Jurnal Basis Data, *ICT Research Center* UNAS
Vol.6, No.1, Mei 2011 ISSN 1978-9483

PENERAPAN TEKNIK KRIPTOGRAFI STREAM - CIPHER UNTUK PENGAMAN BASIS DATA

Novi Dian Nathasia[1] & Anang Eko Wicaksono[2]
Program Studi Sistem Informasi, FTKI, Universitas Nasional, Jakarta Selatan, Indonesia
Program Studi Teknik Informatika, STMIK PPKIA Pradnya Paramita, Malang, Indonesia
Email : ivanovic77@yahoo.com arema007@yahoo.com

Abstract

An information system is build by several components. One of the component is a database. Database is a place where we store data. In a simple database management system, there is no any security feature. Therefore, we need to build some tools in order to increase the security level. For that purpose, we applying encryption to a database. In cryptography, encryption means a process to transform a plain data to a cipher text. A cipher text is an information other people could not understand. If an information which is very secret stored in a form of cipher text, then we can achieve more security level of a information system.

Keywords: cryptography, *cipher text, Database, security feature*

1. Pendahuluan

Di abad modern ini, teknologi informasi semakin berkembang sesuai dengan perannya dalam membantu pekerjaan manusia. Teknologi informasi merupakan seperangkat alat yang membantu pekerjaan yang berkaitan dengan informasi dan melakukan tugas-tugas yang berhubungan dengan pemrosesan informasi serta mendistribusikan informasi tersebut dengan menggunakan saluran komunikasi. Teknologi informasi merupakan gabungan antara teknologi komputer dan teknologi telekomunikasi dimana keduanya saling berkaitan.

Perpaduan pekerjaan manusia dengan teknologi komputer membentuk suatu sistem yang bekerja bersama-sama untuk melakukan pengolahan informasi mulai pengumpulan, pengolahan, penyimpanan sampai pendistribusian (pengiriman) informasi. Sistem paduan tersebut dikenal dengan sistem informasi. Sistem informasi pun mengalami perkembangan.

Suatu sistem informasi terdiri dari beberapa komponen pendukung. Salah satu komponen tersebut adalah basisdata (database). Database adalah tempat dimana kita menyimpan data. Pada umumnya, suatu sistem informasi mempunyai database yang dilengkapi dengan keamanan, yaitu berupa *password* bagi *administrator*. Tetapi jika *password* tersebut bisa diketahui/dipecahkan oleh orang lain maka isi database yang mungkin bersifat sangat rahasia dapat dibaca oleh orang lain yang tidak berkepentingan untuk membacanya. Untuk tujuan keamanan tersebut maka perlu dilakukan enkripsi pada database.

Enkripsi (penyandian) merupakan istilah dalam kriptografi yang berarti proses menyandikan suatu data atau informasi berbentuk teks menjadi suatu bentuk lain yang tidak dapat dipahami. Jika informasi yang tersimpan dalam database merupakan suatu bentuk informasi yang tidak dipahami oleh orang lain, tetapi hanya dipahami oleh pihak yang berkepentingan maka semakin amanlah informasi yang tersimpan tersebut. Dengan demikian, jika suatu saat keamanan yang berupa *password* suatu database sudah

terpecahkan masih tersedia keamanan tambahan bagi sistem informasi tersebut yaitu nilai (*value*) pada database yang ter-enkripsi.

2. Tinjauan Pustaka

Pengertian Kriptografi

Kriptografi (cryptography) berasal dari bahasa Yunani: "*cryptos*" artinya "*secret*" (rahasia), sedangkan "*graphein*" artinya "*writing*" (tulisan). Jadi, kriptografi adalah ilmu dan seni untuk manjaga keamanan pesan (Munir, 2006:2). Kata "seni" tersebut berasal dari fakta sejarah bahwa pada masa-masa awal sejarah kriptografi, setiap orang mungkin mempunyai cara yang unik untuk merahasiakan pesan. Cara-cara unik tersebut mungkin berbeda-beda pada setiap pelaku kriptografi sehingga setiap cara menulis pesan rahasia pesan mempunyai nilai estetika tersendiri sehingga kriptografi berkembang menjadi sebuah seni merahasiakan pesan.

Sebagai pembanding, selain definisi tersebut, terdapat pula definisi yang lain yaitu: "Kriptografi adalah ilmu yang memperlajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi."

