

Huruf pertama pada teks terang, S, disandikan dengan menggunakan baris berjudul P, huruf pertama pada kata kunci. Pada baris P dan kolom S di tabel Vigenère, terdapat huruf H. Demikian pula untuk huruf kedua, digunakan huruf yang terletak pada baris I (huruf kedua kata kunci) dan kolom E (huruf kedua teks terang), yaitu huruf M. Proses ini dijalankan terus sehingga

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

Teks terang:	`serbuberlin`
Kata kunci:	`PIZZAPIZZAP`
Teks bersandi:	`HMQAUQMQKIC`

Proses sebalinya (disebut dekripsi), dilakukan dengan mencari huruf teks bersandi pada baris berjudul huruf dari kata kunci. Misalnya, pada contoh diatas, untuk huruf pertama, kita mencari huruf H (huruf pertama teks tersandi) pada baris P (huruf pertama pada kata kunci), yang terdapat pada kolom S, sehingga huruf pertama adalah S. Lalu M terdapat pada baris I di kolom E, sehingga diketahui huruf kedua teks terang adalah E, dan seterusnya hingga didapat perintah "serbuberlin".

Implementasi pada php

- ***CAESAR***

Langkah pertama yang dilakukan adalah membuat beberapa fungsi, dimana kegunaannya adalah untuk mengkonversi karakterASCII ke bentuk desimalnya, hal ini agar lebih mudah melakukan perhitungan.

```
function char_to_dec($a){
	$i=ord($a);
	if ($i>=97 && $i<=122){
		return ($i-96);
	} else if ($i>=65 && $i<=90){
		return ($i-38);
	} else {
		return null;
	}
}
```

Fungsi `ord()` berfungsi memberi nilai kembalian berupa nilai desimal dari karakter ASCII. Yang kita gunakan disini adalah karakter a-z dan A-Z.

ASCII	Desimal	Return ($i-96)	Return ($i-38)
a	97	1	-
b	98	2	-
z	122	26	-
A	65	-	27
B	66	-	28
Z	90	-	52

```
	if ($i>=97 && $i<=122){
```

Dari tabel diatas, bisa dilihat jelas kenapa pengkondisian mengambil interval dari 97 sampai 122, yaitu hanya untuk karakter a-z. Dan ini akan menempati angka 1-26

```
	} else if ($i>=65 && $i<=90){
```

Kemudian dari interval 65 sampai 90, yaitu hanya untuk karakter A-Z. Dan menempati angka 27-52. Dan terakhir akan mengembalikan **null** jika tidak bukan diantara interval diatas.

Setelah dilakukan konversi diatas menjadi bilangan desimal urut dari 1 – 52, maka akan dengan mudah melakukan pergeseran ke kanan (yang dibahas adalah pergeseran kekanan) yaitu dengan menambah bilangan desimal tersebut dengan berapa kali pergeseran akan dilakukan.

Kemudian dari konsep fungsi diatas, dapat dilakukan proses sebaliknya, yaitu konversi desimal kebentuk ASCII kembali.

```
function dec_to_char($a){
        if ($a>=1 && $a<=26){
             return (chr($a+96));
        } else if ($a>=27 && $a<=52){
             return (chr($a+38));
        } else {
             return null;
        }
}
```

Jika pada fungsi sebelumnya menggunakan pengurangan, maka fungsi diatas menggunakan operasi penjumlahan.

encrypt

Misalnya kita punya $key = 3, maka tiap karakter (a-z, A-Z) pada kata atau kalimat akan di geser sebanyak 3 kali kekanan. Bagaimana dengan karakter lain seperti spasi, titik, koma, dan lain?

Untuk karakter karakter yang disebutkan diatas tidak ikut digeser, karena tidak masuk ke interval yang ada dalam fungsi. Karakter spasi sendiri mempunyai nilai desimal 32.

Pertama tama deklarasikan dulu yang diperlukan :

```
        $key=3;
        $plantext="coba";
```

Sebelumnya, kita tahu bahwa karakter dasar yang kita gunakan adalah a-z dan A-Z, dimana total nya adalah 52. Bagaimana jika seseorang ingin menggeser (encrypt) lebih dari 52?

