ENKRIPSI FIELD TABEL DATABASE DENGAN PGP

M. Didik R. Wahyudi,
Jurusan Teknik Informatika, Fakultas Teknologi Industri
Institut Sains & Teknologi AKPRIND Yogyakarta
Jl. Kalisahak No.28 Komp.Balapan, Yogyakarta 55222

ABSTRAK

The development of electronic data is increasing rapidly, causing the security of electronic data need a fairly powerful. Currently, each person is very easy to exchange information on all matters, including sharing knowledge on how to access the data illegally. Storage of data in the database table is equipped with the password authentication mechanism which is installed on the application can be made. But bad people who intend to seek another way to access the confidential data directly to the database table, without going through the application.

With the possibility of illegal access the databse directly to the table, the need to develop a mechanism that can secure the data from illegal access of people are not responsible for the database table level. There are many ways you can do for the encryption of data stored in the database table.

On this research, suggested techniques using database encrypting by PGP is promoted. PGP is an encryption mechanism that is very good practice with the public and private key. PGP is well known for encryption in e-mail and text files, and some other features. Important data which you want stored in the database field, previously encrypted with PGP practice. Once encrypted, barulan ciphertext is stored in the database table field.

Keywords: Data Encryption, PGP, Security Database

INTISARI

Perkembangan data elektronik yang semakin pesat, mengakibatkan dibutuhkannya pengamanan data elektronik yang cukup handal. Saat ini, setiap orang sangat mudah bertukar informasi mengenai segala hal, termasuk berbagi ilmu mengenai bagaimana caranya mengakses data secara ilegal. Penyimpanan data dalam tabel database yang dilengkapi dengan mekanisme autentikasi password yang dipasang pada aplikasi dapat dilakukan. Namun orang yang berniat jahat dapat mencari jalan lain untuk mengakses data rahasia tersebut, yaitu dengan mengakses langsung ke tabel database, tanpa melalui aplikasi.

Dengan adanya kemungkinan akses ilegal yang dilakukan langsung ke tabel databse tersebut, maka perlu dikembangkan suatu mekanisme yang dapat mengamankan data dari akses ilegal yang dilakukan orang tidak bertanggung jawab pada level tabel database. Ada banyak cara yang bisa dilakukan untuk enkripsi data yang disimpan dalam tabel database.

Pada penelitian ini, penulis mencoba mengusulkan teknik enkripsi tabel database dengan mempergunakan PGP. PGP merupakan mekanisme enkripsi yang sangat baik dengan mempergunakan *public* dan *private key*. PGP lebih dikenal untuk enkripsi pada e-mail dan file teks, serta beberapa fitur lain. Data penting yang hendak disimpan dalam field database, sebelumnya dienkripsi dengan mempergunakan PGP. Setelah dienkripsi, barulan *ciphertext* ini disimpan dalam field tabel database.

Kata kunci : Enkripsi data, PGP, Pengamanan Database

PENDAHULUAN

Alvin Toffler dalam bukunya *The Third Wave* (1984) telah memprediksikan bahwa di era milenium ketiga, teknologi akan memegang peranan yang signifikan dalam kehidupan manusia. Perkembangan ilmu pengetahuan dan teknologi modern ini akan mengimplikasikan berbagai perubahan dalam kinerja manusia.

Salah satu produk inovasi teknologi telekomunikasi adalah internet *(interconection networking*) yaitu suatu koneksi antar jaringan komputer. Aplikasi internet saat ini telah memasuki berbagai segmen aktivitas manusia, baik dalam sektor politik, sosial, budaya, maupun ekonomi dan bisnis.

Seiring dengan perkembangan teknologi informasi berbasis internet dan dijadikannya internet sebagai sarana pengolahan data, masalah keamanan data menjadi topik utama. Berdasarkan *CSI/FBI Computer Crime and*

Security Survey 2005, mayoritas perusahaan memberikan anggaran untuk pengembangan sistem dan keamanan data sekitar dibawah 5% dari total anggaran perusahaan tersebut. Salah satu contoh masalah keamanan komputer yang pernah terjadi di Amerika terjadi pada Maret 2005. Seorang mahasiswi dari UCSB dituduh melakukan kejahatan mengubah data-data nilai ujiannya (dan beberapa mahasiswa lainnya). Dia melakukan hal tersebut dengan mencuri identitas dua orang profesor (Gordon,2005).