Pada perkembangan selanjutnya, kriptografi berkembang menjadi sebuah disiplin ilmu sendiri karena teknik-teknik kriptografi dapat diformulasikan secara matematik sehingga menjadi sebuah metode yang formal. Di dalam kriptografi akan sering ditemukan berbagai istilah atau terminologi, antara lain:

- Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plaintext* atau teks jelas (*cleartext*). Pesan dapat berupa data atau informasi yang dikirim atau yang disimpan di dalam media perekaman/penyimpanan. Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan ke bentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut cipherteks (*ciphertext*) atau kriptogram (*cryptogram*). Cipherteks harus dapat ditransformasikan kembali menjadi plainteks semula agar pesan yang diterima bisa dibaca.
- Pengirim dan penerima. Pengirim adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima adalah entitas yang menerima pesan. Entitas di sini dapat berupa orang, mesin, kartu kredit dan sebagainya. Jadi orang bisa bertukar pesan dengan orang lainnya, sedangkan di dalam jaringan komputer, mesin berkomunikasi dengan mesin. Pengirim tentu menginginkan pesan dapat dikirim secara aman, yaitu ia yakin bahwa pihak lain tidak dapat membaca isi pesan yang ia kirim. Solusinya adalah dengan cara menyandikan pesan menjadi cipherteks.
- Enkripsi dan Dekripsi. Enkripsi adalah proses menyandikan plainteks menjadi cipherteks. Sedangkan proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi. Enkripsi dan dekripsi dapat diterapkan baik pada pesan yang dikirim maupun pada pesan tersimpan.
- Kriptoanalisis (*cryptoanalysis*). Kriptografi berkembang sedemikian rupa sehingga melahirkan bidang ilmu yang berlawanan yaitu kriptoanalisis. Kriptoanalisis (*cryptoanalysis*) adalah ilmu dan seni untuk memecahkan cipherteks menjadi plainteks tanpa mengetahui kunci yang digunakan. Pelakunya disebut kriptoanalis. Jika seorang kriptografer (*cryptographer*) mentransformasikan plainteks menjadi cipherteks dengan suatu algoritma dan kunci, maka sebaliknya seorang kriptoanalis berusaha untuk memcahkan cipherteks tersebut untuk menemukan plainteks atau

kunci. Pada awalnya kriptoanalisis menggunakan teknik analisis frekuensi, yaitu teknik untuk memecahkan cipherteks berdasarkan frekuensi kemunculan karakter di dalam pesan dan kaitannya dengan frekuensi kemunculan karakter di dalam alfabet. Saat ini, perkembangan komputer pun ikut membantu kegiatan kriptoanalisis. Sejarah kriptoanalisis mencatat hasil gemilang seperti penecahan Telegram Zimmermann yang membawa Amerika Serikat ke kancah Perang Dunia I, dan pemecahan cipherteks dari mesin Enigma yang mengakhiri Perang Dunia II.

- *Cipher* dan *key* (kunci). Algoritma kriptografi disebut juga *cipher* yaitu aturan untuk *enciphering* dan *deciphering*, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yaitu himpunan yang berisi elemen-elemen plainteks dan himpunan yang berisi cipherteks. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara kedua himpunan tersebut. Misalkan P menyatakan plainteks dan C menyatakan cipherteks, maka fungsi enkripsi E memetakan P ke C:

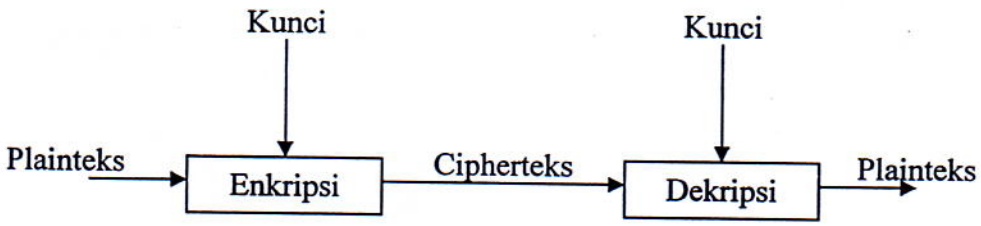
  $$E(P) = C$$

  dan fungsi dekripsi D memetakan C ke P:

  $$D(C) = P$$

Kriptografi modern yang berkembang sekarang ini menggunakan kunci yang harus dijaga kerahasiaannya. Kunci (*key*) adalah parameter yang digunakan untuk transformasi *enciphering* dan *deciphering*. Kunci biasanya berupa string atau deretan bilangan. Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi dapat ditulis sebagai:

