

# Implementasi Kriptografi pada PHP dan MySQL

Whilda Chaq - 13511601  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13511601@std.stei.itb.ac.id

**Abstrak** – Kriptografi sangat penting ketika kita membicarakan tentang keamanan suatu data. Kriptografi berguna untuk membuat sebuah pesan yang bisa dibaca (plaintext) menjadi pesan yang sudah di terjemahkan sehingga sulit untuk di baca secara langsung (chipertext). Proses mengubah dari plaintext menjadi chipertext di sebut enkripsi data dan dekripsi data adalah kebalikan dari enkripsi. Sudah banyak dikembangkan beberapa algoritma enkripsi data. Semakin rumit algoritma enkripsinya, semakin sulit pula chipertext dapat di baca(kerahasiaan lebih terjamin). Kini perkembangan jaman telah sampai kepada era digital dimana internet menjadi andalan di setiap sisi kehidupan. Dalam makalah ini ingin menunjukkan aplikasi kriptografi pada sebuah sistem berbasis web menggunakan bahasa pemrograman PHP dan database MySQL.

**Kata kunci** – Kriptografi, Enkripsi, PHP, MySQL, Dekripsi, Chipertext, dan Plaintext.

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Dewasa ini internet tengah menjadi ujung tombak dari kehidupan. Situs jejaring sosial, jual beli on-line, berbagai ilmu pengetahuan, gosip, mitos, organisasi dan masih banyak lagi semua tersedia dalam internet. Kebutuhan yang tinggi akan internet menjadi boomerang kepada pemakai jika kurang hati hati dalam memanfaatkan internet. Data yang kita unggah melalui alamat website tertentu secara tidak sadar bahwa kita telah mempublikasikan data pribadi kita. Lalu siapa yang menjamin dari kerahasiaan data yang kita unggah di internet. Password sebuah akun menjadi pusat perhatian karena dengan memasukkan sebuah password maka seorang pemakai internet bisa masuk kedalam suatu website tertentu dan dapat melakukan berbagai macam aksi sesuai dengan alam web yang di kunjunginya. Data yang kita unggah semua di tampung dalam sebuah server (database) dimana hanya kalangan terbatas dari administrator website tersebut yang dapat mengetahui data yang kita unggah. Peran kriptografi disini adalah merahasiakan data yang kita unggah agar administrator web sekalipun tidak dapat mengetahui data yang kita unggah. Pengolahan data hanya terjadi antara pemakai dan piranti yang di gunakan.

### 1.2 Tujuan

Tujuan dari makalah ini adalah untuk menunjukkan dengan memanfaatkan kriptografi pemakai internet tidak perlu khawatir mengenai kerahasiaan data yang kita unggah. Meskipun tidak semua data yang kita unggah dienkripsi, namun administrator website tentu juga memiliki kode etik tersendiri sehingga data data yang sekiranya bersifat rahasia maka akan di enkripsi sehingga data yang kita unggah benar benar terjaga kerahasiaannya.

## 2. LANDASAN TEORI

### 2.1 Kriptografi

Kriptografi, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita [Bruce Schneier - Applied Cryptography]. Selain pengertian tersebut terdapat pula pengertian ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data [A. Menezes, P. van Oorschot and S. Vanstone - Handbook of Applied Cryptography]. Tidak semua aspek keamanan informasi ditangani oleh kriptografi.

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

1. Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
2. Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
3. Autentikasi, adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan

melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.

4. Non-repudiasi., atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

algoritma sandi adalah algoritma yang berfungsi untuk melakukan tujuan kriptografis. Algoritma tersebut harus memiliki kekuatan untuk melakukan (dikemukakan oleh Shannon):

1. konfusi/pembingungan (confusion), dari teks terang sehingga sulit untuk direkonstruksikan secara langsung tanpa menggunakan algoritma dekripsinya
2. difusi/pelebaran (difusion), dari teks terang sehingga karakteristik dari teks terang tersebut hilang.

sehingga dapat digunakan untuk mengamankan informasi. Pada implementasinya sebuah algoritmas sandi harus memperhatikan kualitas layanan/Quality of Service atau QoS dari keseluruhan sistem dimana dia diimplementasikan. Algoritma sandi yang handal adalah algoritma sandi yang kekuatannya terletak pada kunci, bukan pada kerahasiaan algoritma itu sendiri. Teknik dan metode untuk menguji kehandalan algoritma sandi adalah kriptanalisa.

Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi antara dua himpunan yaitu yang berisi elemen teks terang /plaintext dan yang berisi elemen teks sandi/ciphertext. Enkripsi dan dekripsi merupakan fungsi transformasi antara himpunan-himpunan tersebut. Apabila elemen-elemen teks terang dinotasikan dengan P, elemen-elemen teks sandi dinotasikan dengan C, sedang untuk proses enkripsi dinotasikan dengan E, dekripsi dengan notasi D.

Enkripsi :  $E(P) = C$

Dekripsi :  $D(C) = P$  atau  $D(E(P)) = P$

Secara umum berdasarkan kesamaan kuncinya, algoritma sandi dibedakan menjadi :

1. kunci-simetris/symetric-key, sering disebut juga algoritma sandi konvensional karena umumnya diterapkan pada algoritma sandi klasik
2. kunci-asimetris/asymetric-key

Berdasarkan arah implementasi dan pembabakan jamannya dibedakan menjadi :

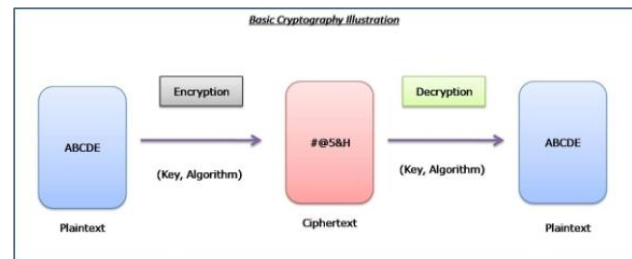
1. algoritma sandi klasik classic cryptography
2. algoritma sandi modern modern cryptography

Berdasarkan kerahasiaan kuncinya dibedakan menjadi :

1. algoritma sandi kunci rahasia secret-key
2. algoritma sandi kunci publik publik-key

Pada skema kunci-simetris, digunakan sebuah kunci rahasia yang sama untuk melakukan proses enkripsi dan

dekripsinya. Sedangkan pada sistem kunci-asimetris digunakan sepasang kunci yang berbeda, umumnya disebut kunci publik(public key) dan kunci pribadi (private key), digunakan untuk proses enkripsi dan proses dekripsinya. Bila elemen teks terang dienkripsi dengan menggunakan kunci pribadi maka elemen teks sandi yang dihasilkannya hanya bisa didekripsikan dengan menggunakan pasangan kunci pribadinya. Begitu juga sebaliknya, jika kunci pribadi digunakan untuk proses enkripsi maka proses dekripsi harus menggunakan kunci publik pasangannya.



**Gambar 2-1 Ilustrasi dalam teknik kriptografi**

Beberapa algoritma yang dikenal luas dan masih digunakan hingga sekarang antara lain:

- CBC (Cipher Block Chaining)
- DES (Data Encryption Standard)
- AES (Advanced Encryption Standard)
- Diffie-Hellman
- RSA (Rivest, Shamir, Adleman)
- dll.

## 2.2 HTML

HyperText Markup Language (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah Penjelajah web Internet dan formatting hypertext sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan kedalam format ASCII normal sehingga menjadi home page dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (Standard Generalized Markup Language), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C)

Contoh dokumen HTML sederhana  
<html>

```

<head>
  <title>' 'Selamat Malam' ' '
HTML</title>
</head>
<body>
  <p>Nama saya Adrianus!</p>
</body>
</html>

```

### 2.3 PHP

PHP: Hypertext Preprocessor[1] adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML.[2][3] PHP banyak dipakai untuk memrogram situs web dinamis. PHP dapat digunakan untuk membangun sebuah CMS.