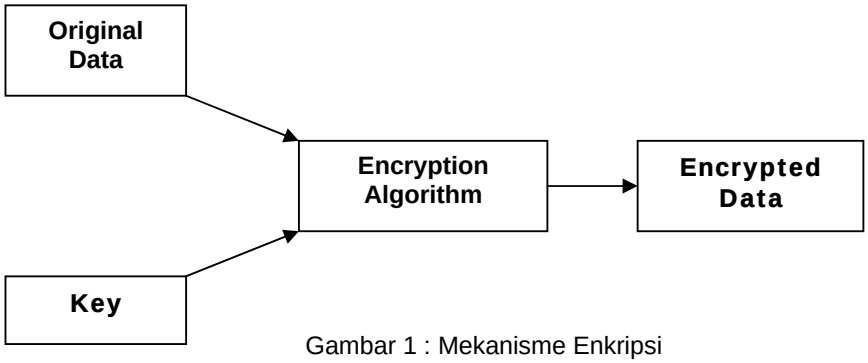
Di Indonesia, ada beberapa kasus sehubungan dengan kejahatan komputer. Seorang cracker Indonesia (yang dikenal dengan nama hc) tertangkap di Singapura ketika mencoba menjebol sebuah perusahaan di Singapura. September dan Oktober 2000. Setelah berhasil membobol bank Lippo, kembali Fabian Clone beraksi dengan menjebol web milik Bank Bali. Perlu diketahui bahwa kedua bank ini memberikan layanan Internet banking. Hasil poling yang dilakukan Majalah Warta Ekonomi secara online tahun 2001 menyatakan bahwa dari 75 pengunjung, 37% mengatakan meragukan keamanan transaksi secara online, 38% meragukannya, dan 27% merasa aman. Masih ditahun yang sama, Polda DIY meringkus seorang *carder* Yogya. Tersangka yang masih mahasiswa diringkus di Bantul dengan barang bukti sebuah paket yang berisi lukisan (Rumah dan Orang Indian) berharga Rp 30 juta. (Gordon,2005)..

Kebutuhan akan keamanan database timbul dari kebutuhan untuk melindungi data. Pertama, dari kehilangan dan kerusakan data. Kedua, adanya pihak yang tidak diijinkan hendak mengakses atau mengubah data. Permasalahan lainnya mencakup perlindungan data dari *delay* yang berlebihan dalam mengakses atau menggunakan data, atau mengatasi gangguan *denial of service*.

Kontrol akses terhadap terhadap informasi yang sensitif merupakan perhatian terutama oleh manajer, pekerja di bidang informasi, *application developer*, dan DBA. Kontrol akses selektif berdasarkan authorisasi keamanan dari level user dapat menjamin kerahasiaan tanpa batasan yang terlalu luas. Level dari kontrol akses ini menjamin rahasia informasi sensitif yang tidak akan tersedia untuk orang yang tidak diberi ijin (authorisasi) bahkan terhadap user umum yang memiliki akses terhadap informasi yang dibutuhkan, kadang-kadang pada tabel yang sama.

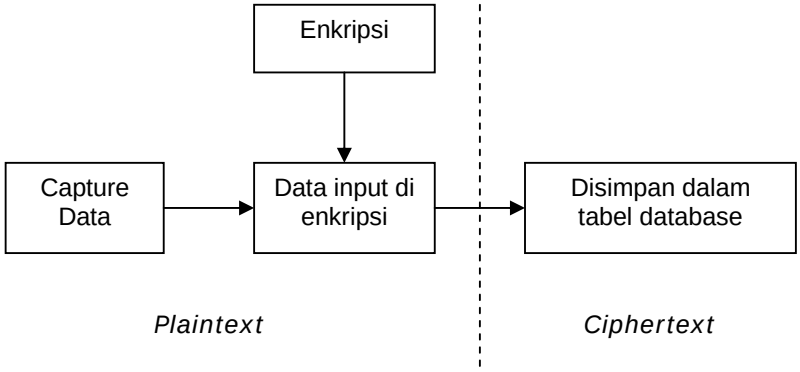
Mengijinkan informasi dapat dilihat atau digunakan oleh orang yang tidak tepat dapat menyulitkan, merusak, atau membahayakan individu, karir, organisasi, agensi, pemerintah, atau negara. Namun untuk data tertentu seringkali bercampur dengan data lainnya, yaitu pada informasi yang kurang sensitif yang secara legal dibutuhkan oleh berbagai user. Membatasi akses terhadap semua table atau memisahkan data sensitive ke database terpisah dapat menciptakan lingkungan kerja yang tidak nyaman yang membutuhkan biaya besar pada hardware, software, waktu user, dan administrasi.