$E_K (P) = C$ dan
$D_K (C) = P$



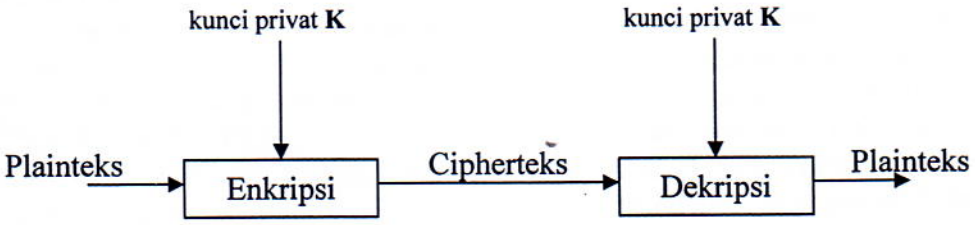
Gambar 1. Skema enkripsi dan dekripsi menggunakan kunci.

- Kriptografi membentuk sebuah sistem yang dinamakan sistem kriptografi. Sistem kriptografi (*cryptosystem*) adalah kumpulan yang terdiri dari algoritma kriptografi, semua plainteks dan cipherteks yang mungkin dan kunci.

Berdasarkan kunci yang digunakan untuk enkripsi dan dekripsi, kriptografi dapat dibedakan menjadi:xf

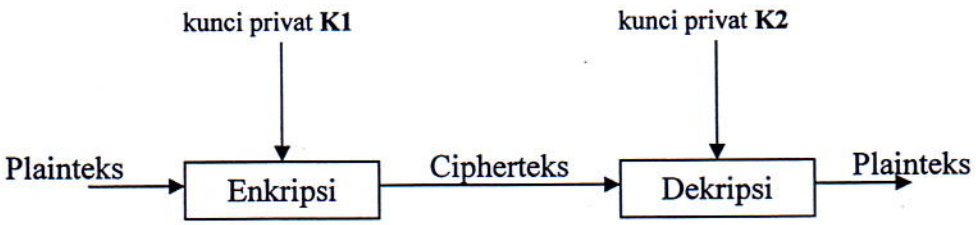
- Kriptografi kunci-simetri (kriptografi kunci privat)
  Pada sistem kriptografi kunci-simetri, kunci untuk enkripsi sama dengan kunci untuk dekripsi. Sistem kriptografi ini mengasumsikan pengirim dan penerima pesan sudah berbagi kunci yang sama sebelum bertukar pesan. Keamanan sistem kriptografi simetri terletak pada kerahasiaaan kuncinya. Aplikasi kriptografi simetri yang utama adalah melindungi kerahasiaan data yang disimpan pada media yang tidak aman. Kelemahan dari sistem ini adalah baik pengirim maupun penerima

pesan harus memiliki kunci yang sama, sehingga pengirim pesan harus mencari cara yang aman untuk memberitahukan kunci kepada penerima pesan.



Gambar 2. Skema kriptografi simetri. Kunci enkripsi sama dengan kunci dekripsi, yaitu K.

- Kriptografi kunci-asimetri (kriptografi kunci publik)
  Pada kriptografi ini, kunci untuk enkripsi tidak rahasia dan dapat diketahui oleh siapapun, sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan. Setiap orang yang berkomunikasi mempunyai sepasang kunci, yaitu kunci privat dan kunci publik. Pengirim mengenkripsi pesan dengan menggunakan kunci publik si penerima pesan. Hanya penerima pesan yang dapat mendekripsi pesan karena hanya ia yang mengetahui kunci privatnya sendiri.



Gambar 3. Skema kriptografi asimetri.