Mudah saja, tinggal buat decrement untuk menguragi key tersebut dengan 52 jika key yang dimasukkan melebihi 52.

```
        while ($key>52){
             $key=$key-52;
        }
```

kita tahu bahwa $plantext adalah string, agar mudah dioperasikan, kita perlu merupahnya (split) kedalam bentuk array.

```
        $split_chr=str_split($plantext);
```

Dengan begitu tiap karakter pada $plantext akan disimpan pada tiap index pada arrat $split_chr. Dengan menggunakan perulangan, akan lebih mudah untuk mengkonversikan tiap karakter dari bentuk ASCII nya ke dalam nilai desimalnya. Hasilnya juga disimpan kedalam bentuk array, yaitu $split_nmbr :

```
        foreach($split_chr as $chr){
             if (char_to_dec($chr)!=null){
                   $split_nmbr[$i]=char_to_dec($chr);
             } else {
                   $split_nmbr[$i]=$chr;
             }
             $i++;
        }
```

Dengan begini, $split_nmbr sudah menyimpan nilai desimal dari tiap karakter $plantext.

Pada perulangan diatas, ada pengkondisian jika pada fungsi char_to_dec() dimasukkan argumen karakter dan fungsi mengembalikan null, maka karakter tersebut akan disimpan langsung pada $split_nmbr tanpa dilakukan konversi.

```
        ...
             if (char_to_dec($chr)!=null){
                   $split_nmbr[$i]=char_to_dec($chr);
             } else {
                   $split_nmbr[$i]=$chr;
             }
        ...
```

Setelah itu mengkonversikan kembali bilangan desimal tersebut ke bentuk karakter ASCII nya, sebelumny dibuat dulu fungsi seperti berikut :

```
function dec_to_char($a){
        if ($a>=1 && $a<=26){
             return (chr($a+96));
        } else if ($a>=27 && $a<=52){
             return (chr($a+38));
        } else {
             return null;
        }
}
```

sama seperti fungsi sebelumnya (char_to_dec()), tapi bedanya yang ini menggunakan fungsi `chr()` yang berfungsi untuk mengembalikan bentuk karakter dari argumen desimal yang diberikan.

Membuat perulangan untuk menginputkan tiap nilai desimal pada tiap index dijumlahkan dengan $key (untuk pergeseran) pada $split_nmbr. Lalu bagaimana juga hasil penjumlahannya melebihi 52 (sudah dijelaskan diatas kita mengambil a-z, A-X yang totalnya 52) ?

jika melebihi 52 kita tinggal mengatur agar mulai mengulang kembali dari 1. sebagai contoh jika kita punya karakter 'Y' (50) dan $key 4, maka kita buat kondisi 'jika' hasil penjumlahan 'Y' (50) dan 4 melebihi 52, hasilnya kita kurangi dengan 52, sehingga akan didapat hasil

(50+4)-52 = 2

```
foreach($split_nmbr as $nmbr){
       if (($nmbr+$key)>52){
            if (dec_to_char($nmbr)!=null){
                  echo dec_to_char(($nmbr+$key)-52);
            } else {
                  echo $nmbr;
            }
       } else {
            if (dec_to_char($nmbr)!=null){
                  echo dec_to_char($nmbr+$key);
            } else {
                  echo $nmbr;
            }
       }
}
```

kenapa kita gunakan 'jika' ? Bagaimana jika hasil penjumlahannya melebihi 2 atau 3 kali lipat dari 52, sedang kita gunakan if, maka hanya akan dikurangi 52 sebanyak satu kali?? :D

mungkin flashback sedikit, tadi diatas sudah dibahas, jika ada yang mencoba menginputkan $key lebih besar dari pada 52, maka akan dikurangi 52 selama hasilnya masih diatas 52 :)

```
       while ($key>52){
            $key=$key-52;
       }
```

jadi hasil penjumlahannya tidak mungkin akan lebih besar 2 atau 3 kali lipat dari 52. dari pengkondisian diatas, digunakan beberapa tingkat pengkondisian :