Pengamanan data yang paling dasar adalah dengan cara memasang firewall. Semua akses yang akan dilakukan di server harus melewati firewall ini. Beberapa aturan dapat dipasang pada konfigurasi firewall, seperti terminal mana yang boleh masuk dan mengakses data pada sever atau melakukan perubahan data tertentu pada database server. Pengamanan dengan firewall saja belum cukup untuk mengamankan suatu data penting. Penyusup/*cracker* dapat melakukan penyusupan/eksploitasi keamanan dengan mempergunkaan teknik tertentu, sehingga dapat mengakses data rahasia yang sebenarnya telah diamankan sehingga dapat memperoleh suatu informasi dengan cara langsung mengakses tabel database, kemudian memprosesnya dengan metode tertentu tanpa melalui program aplikasi. Apabila hal ini terjadi maka sebaiknya data yang disimpan dalam database sebaiknya juga “diamankan” dengan mempergunakan teknik tertentu, sehigga walaupun data tersebut dapat diambil oleh orang yang tidak berhak, maka data tersebut tidak mempunyai arti karena dibutuhkan suatu cara untuk menerjemahkan isi data tersebut. Salah satu cara yang dapat dipergunakan untuk mengamankan data agar tidak mudah dibaca oleh orang yang tidak berhak adalah dengan cara melakukan penyandian terhadap data penting yang disimpan dalam database(Nada,2005). Sebagai contoh terdapat sebuah data dengan nomor account 2 dengan saldo 150000. Data ini dapat disimpan dengan cara menggabung kedua angka tersebut dengan tanda tertentu menjadi150000.2. Sehingga makna yang sesungguhnya dari data tersebut hanya diketahui oleh database administrator. Teknik ini disebut *encryption*. Untuk mengembalikan data tersebut sebagaimana makna yang sesungguhnya dapat dilakukan dengan membuat sebuah *script* untuk mengembalikan nilai data yang sesungguhnya. Proses pengembalian data yang disandikan ini disebut *decryption*. Berikut ini adalah mekanisme *encription* data :



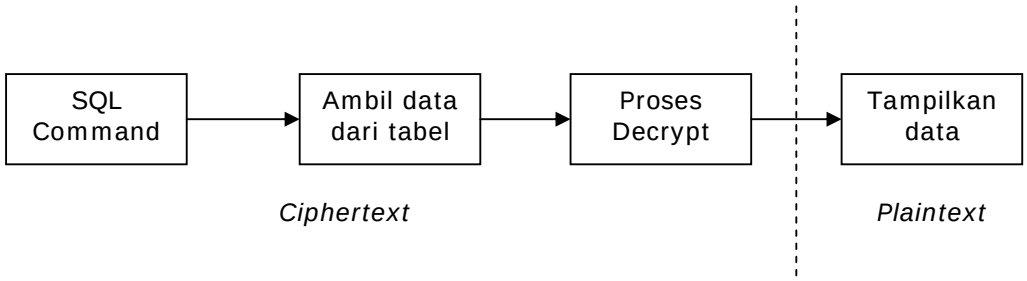
Gambar 1 : Mekanisme Enkripsi

Sebuah data sumber (*plaintext*) yang akan disandikan (di-*encrypt*) diproses dengan *encryption algorithm* dengan mempergunakan kunci yang sudah ditetapkan, sehingga menjadi data ter-*encrypt (ciphertext)*. Proses ini dilakukan sebelum data disimpan dalam database.