Algoritma Kriptografi Klasik

Algoritma kriptografi klasik adalah algoritma kriptografi yang berbasis karakter, yaitu enkripsi dan dekripsi dilakukan pada setiap karakter pesan. Algoritma krptografi klasik dapat dikelompokkan ke dalam dua macam *cipher*:

1. Cipher Substitusi
   Di dalam cipher substitusi setiap unit plainteks diganti dengan satu unit cipherteks. Satu unit bisa berarti satu huruf, pasangan huruf atau kelompok lebih dari dua huruf. Algoritma substitusi tertua yang diketahui adalah *Caesar Cipher* yang digunakan oleh kaisar Romawi, Julius Caesar untuk menyandikan pesan yang ia kirim kepada para gubernurnya.
   Pada *Caesar Cipher*, tiap huruf disubstitusi dengan huruf ketiga berikutnya dari susunan alfabet yang sama. Dalam hal ini kuncinya adalah jumlah pergeseran huruf yaitu 3. Susunan alfabet setelah digeser sejauh 3 huruf membentuk sebuah tabel substitusi sebagai berikut:

Plainteks:	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
Cipherteks:	**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**

Jadi, huruf A pada pleinteks disubstirusi dengan **D,** huruf B disubstitusi dengan **E,** demikian seterusnya. Dengan mengkodekan setiap huruf alfabet dengan integer secara matematis pergeseran 3 huruf alfabet ekivalen dengan melakukan operasi modulo terhadap plainteks P menjadi cipherteks C dengan persamaan:

$$C = E(P) = (P+3) \bmod 26$$

karena ada 26 huruf di dalam alfabet.

Penerima pesan mengembalikan lagi cipherteks dengan operasi kebalikan, yang secara matematis dapat dinyatakan dengan persamaan:

$$P = D(C) = (C-3) \bmod 26$$

Untuk mengenkripsi pesan yang disusun oleh 256 karakter ASCII, maka persamaan tersebut dapat diperluas menjadi:

$$C = E(P) = (P+3) \bmod 256$$

dan fungsi dekripsi adalah:

$$P = D(C) = (C-3) \bmod 256$$

2. Cipher Transposisi

Pada cipher tranposisi, huruf-huruf di dalam plainteks tetap sama, hanya saja urutannya diubah. Nama lain untuk metode ini adalah permutasi atau pengacakan karena metode yang digunakan adalah dengan cara mempermutasikan karakter-karakter yang ada dalam suatu teks.

Misalkan plainteksnya adalah:

JURUSAN TEKNIK INFORMATIKA

Untuk mengenkripsi pesan, plainteks ditulis secara horizontal dengan lebar kolom tetap, misal selebar 6 karakter (kunci k=6):

JURUSA
NTEKNI
KINFOR
MATIKA

maka ciphertkesnya dibaca secara vertikal menjadi:

JNKMUTIARENTUKFISNOKAIRA

Untuk mendekripsi, kita membagi panjang cipherteks dengan jumlah baris (4):

JNKM
UTIA
RENT
UKFI
SNOK
AIRA

Dengan membaca setiap kolom kita memperoleh pesan semula:

JURUSAN TEKNIK INFORMATIKA

Algoritma Kriptografi Modern

Kriptografi modern menggunakan gagasan dasar yang sama seperti kriptografi klasik tetapi penekanannya berbeda. Pada kriptografi klasik, kriptografer menggunakan algoritma yang sederhana yang memungkinkan cipherteks dapat dipecahkan dengan mudah, antara lain dengan penggunaan statistik, terkaan, teknik analisis frekuensi, dan

lain-lain. Algoritma kriptografi modern dibuat sedemikian kompleks sehingga kriptoanalis sangat sulit memecahkan cipherteks tanpa mengetahui kunci.

Algoritma kriptografi modern umumnya beroperasi dalam mode bit ketimbang mode karakter, seperti cipher substitusi atau cipher transposisi pada algoritma kriptografi klasik. Operasi dalam mode bit berarti semua data dan informasi baik kunci, plainteks maupun cipherteks dinyatakan dalam rangkaian bit biner 0 dan 1. Algoritma enkripsi dan dekripsi memproses semua data dan informasi dalam bentuk rangkaian bit. Rangkaian bit yang menyatakan plainteks di-enkripsi menjadi cipherteks dalam bentuk rangkaian bit, demikian sebaliknya.