sebelumnya pada fungsi dec_to_char() jika argument desimal yang di inputkan kurang atau lebih dari interval 1-52, maka akan di kembalikan berupa null. Jadi pengkondisian diatas berfungsi untuk mengecek jika yang dikembalikan adalah null (duluar interval) maka karakter tersebut akan langsung ditampilkan.

```
       ...
            if (dec_to_char($nmbr)!=null){
                  echo dec_to_char(($nmbr+$key)-52);
            } else {
                  echo $nmbr;
            }
       ...
```

decrypt

untuk decryptnya menggunakan algoritma yang sama, hanya saja di balik (reverse) dari fungsi fungsi yang sudah dijelaskan diatas,

Mengenai script lengkapnya bisa dilihat langsung pada file **index.php** dan **convert.php**

- ***VIGENERE***

berbeda dengan cipher sebelumnya, pada vigenere cipher kita perlu membuat fungsi untuk membentuk sebuah tabel vigenere seperti gambar dibawah ini :

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

tapi sebelumnya jangan lupa untuk mengulang key sehingga panjang karakter key sama dengan panjang karakter plantext nya.

encrypt

Kita anggap saja bahwa yang horizontal adalah tiap karakter plantext nya, dan vertikalnya adalah key nya. Misal kita punya karakter pertama untuk plantext “z”, key “n” maka karakter cipher text yang harus didapat adalah “m” sesuai tabel.

Kita gunakan fungsi konversi char_to_dec() yang telah buat diatas, untuk mengambil nilai desimal dari karakter “z”, “n”, dan “m”.

z = 26
n = 14
m = 13

setelah melakukan beberapa perhitungan dengan menggunakan beberapa sample, akhirnya didapat rumus :

c = p + k – 1 → jika c > 26, maka c = c - 26

dimana : c = karakter ciper text yang dicari
p = karakter plantext
k = karakter key

misal : m = 26 + 14 – 1
m = 39 (m > 26)
m = 39 – 26
m = 13 (terbukti)

dari penjelasan diatas, saya kira tidak sulit untuk membuat code nya dalam php :)

```
function tabel_vigenere_encrypt($a, $b){
       $i=$a+$b-1;
       if ($i>26){
            $i=$i-26;
       }
       return (dec_to_char($i));
}
```

fungsi diatas meminta parameter berupa nilai desimal dari karakter plantext ($a) dan nilai desimal dari karakter key ($b), kemudian mengembalikan bentuk karakter dari hasil perhitungan pada tabel vigenere dengan menggunakan fungsi dec_to_char().

Sisanya adalah tinggal bagaimana menginputkan plantext dan key kedalam fungsi, kemudian mendapatkan hasil yang dikembalikan oleh fungsi.

decrypt

untuk decrypt juga menggunakan algoritma yang sama, hanya saja logikanya dibalik (reverse). Untuk lebih jelasnya bisa dilihat langsung pada file index.php dan **conver.php**

Penutup

Saya menyadari bahwa tulisan ini masih jauh dari sempurna, karena kekurangan adalah milik sy pribadi dan kesempurnaan hanyalah milik sang Pencipta.

Referensi

http://elkom08.x10.bz/blog/?p=49
http://id.wikipedia.org
http://www.w3schools.com/

Biografi Penulis

Ahmad Zafrullah : Alumni **SMKN 2 Kuripan**, menempuh kuliah **S1 Teknik Elektro** bidang keahlian **Informatika** di **Universitas Mataram** angkatan 2008. Mulai tertarik dengan dunia pemrograman sejak dari semester awal.
Saat itu pertama kali diperkenalkan tentang bahasa C/C++, kemudian keinginan untuk mendalami pemrograman telah mengantarkan pada bahasa Java, HTML, dan PHP.
Membuat web pertamanya untuk angkatannya pada semester 4 menggunakan PHP dan MySQL (*http://e08.x10.bz*)
Mempunyai cita-cita membuat WEB CMS sendiri berbasis Open Source.