Gambar 2 : proses enkripsi data dan penyimpanan data dalam tabel database

Proses pembalikan data dari *ciphertext* ke *plaintext* dapat dilakukan dengan cara pembalikan proses enkripsi. Proses ini dilakukan ketika data sudah diambil dari database dan sebelum data ditampilkan/diproses, sehingga data yang ditampilkan/diproses adalah data yang sudah di-*decrypt (plaintext)*.



Gambar 3 : proses *decrypt* data untuk menangani permintaan penampilan data

PEMBAHASAN

Pada saat ini ada beberapa cara enkripsi data, seperti dengan MD5 yang sangat populer pada aplikasi berbasis web dan PGP yang sering dipakai untuk enkripsi data dan e-mail. PGP memiliki kelebihan dengan kemampuannya untuk enkripsi dengan mempergunakan *publik/private key cryptosystem.* Pada penelitian ini, akan dipergunakan teknik enkripsi dengan

mempergunakan PGP yang berjalan dibawah perintah *shell* DOS.

Untuk melakukan proses enkripsi dan dekripsi data, PGP yang berjalan dibawah *shell* DOS ini akan bekerja sama dengan PHP yang berperan sebagai perantara untuk menyimpan data dari suatu *field* tabel database yang akan disimpan dalam tabel database. Aplikasi ini berjalan pada sistem operasi berbasis Windows. Berikut ini perintah enkripsi data yang disimpan dalam file teks dengan nama pln.txt dengan mempergunakan PGP yang berjalan dibawah shell DOS dan hasil enkripsi diletakkan kedalam file teks chp.txt :

C:\PGP>pgpe -c -aftz -o pln.txt > chp.txt

Pada perintah enkripsi diatas, apabila file teks pln.txt berisi sebuah nilai string "555453786432", maka file chp.txt yang merupakan hasil enkripsi akan menghasilkan nilai string :

```
-----BEGIN PGP MESSAGE---
Version: PGPfreeware 5.0i for non-
commercial use
MessageID:
ge/oBNFNAn95lAh9vZR3UFZsvujwYIA0

pCsMgO6Xvmsm8A5yG61wHXfw0ZPOj
6TpzN3RoWIt+RueQSphF2rG3Sx0R4U
E
=h1FP
-----END PGP MESSAGE-----
```

Berikut ini perintah yang dipergunakan untuk mengembalikan *ciphertext* ke dalam *plaintext*. File yang berisi *ciphertext* diberi nama chp.txt dan file hasil dekripsi disimpan dalam file yang diberi nama pln.txt.

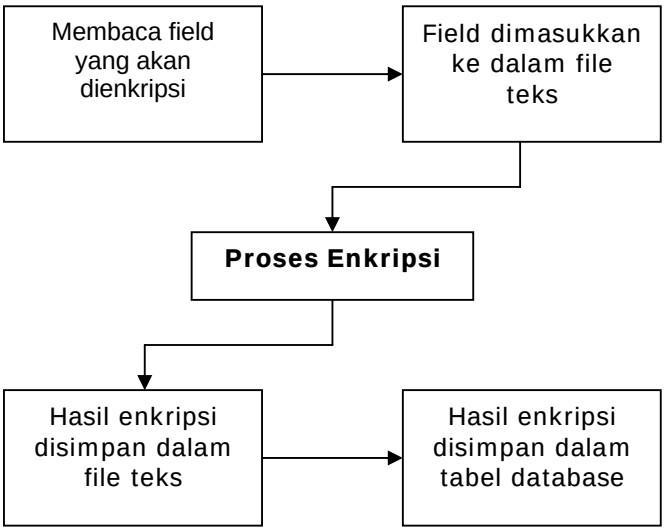
C:\PGP>pgpv -fz -o chp.txt > pln.txt

Pada perintah enkripsi dan dekripsi diatas, proses *encrypt* dan *decrypt*, dilakukan dengan mempergunakan pgp tanpa mempergunakan *private/public key* yang menjadi ciri khas dari PGP. Hal ini dikarenakan, apabila mempergunakan *private/public key*, maka akan ada beberapa parameter yang harus diisikan, menyertai eksekusi perintah tersebut. Hasil eksekusi perintah *decrypt* diatas akan mengembalikan *ciphertext* kedalam *plaintext*.