Perkembangan algoritma kriptografi modern berbasis bit didorong oleh penggunaan komputer digital yang merepresentasikan data dalam bentuk biner. Sehingga algoritma kriptografi modern saat ini berbasis bit biner. Algoritma kriptografi modern berbasis bit biner. Umumnya algoritma kriptografi modern memproses data dalam bentuk blok-blok bit. Rangkaian bit yang dipecah menjadi blok-blok bit dapat ditulis dalam sejumlah cara bergantung pada panjang blok. Misalnya, plainteks 100111010110 dibagi menjadi blok bit yang panjangnya 4 menjadi

1001 1101 0110

yang dalam notasi Hexadecimal adalah

9 D 6

Operator biner yang sering dugunakan dalam cipher yang beroperasi dalam mode bit adalah exclusive-or (XOR). Operator XOR (⊕) dioperasikan dengan aturan sebagai berikut:

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

Algoritma Kunci Simetri Modern

Algoritma kunci simetri mengacu pada metode enkripsi yang dalam hal ini baik pengirim maupun penerima memiliki kunci yang sama. Algoritma kunci simetri modern beroperasi dalam mode bit dan dapat dikelompokkan menjadi dua kategori:

1. Cipher Aliran (*Stream Cipher*)

   Algoritma kriptografi beroperasi pada plainteks/ciphertkes dalam bentuk bit tunggal dimana rangkaian bit dienkripsikan/didekripsikan bit per bit. Cipher aliran mengenkripsi satu bit setiap kali.

   Pada cipher aliran, bit hanya mempunyai dua buah nilai, sehingga proses enkripsi hanya menyebabkan dua keadaan pada bit tersebut: berubah atau tidak berubah. Dua keadaan tersebut ditentukan oleh kunci enkripsi yang disebut aliran-kunci (*keystream*) yang dibangkitkan dari sebuah pembangkit-aliran-kunci (*keystream generator*). Aliran kunci di-XOR-kan dengan aliran bit-bit plainteks untuk menghasilkan aliran bit cipherteks:

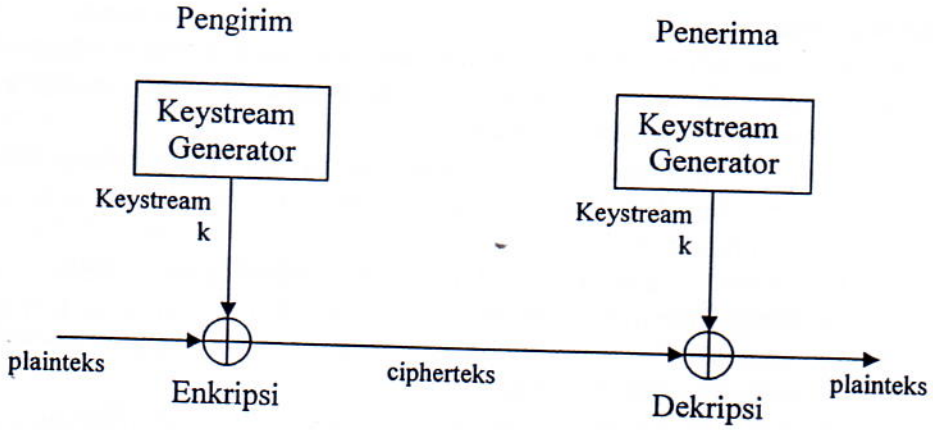
   $$C(i) = P(i) \oplus K(i)$$

   Di sisi penerima, bit-bit cipherteks di-XOR-kan dengan aliran kunci yang sama untuk menghasilkan bit-bit palinteks:

   $$P(i) = C(i) \oplus K(i)$$

   Proses enkripsi dua kali berturut-turut dengan operator XOR menghasilkan kembali plainteks semula. Gambar 4. berikut memperlihatkan konsep cipher aliran.

ISSN 1978-9483



Gambar 4. Konsep cipher aliran

Keamanan sistem cipher aliran bergantung seluruhnya pada pembangkit aliran kunci. Jika pembangkit mengeluarkan aliran kunci yang benar-benar acak (*trully random*) maka algoritma enkripsinya berada pada tingkat keamanan yang tinggi.
Cipher aliran cocok untuk mengenkripsikan aliran data yang terus menerus melalui saluran komunikasi, misalnya mengenkripsi data paada saluran yang menghubungkan antara dua buah komputer, atau mengenkripsi suara pada jaringan telepon *mobile* GSM.
Contoh algoritma yang menggunakan cipher aliran adalah RC4. RC4 adalah cipher aliran yang digunakan secara luas pada sistem keamanan seperti protokol SSL (*Secure Socket Layer*). Contoh lain adalah A5. A5 adalah cipher aliran yang digunakan untuk mengenkripsi transmisi sinyal percakapan dari standar telepon seluler GSM (*Group Special Mobile*).