Program Hello World yang ditulis menggunakan PHP adalah sebagai berikut:

```

<?php
  echo "Hello World";
?>

```

Dalam Php sendiri menyediakan fungsi yang dapat di gunakan untuk proses enkripsi standart, antara lain :

1. FUNGSI CRC32()
 

Fungsi ini digunakan untuk menghitung polinomial crc32 terhadap argumennya, yaitu untuk menguji validasi integritas data yang dikirim dari satu tempat ke tempat lain melalui peranti komunikasi. Fungsi ini akan menghasilkan data bertipe integer.

Bentuk : crc32(string);
2. FUNGSI CRYPT()
 

Fungsi yang ini menggunakan metode DES ( Data Encryption Standard ) dalam mengolah data string yang dienkripsi. Untuk penjelasan DES , wajib baca ini.

Bentuk : crypt(string str,string salt);
3. FUNGSI MD5()
 

Untuk membuat data enkripsi, juga dapat menggunakan fungsi md5(). Fungsi ini akan menghasilkan data acak dan kombinasi antara huruf dan angka sebanyak 32 karakter. Kebanyakan fungsi ini digunakan para programmer PHP.

Bentuk : Md5(string);

### 2.4 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

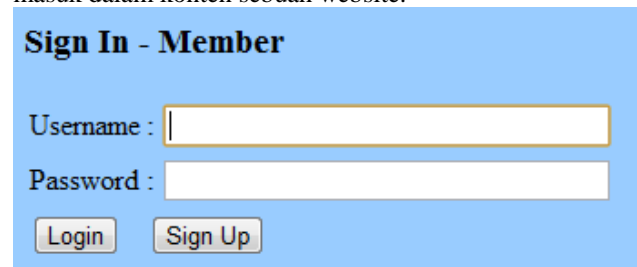
Contoh syntax MySQL :

- `select * from buktamu`

- `insert into buktamu values ('Aa','Ezha','Palembang','21')`

### 3. PEMBAHASAN APLIKASI

Aplikasi di mulai dari form login. Di ibaratkan sebuah website yang mengharuskan user untuk login untuk dapat masuk dalam konten sebuah website.



Gambar 3.1 Form Login

Seolah olah user belum mendaftar dalam website tersebut, maka pertama harus melakukan pendaftaran / sign up. Berikut merupakan contoh aplikasi sederhana untuk sign up / pendaftaran di suatu alamat web tertentu belum memanfaatkan kriptografi dalam implementasinya :



Gambar 3-2 Form Pendaftaran

Sepintas tidak ada masalah dalam form diatas, pada text field untuk isian password sudah menunjukkan kerahasiaan data user. Pada kasus ini pada text field Password kita isi "GAWAT123". Setelah user klik button daftar, maka ada pemberitahuan bahwa data telah berhasil di simpan dalam bentuk ucapan terima kasih.

Sekarang pertanyaannya bagaimana keadaan data yang kita unggah di dalam database server. Apakah data yang kita kirimkan bisa terjaga kerahasiaannya. Berikut kondisi data yang di terima oleh server.

id	nama	password	ktp	um
7	Joko Sulisty	GAWAT123	31.2312.655436.0001	35
6	Whilda Chaq	ABANZMIF	0980828619284721	20

Gambar 3-3 Screenshoot pada database

Pada kolom password data tersimpan polos tanpa ada perubahan dari yang user ketikkan. sehingga admin sebuah web dapat mengetahui password user dan kerahasiaan data user sudah tidak bisa terjamin lagi.

Setelah mempelajari celah yang di dapat, kemudian memasukkan ilmu kriptografi dalam code PHP.