Enkripsi dan dekripsi *field database* dengan mempergunakan PGP

Proses enkripsi dan dekripsi isi field dari suatu tabel database, disimulasikan dengan studi kasus berikut ini. Suatu perusahaan bermaksud menyimpan suatu data rahasia yang hanya bisa diketahui isi yang sebenarnya dengan suatu *script* atau algoritma tertentu. Pengaksesan data dengan cara mengakses data langsung ke tabel database tanpa melalui aplikasi, tidak diperkenankan. Proses menyimpan data kedalam mode ter-enkrip (*ciphertext*) dalam database dilakukan melalui skrip tertentu. Begitu juga proses menerjemahkan data ter-enkrip (*ciphertext*) kedalam bentuk text biasa (*plaintext*) juga dilakukan dengan bantuan script tertentu.

Untuk mengakomodir kebutuhan tersebut, maka proses enkripsi dan dekripsi dilakukan dengan mempergunakan PGP. Berikut ini diagram proses yang terjadi untuk proses enkripsi field database :



Gambar 4 : proses enkripsi field tabel database

Untuk membantu proses enkripsi, maka isi field tabel database yang hendak dienkrip dimasukkan dalam suatu file teks, kemudian dienkripsi. Hasil enkripsi diletakkan kedalam file text lainnya. Selanjutnya file teks yang berisi field yang sudah dienkripsi dibaca dan isi file tersebut dimasukkan kedalam tabel database. Karena panjang karakter dari text yang sudah dienkripsi dengan PGP menjadi sangat panjang (kurang lebih 250 karakter), maka lebar field tabel database yang akan diisikan text yang sudah dienkripsi harus disesuaikan. Berikut ini script PHP yang berfungsi untuk melakukan tugas diatas :

1. Berikut ini script yang berfungsi untuk membaca field yang akan dienkripsi dari tabel dan kemudian dimasukkan dalam file teks :

```
<form action=ecr.php method=post>
<font face=verdana size=2
color=#334094>
   Nomor CC  <input type=text
name=vnocc size=25>
   <input type=submit value='OK'>
<input type=reset>
</font>
</form>
```

Berikut ini isi file ecr.php yang akan menangkap isi variabel vnocc dan disimpan kedalam file teks untuk kemudian dienkripsi :

```
$CrFile = fopen("flasl.txt","w");
if(!($CrFile))
{
  print("Error : ");
  print("'flasl.txt' gagal dibuat\n");
  exit;
}
fputs($CrFile,$vnocc);
fclose($CrFile);
```

2. Selanjutnya field yang sudah masuk dalam file teks dienkripsi dengan perintah berikut ini :

```
$outputc = shell_exec('pgpe -c -aftz -o
flasl.txt>flcrp.txt');
     echo "<pre>$outputc</pre>";
```

Perintah php shell_exec berfungsi untuk menjalankan perintah dos yang berfungsi untuk mengenkripsi file teks. Perintah echo "<pre>$outputc</pre>"; berfungsi menjalankan *script* yang sudah tersimpan di variabel $outputc yaitu : shell_exec('pgpe -c -aftz –o flasl.txt>flcrp.txt');