2. Cipher Blok (*Blok Cipher*)
Algoritma kriptografi beroperasi pada plainteks/cipherteks dalam bentuk blok bit dimana rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya. Misalnya panjang blok adalah 64 bit, maka algoritma enkripsi memperlakukan 8 karakter setiap kali enkripsi. Cipher blok mengenkripsi satu blok bit setiap kali.
Contoh: algoritma RC5, DES, AES, IDEA, Blowfish.

Pengertian Database

Database adalah kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu (Irmansyah, 2003).
Mengapa Diperlukan Database:

- Salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
- Menentukan kualitas informasi: akurat, tepat pada waktunya dan relevan. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
- Mengurangi duplikasi data (*data redudancy*).
- Hubungan data dapat ditingkatkan.
- Mengurangi pemborosan tempat simpanan luar.

Jenjang Data:

- *Characters*, merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus (*special characters*) yang membentuk suatu item data.
- *Field*, merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari *field* membentuk suatu *record*.
- *Record*, kumpulan dari *field* membentuk suatu *record*. *Record* menggambarkan suatu unit data individu yang tertentu. Kumpulan dari record membentuk suatu *file*. Misalnya *file* personalia, tiap-tiap *record* dapat mewakili data tiap-tiap karyawan.
- *File*. *File* terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Misalnya *file* mata pelajaran berisi data tentang semua mata pelajaran yang ada.
- Database, merupakan kumpulan dari *file* atau tabel membentuk suatu database.

Fitur Basisdata

Aplikasi basisdata mengijinkan para pemakai untuk saling berhubungan dengan informasi yang disimpan di dalam basisdata (Marcus, 2002). Basisdata menyediakan struktur untuk informasi, dan mengijinkan berbagi data antar aplikasi yang berbeda. Delphi menyediakan jenis basisdata relasi (*database relational*) untuk aplikasi basisdatanya. Basisdata relasi mengorganisir informasi ke dalam tabel, yang mana berisi baris (*row/record*) dan kolom (*column/field*).

Ketika perancangan suatu aplikasi basisdata, anda harus memahami bagaimana data tersusun. Berdasar pada struktur itu, anda kemudian bisa mendisain antarmuka untuk menampilkan data kepada pemakai dan mengijinkan pemakai untuk memasukkan informasi baru atau memodifikasi data yang ada.

Pada Delphi anda dapat menggunakan basisdata yang beragam. Delphi menyediakan banyak komponen untuk mengakses basisdata tersebut. Komponen pada halaman Data Access, halaman ADO, atau halaman Interbase dari Component Pallette mengijinkan aplikasi untuk membaca atau menulis ke basisdata. Komponen pada halaman Data Access menggunakan Borland Database Engine (BDE) untuk mengakses informasi basisdata. Komponen pada halaman ADO menggunakan ActiveX Data Objext (ADO) untuk mengakses informasi basisdata melalui OLEDB. Komponen pada halaman Interbase mengakses suatu basisdata Interbase secara langsung.

Tergantung pada versi Delphi yang dipakai, BDE terdiri dari beberapa *driver* yang berbeda untuk jenis basisdata yang berbeda. Apapun jenis basisdatanya yang terdiri dari tabel yang menyimpan informasi, akan mendukung keamanan basisdata.

Basisdata sering berisi informasi sensitif. Perbedaan jenis basisdata adalah dalam menyediakan skema keamanan untuk melindungi informasi itu. Beberapa basisdata, seperti Paradox dan dBASE, yang hanya menyediakan keamanan di tingkat *field* atau tabel. Saat pemakai mencoba untuk mengakses tabel yang terproteksi, mereka harus memasukkan suatu kata sandi. Ketika sandi pemakai telah terbukti keasliannya, maka mereka hanya dapat melihat *field* itu dimana mereka diberi akses/ijin.

Kebanyakan SQL server memerlukan kata sandi dan pemakai (*password* dan *username*) untuk menggunakan semua basisdata server. Sekali pemakai telah masuk ke dalam basisdata dengan *username* dan kata sandi yang telah didefinisikan, maka pemakai hanya dapat mengakses tabel tertentu yang telah didefinisikan untuknya.