**Gambar 3-4 Form Pendaftaran setelah penambahan fungsi enkripsi**

Jika dibandingkan dengan form sebelumnya memang tidak terdapat perubahan. Namun setelah kita klik button daftar. Data yang bersifat rahasia akan di enkripsi sehingga pada database tetap menjaga kerahasiaan data dari pengguna. Pada kasus ini text field pada password di isi “315757”, maka hasil enkripsinya dapat dilihat seperti gambar di bawah.

id	nama	password	ktp	umur
7	Joko Sulistyio	GAWAT123	31.2312.655436.0001	39
6	Whilda Chaq	ABAN2MIF	0980828619284721	20
8	E. B. Taqwadasbriliani	14o076iBq.Ijc	33.2201.234667.0002	19

**Gambar 3.5 Screenshoot database setelah memasukkan fungsi enkripsi**

Pada kolom password dari data yang memiliki nama E.B. Taqwadasbriliani sudah terenkripsi sehingga data dari masukan user yang bersifat rahasia tetap terjaga kerahasiaanya.

Pada dasarnya pemanfaatan ilmu kriptografi pada suatu aplikasi memang tidak mempengaruhi alur cara kerja aplikasi itu sendiri. Melainkan hanya data yang bersifat rahasia dapat di jaga kerahasiaanya.

Setelah mendaftar barulah kita mencoba untuk login ke dalam website yang kita tuju.

**Gambar 3.6 Form Login**

Setelah mendaftar, untuk login ke dalam website tersebut cukup memasukkan data yang kita input waktu mendaftar. Yang perlu di perhatikan bahwa password yang kita gunakan tetap menjadi rahasia.

Berikut tampilan aplikasi ketika berhasil untuk login

**Gambar 3-7 Tampilan aplikasi ketika berhasil login**

Setelah berhasil login biasanya aplikasi akan mendirect sebuah alamat konten dari web aplikasi itu sendiri. sebagai contoh : facebook, yahoo, dan masih banyak lagi yang lainnya.

#### 4. SOURCE CODE DAN DATABASE YANG DI GUNAKAN

Karena aplikasi yang sebenarnya dapat di akses banyak orang, maka di butuhkan satu tempat untuk menampung seluruh data-data dalam suatu sistem aplikasi. Disinilah database berperan. Dalam contoh aplikasi ini hanya cukup 1 database dan 1 tabel di dalamnya. Karena memang tidak memerlukan kapasitas database yang besar. Berikut design tabel yang digunakan :

**Tabel 4-1 Design Tabel database yang digunakan**

Name	Type	Null	Default	Extra
id (PK)	Int(10)	No		A_I
nama	varchar(50)	Yes	NULL	
password	varchar(50)	Yes	NULL	
Ktp	varchar(50)	Yes	NULL	
umur	varchar(50)	Yes	NULL	
alamat	varchar(150)	Yes	NULL	

Ket : PK → Primary Key

A\_I → Auto Increment

Aplikasi web ini di bangun dengan bahasa pemrograman html dan php. Untuk mengkombinasikan dengan database perlu memasukan sedikit syntax MySQL untuk memasukkan data dan me-retrieve data. Berikut penjelasan mengenai file file yang di digunakan dalam makalah ini.

Tabel 4-2 Keterangan file

No.	Nama File	Keterangan
1.	login.php	Form login
2.	proseslogin.php	Proses username dan password
3.	register.php	Form pendaftaran
4.	prosesregister.php	Proses data user di masukkan ke database
5.	listmember.php	Untuk menampilkan seluruh member yang terdaftar

#### 4.1 login.php

```
<html>
<head>
<title>Form - Bukutamu</title>
</head>
<body bgcolor="#99CCFF">
<h3>Sign In - Member</h3>
<form action="proseslogin.php"
method="post">
<table border=0>
<tr><td>Username</td><td>: <input
type="text" name="username"
size="35"></td></tr>

<tr><td>Password</td><td>: <input
type="password" name="password"
size="35"></td></tr>
<tr><td><input
type="submit"
value="Login"></td><td><a
href="register.php"><input
type="button" value="Sign
Up"></a></td></tr>
</table>
</form>
</body>
</html>
```