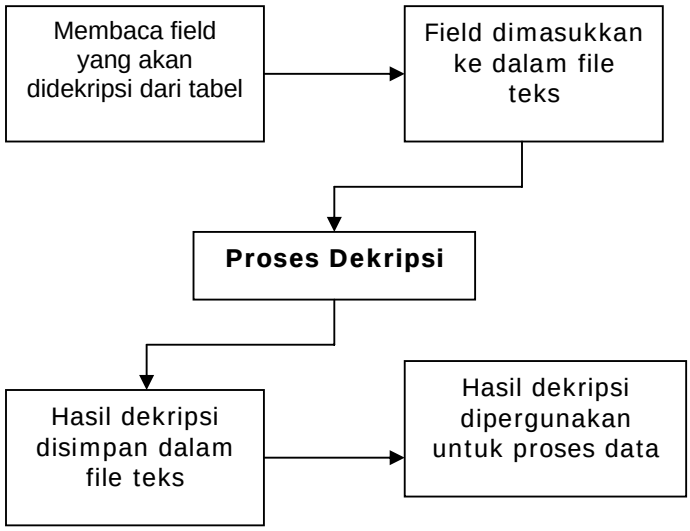
3. Field yang telah dienkrip dan disimpan dalam *file* flcrp.txt, berikutnya dimasukkan dalam tabel database. Teks enkripsi inilah yang akan dipergunakan untuk proses data. Berikut ini script yang berfungsi untuk memasukkan field terenkrip kedalam tabel database :

```
$CrFile = fopen("flcrp.txt","r");
if(!($CrFile))
{
  print("Error : ");
  print("'flcrp.txt' gagal dibuka\n");
  exit;
}
$IsiCrp = "";
while(!feof($CrFile))
{
  $IsiFl = fgets($CrFile, 255);
  $IsiCrp .= "$IsiFl";
}
fclose($CrFile);
//mengisikan ke database
$dbconect=odbc_connect("DBTeliti","","",
);
$sqlu="insert into Account (Nomor)
values ('$IsiCrp')";
$tbdata=odbc_exec($dbconect,$sqlu);
if ($tbdata)
  echo("Data berhasil disimpan");
else
        echo("Data Gagal disimpan");
```

File yang sudah dienkrip dibuka dengan perintah fopen("flcrp.txt","r");. Pembukaan file dilakukan dalam mode *read* yang ditunjukkan dengan parameter “r” pada perintah fopen. Isi file dibaca dengan script fgets($CrFile, 255); dan diletakkan kedalam variabel $IsiCrp. Proses pembacaan isi file ini berulang, sampai file tersebut sudah sampai pada akhir isi file yang ditunjukkan dengan perintah while(!feof($CrFile)) Setelah pembacaan file selesai, file ditutup dengan script fclose($CrFile);. Field yang sudah dienkrip, kemudian dimasukkan kedalam tabel database. Langkah pertama, menyambungkan dengan database dengan fungsi ODBC, yaitu : odbc_connect("DBTeliti","","");. Langkah berikutnya, dimasukkan kedalam tabel database dengan perintah : $sqlu="insert into Account (Nomor) values ('$IsiCrp')";. Perintah SQL

tersebut dieksekusi dengan perintah $tbdata=odbc_exec($dbconect,$sqlu);if ($tbdata). Apabila penambahan data sukses, maka akan muncul komentar : echo("Data berhasil disimpan");, dan apabila penambahan data gagal, maka akan muncul komentar : echo("Data Gagal disimpan");.

Untuk mengembalikan isi field tabel database yang disimpan dalam bentuk ter-enkrip, maka dilakukan proses dekripsi. Berikut ini diagram proses dekripsi atas field *ciphertext* ke dalam bentuk *plaintext.*



Gambar 5 : proses dekripsi field tabel database

Proses dekripsi dilakukan untuk memperoleh kembali isi file teks asal (*plaintext*). Untuk memperoleh *plaintext* tersebut, maka isi field tabel database yang disimpan dalam *ciphertext* dimasukkan dalam suatu file teks, kemudian didekripsi dengan mempergukanan script dekripsi. *Plaintext* hasil dekripsi disimpan dalam file text lainnya. Plaintext yang tersimpan dalam file ini kemudian dibaca dan dipergunakan untuk proses berikutnya. Berikut ini script yang berfungsi melakukan tugas diatas :

1. Langkah pertama, field yang hendak di *decrypt* dibaca dari tabel databse dan dimasukkan kedalam file teks.

```
$dbconect=odbc_connect("DBTeliti","","");
$qkategori="SELECT nocc FROM acc";
$hasil=odbc_exec($dbconect,$qkategori);
while (odbc_fetch_into($hasil, $data))
{
    $CrFile = fopen("isipln.txt","w");
        if(!($CrFile))
{
  print("Error : ");
  print("'flasl.txt' gagal dibuat\n");
  exit;
}
  fputs($CrFile,$data[0]);
  fclose($CrFile);
}
```

Proses dekripsi isi field tabel database yang disimpan dalam bentuk *ciphertext*, dilakukan dengan langkah berikut. Pertama menyambungkan dengan database yang berisi field tersebut dengan perintah : $dbconect=odbc_connect("DBTeliti","","");. Langkah berikutnya field dibaca dengan perintah SQL : $qkategori="SELECT nocc FROM acc";. $hasil=odbc_exec($dbconect,$qkategori); Isi field terenkrip dimasukkan kedalam file teks untuk *decrypt* dengan perintah : while (odbc_fetch_into($hasil, $data)){$CrFile = fopen("isipln.txt","w");