Ketika perancangan aplikasi basisdata, anda harus mempertimbangkan seperti apa macam pengesahan/otentifikasi yang diperlukan oleh basisdata server. Jika anda tidak ingin para pemakai harus menyediakan suatu kata sandi, maka anda harus yang menggunakan suatu basisdata yang tidak memerlukan kata sandi dan *userrname*, atau anda membuatnya secara program.

Teknologi Akses Basisdata

Delphi mendukung beberapa mesin basisdata, sehingga basisdata apapun dapat diakses melaluinya. Delphi menyediakan beberapa alternatif untuk mengakses database. Ini memudahkan pengembang dalam membuat program aplikasi. Mesin-mesin basisdata (*database engine*) itu diantaranya: BDE (Borland Database Engine), SQL Links, ODBC (*Open Database Connection*), ADO (*ActiveX Data Object*), OLE DB, IBX. Sedangkan basisdata yang dapat diakses diantaranya dBase, Paradox, MS Access, Text File, DB2, Oracle, MS SQL Server.

Model Pemrograman ADO

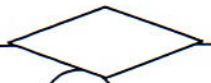
Model pemrograman ADO dibangun meliputi beberapa obyek ADO yang digunakan untuk mengakses bermacam-macam sumber data. Obyek-obyek ini menyediakan kemampuan untuk menghubungkan ke sumber data, *query* dan memperbarui kumpulan *record*, dan melaporkan kesalahan. Delphi, melalui beberapa komponen VCL menyediakan pembungkusan komponen untuk mengakses obyek-obyek, diantaranya:

- *Connection Object*, menghadirkan suatu koneksi ke sumber data dengan *string* koneksi. Di dalam Delphi, suatu obyek *connection* adalah suatu kombinasi komponen database dan *session*.
- *Command Object*, memungkinkan kita untuk mengoperasikan pada suatu sumber data, yang menghadirkan suatu perintah (juga dikenal sebagai suatu *query* atau *statement*) untuk dapat diproses yaitu untuk menambahkan data, menghapus data, *query* atau memperbarui data di dalam suatu basisdata.

Recordset Object adalah suatu hasil perintah *query*. Masing-masing baris yang dikembalikan *Recordset* terdiri dari banyak *field object*.

Flowchart

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program (Rieysha, 2009). Flowchart menolong analis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

SIMBOL	NAMA	FUNGSI
	**TERMINATOR**	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	**PREPARATION**	Proses inisialisasi/ pemberian harga awal
	**PROSES**	Proses penghitungan/ proses pengolahan data
	**INPUT/OUTPUT DATA**	Proses input/output data, parameter, informasi
	**PREDEFINED PROCESS (SUB PROGRAM)**	Permulaan sub program/ proses menjalankan sub program
	**DECISION**	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	**ON PAGE CONNECTOR**	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	**OFF PAGE CONNECTOR**	Penghubung bagian-bagian flowchart yang berada pada halaman yang berbeda

Tabel 1. Simbol Flowchart

3. Hasil dan Pembahasan

Kelemahan Database Konvensional

Semua data dan informasi yang terdapat dalam suatu sistem biasanya disimpan dalam suatu database. Jika seseorang memperoleh akses untuk membuka file database maka orang tersebut telah mengetahui semua data dan informasi yang ada dalam sistem. Informasi yang bersifat rahasia yang tersimpan dalam database juga ikut terbaca dan mungkin dapat tersebar di luar lingkungan dimana sistem tersebut dikembangkan. Sehingga kita dapat mengetahui kelemahan database yang umumnya digunakan dari sudut pandang kriptologi yaitu data yang tersimpan berbentuk asli atau apa adanya, sehingga hal tersebut merupakan sesuatu yang kurang dari segi keamanan informasi.
Permasalahan yang dihadapi adalah:

1. bagaimana merahasiakan informasi yang tersimpan dalam suatu *file* database.
2. bagaimana mengimplementasikan suatu algoritma kriptografi ke dalam sebuah aplikasi yang mendukung keamanan informasi dalam suatu sistem.

Untuk memecahkan masalah yang ada maka perlu dilakukan:

1. proses enkripsi data yang disimpan dalam database dengan menggunakan algoritma cipher aliran (*stream cipher*).
2. membangun sebuah aplikasi yang mengandung algoritma kriptografi untuk keamanan informasi yang tersimpan dalam suatu database.