#### 4.2 proseslogin.php

```
<html>
<head>
<title>Vaidation -
Bukutamu</title>
</head>

<body bgcolor="#99CCFF">
<h3>Sign In - Member</h3>
<?php

if($_POST["username"]||$_POST["pass
word"]){
require("koneksi.php");
$link=open connection();
$tablename="member";
$username=$_POST["username"];
$password=crypt($_POST["password
"],14);
$sqlstr="select nama,password
from member where
nama='$username'";
```

```
$result = mysql_query ($sqlstr)
or die ("Invalid query");
if($row = mysql_fetch_object
($result)){

$passwordddb = $row->password;

if($passwordddb==$password){
echo("SELAMAT LOGIN
BERHASIL... ");
}else{
echo("PASSWORD / USERNAME
SALAH !!");}
}else{
echo("Username tidak
ditemukan");
}
mysql_close($link);
}
?>
<ul>
<li><a href="listmember.php">View
Member</a></li>
<li><a href="login.php">Form
Login</a></li></ul>
</body>
</html>
```

#### 4.3 register.php

```
<html>
<head>
<title>Form - Bukutamu</title>
</head>
<body bgcolor="#99CCFF">
<h3>Form Daftar Member Baru</h3>
<form
action="prosesregister.php"
method="post">
<table border=0>
<tr><td>Nama</td><td>: <input
type="text" name="nama"
size="35"></td></tr>
<tr><td>Password</td><td>:
<input type="password"
name="password"
size="35"></td></tr>
<tr><td>No. KTP</td><td>: <input
type="text" name="ktp"
size="35"></td></tr>
<tr><td>Umur</td><td>: <input
type="text" name="umur"
size="35"></td></tr>
<tr><td>Telepon</td><td>: <input
type="text" name="telepon"
size="35"></td></tr>
<tr><td>Alamat</td><td><textarea
name="alamat" rows=5
cols=30></td></tr>
<tr><td><input type="submit"
value="Daftar"></td><td><input
type="reset"
value="reset"></td></tr>
</table>
</form>
<a href="listmember.php">View
guestbook</a>
```

```

    </body>
</html>

```

#### 4.4 prosesregister.php

```

<html>
  <head>
    <title>Vaidation -
Bukutamu</title>
  </head>
  <body bgcolor="#99CCFF">
    <h3>SignUp - GuestBook</h3>
    <?php

if($_POST["nama"]||$_POST["password"]||$alamat=$_POST["alamat"]||
$_POST["ktp"]||$_POST["umur"]||
$_POST["telepon"])
    {

require("koneksi.php");

$link=open_connection();
    $tablename="member";

$nama=$_POST["nama"];

$password=crypt($_POST["password"],14);

$noktp=$_POST["ktp"];

$umur=$_POST["umur"];

$telepon=$_POST["telepon"];

$alamat=$_POST["alamat"];
    $sqlstr="INSERT INTO
$tablename
VALUES ('', '$nama', '$password', '$
noktp', '$umur', '$telepon', '$alamat')";

    if (!mysql_query
($sqlstr)) {
        echo ("Invalid
Query!<br>Please register
again...<br>");
        exit;
    }
    mysql_close($link);
    echo ("Thank you for
signing up...");
    }

?>

    <ul>
    <li><a
href="listmember.php">View Guest
Book</a></li>
    <li><a
href="login.php">Form
Login</a></li>
    </ul>
  </body>
</html>

```

#### 4.5 listmember

```

<html>
<head><title>View -
GuestBook</title></head>
<body bgcolor="#99CCFF">
  <h3>View GuestBook</h3>
  <hr size=1 width=100%>
  <?php
    require("koneksi.php");
    $link=open_connection();
    $tablename="member";
    $sqlstr="SELECT * FROM
$tablename ORDER BY id DESC";
    $result = mysql_query
($sqlstr) or die ("Invalid
query");
    while ($row =
mysql_fetch_object ($result)) {
        $nama=$row->nama;
        $password=$row->
password;
        $ktp=$row->ktp;
        $umur=$row->umur;
        $telp=$row->telp;

$alamat=stripslashes($row->
alamat);
        echo ("<b>Nama </b>:
$nama<br>");
        echo ("<b>Password
</b>: $password<br>");
        echo ("<b>No. KTP
:</b> $ktp<br>");
        echo ("<b>Umur :</b>
$umur<br>");
        echo ("<b>Telepon
:</b>$umur<br>");
        echo ("<b>Alamat
:</b>$alamat<br>");
        echo ("<hr size=1
width=100%>");
    }
    mysql_close($link);
    ?>
    <a href="login.php">Form
- Login</a>
  </body>
</html>