2. Field ter-*encrypt* yang sudah disimpan dalam file teks di-*decrypt* dengan mempergunakan perintah berikut ini :

```
$outputd = shell_exec('pgpv -fz -o
flcrp.txt > fldcr.txt');
echo "<pre>$outputd</pre>";
```

Perintah dos yang akan dipergunakan untuk proses dekripsi disimpan dalam

variabel $outputd yang berisi shell_exec('pgpv -fz -o flcrp.txt>fldcr.txt');. Perintah ini dieksekusi dengan script echo "<pre>$outputd</pre>";. Setelah eksekusi selesai, maka hasil dekripsi file yang terenkrip disimpan dalam file teks fldcr.txt.

3. Hasil field yang sudah di-decrypt/plaintext disimpan dalam file teks. Untuk penggunaan lebih lanjut, maka isi teks dibaca kedalam variabel dengan perintah berikut ini :

```
$DrFile = fopen("fldcr.txt","r");
if(!($DrFile))
{
  print("Error : ");
  print("'fldcr.txt' gagal dibuka\n");
  exit;
}
while(!feof($DrFile))
{
  $IsiFl = fgets($DrFile, 255);
  print("$IsiFl <br>\n");

}
fclose($DrFile);
```

File teks yang berisi *plaintext* hasil dekripsi kemudian dibaca dengan mode *read only* pada script fopen("fldcr.txt","r"); File dibaca dan diletakkan kedalam variabel $IsiFl dengan perintah fgets($DrFile, 255). Pembacaan file diulang hingga sampai pada akhir isi file dengan skrip while(!feof($DrFile)) dan file ditutup dengan skrip fclose($DrFile).

KESIMPULAN

Dari penelitian diatas, dapat disimpulkan beberapa aspek tentang teknik enkripsi field tabel database dengan mempergunakan PGP sebagai berikut :

1. Enkripsi dengan mempergunakan PGP, menghasilkan suatu hasil enkripsi yang relatif besar, sehingga penyimpanannya membutuhkan panjang field tabel databse cukup besar sekitar 200 karakter.
2. Teknik untuk enkripsi dan dekripsi mempergunakan bantuan file teks eksternal. Apabila pada waktu yang bersamaan terjadi proses enkripsi dan dekripsi lebih dari satu, perlu ada pengembangan lebih lanjut mengenai nama file yang dipergunakan, sehingga tidak terjadi pertukaran data.
3. Salah satu kelebihan PGP adalah enkripsi dengan mempergunakan kunci public. Sehingga perlu adanya pengembangan lebih lanjut tentang teknik enkripsi yang mempergunakan public dan private key. Dimana private key akan dipegang pemilik record/account tersebut.

DAFTAR PUSTAKA

Atkinson, L., 1999, “Core PHP Programming : using PHP to build dynamic Web sites”, Prentice Hall.

Bakken, S., S., Aulbach, A., etc., 1997-2001, “PHP Manual”, PHP Documentation Group.

Gordon, Lawrence A., Loeb, Martin P., Lucyshyn W., and Richardson R., 2005*,”2005 CSI/FBI Computer Crime and Security Survey”,* Computer Security Institute

Nada, Arup, January/February 2005,” Encrypt Your Data Assets”, Oracle Magazine

Raharjo, Budi, 2005, “Keamanan Sistem Informasi Berbasis Internet”, PT Insan Infonesia - Bandung & PT INDOCISC – Jakarta

Stallings, William, 1995, “*Network and Internetwork Security*,” Prentice Hall

Tim Berners-Lee, “*Weaving the Web: the past, present and future of the world wide web by its inventor*,” Texere, 2000.

Toffler, Alvin, May 1, 1984, “The Third Wave”, Bantam, Reissue edition

Zimmermann, Phill, 1998, “Security Features And Vulnerabilities”, PGP Documentation

Zimmermann, Phill, 1998, “PGP User Manual”, PGP Documentation