```

Fungsi enkripsi yang digunakan dalam aplikasi ini yaitu fungsi standart yang sudah di sediakan oleh bahasa pemrograman PHP itu sendiri yaitu fungsi crypt(), dalam bentuknya crypt(string1,string2). String1 berupa plaintext yang ingin di enkripsi menjadi chipertext. String2 berguna untuk sebagai key dalam fungsi crypt itu sendiri. Dengan string1 yang sama akan mendapatkan hasil chipertext yang berbeda jika menggunakan string2 yang berbeda pula.

Password yang sudah di enkripsi tidak perlu di dekripsikan lagi karena untuk proses login, password dari user akan di enkripsi lagi dan di dibandingkan dengan chipertext yang terdapat di dalam database server. Dengan demikian kerahasiaan password user masih tetap terjaga.



## 5. LINGKUNGAN HARDWARE DAN SOFTWARE PENDUKUNG YANG DI GUNAKAN

### 5.1 Hardware

Perangkat keras yang terlibat dalam pembuatan makalah ini antara lain :

- a. Laptop Toshiba L645
  1. Processor : Intel(R) Core i3 CPU 2.40Ghz
  2. Memory : 2048MB RAM
  3. VGA : ATI Radeon HD5470 1236MB
- b. Mouse Micropack

### 5.2 Software

Perangkat lunak pendukung dalam pembuatan makalah ini antara lain :

- a. Operating System : Windows 7 Home Premium 32-bit
- b. Text Editor : Notepad++
- c. Server configuration :
  1. WampServer 2.0
  2. Apache version 2.2.11
  3. PHP version 5.2.9-2
  4. MySQL version 5.1.33
- d. Administrator basis data
  1. phpMyAdmin
  2. HeidiSQL
- e. Web Browser : Google Chrome
- f. Programming language : HTML, PHP, MySQL

## 6. KESIMPULAN

Selama pengerjaan makalah ini penulis dapat menyimpulkan beberapa hal, antara lain :

1. Kriptografi memiliki peranan penting terhadap kerahasaan data.
2. Kerumitan algoritma enkripsi memengaruhi tingkat keamanan plaintext yang di enkripsi.
3. Secara realistis dalam dunia kriptografi, tidak ada algoritma kriptografi yang aman 100%. Arti aman adalah data yang di enkripsi tidak dapat di terjemahkan orang lain tanpda seizin pembuat.
4. Untuk lebih mengamankan suatu data yang di enkripsi, sebaiknya di bedakan antara algoritma enkripsi dan dekripsi.
5. Fungsi enkripsi yang sudah menjadi bawaan bahasa pemrograman PHP dapat menjadi alternatif untuk men-enkripsi data namun tidak di anjurkan karena fungsi bawaan memiliki banyak kelemahan dan mudah untuk di "hack".

## 7. DAFTAR PUSTAKA

- [1] <http://id.wikipedia.org/wiki/Kriptografi> (diakses pada 10 desember 2011 pukul 11.35 WIB)
- [2] [http://id.wikipedia.org/wiki/HyperText\\_markup\\_language](http://id.wikipedia.org/wiki/HyperText_markup_language)(diakses pada 10 desember 2011 pukul 12.22)
- [3] <http://id.wikipedia.org/wiki/PHP>(diakses pada 10 Desember 2011 pukul 12.33)

- [4] <http://codeindesign.com/dasar-kriptografi-enkripsi-dan-dekripsi/> (diakses pada 11 Desember 2011 pukul 10.02)

## 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, 11 Desember 2011



Whilda Chaq - 13511601