Database Nasabah Bank

Untuk mempermudah implementasi, peneliti mengambil contoh database nasabah suatu sistem informasi perbankan. Setiap entitas nasabah mempunyai berbagai atribut yang melekat, antara lain nama, alamat, umur, dan nomor pin. Atribut nama, alamat dan umur adalah atribut yang bersifat umum yang dapat diketahui oleh semua orang. Tetapi atribut nomor pin setiap nasabah merupakan informasi yang bersifat rahasia. Dalam suatu sistem yang baik, informasi nomor pin tersebut hanya diketahui oleh administrator database yang ditunjuk untuk mengelola semua data yang tersimpan. Tetapi jika seorang administrator yang ditunjuk kurang bisa menjaga kerahasiaan maka informasi nomor pin setiap nasabah mungkin dapat tersebar luas sehingga dapat diketahui oleh semua orang, yang berujung pada kerugian di pihak nasabah yang bersangkutan. Jadi field yang sesuai dan perlu untuk proses enkripsi adalah field nomor pin nasabah.

Konversi Kode Desimal/Heksadesimal ASCII ke Biner

Setiap karakter yang diinputkan ke komputer melalui papan-ketik (*keyboard*) mempunyai kode ASCII yang berbentuk bilangan desimal atau heksadesimal. Sehingga data maupun informasi yang tersimpan pada sistem komputer pada prinsipnya mempunyai rangkaian kode ASCII. Rangkaian kode ASCII inilah yang dapat dijadikan sebagai bahan dasar untuk memulai proses enkripsi.

Telah dketahui bahwa algoritma kriptografi modern pada umumnya berbasis bit biner. Karena kode ASCII yang setiap karakter mempunyai bentuk desimal atau heksadesimal, maka perlu adanya konversi dari bentuk desimal/heksadesimal menjadi bilangan biner. Setiap bahasa pemrograman saat ini telah menyediakan fungsi konversi bilangan biner, desimal atau heksadesimal.

Enkripsi Cipher Aliran dengan *Keystream Generator*

Cipher aliran merupakan algoritma yang meng-enkripsi setiap bit (bit per bit) dalam sebuah rangkaian bit plainteks. Operasi matematikanya adalah dengan melakukan operasi XOR pada plainteks (P) dengan kunci (K) yang akan menghasilkan cipherteks:

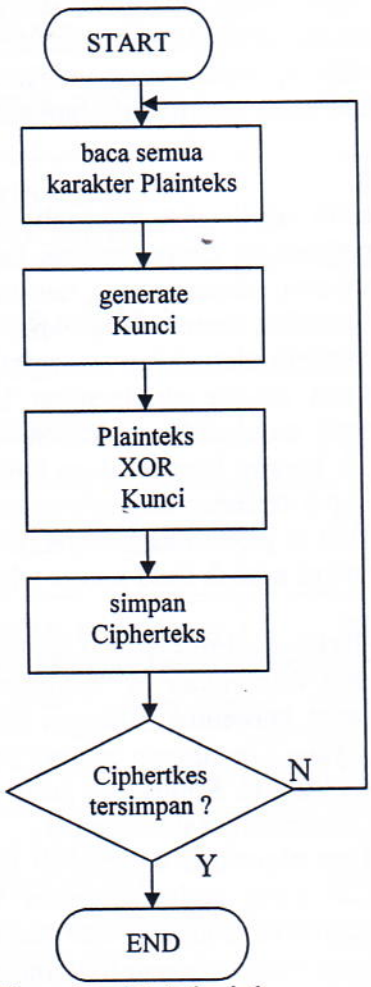
$$C = P \oplus K$$

Karena melakukan operasi XOR nilai yang sama dua kali berturut-turut menghasilkan nilai semula, maka dekripsi cipherteks menjadi plainteks menggunakan persamaan:

$$P = C \oplus K$$

Pembangkit aliran kunci (*Keystram Generator*) dapat membangkitkan bit-bit kunci berbasis bit per bit. Pembangkit aliran kunci diimplementasikan sebagai prosedur algoritma sehingga bit-bit kunci (K) dapat dibangkitkan secara simultan.

 ISSN 1978-9483

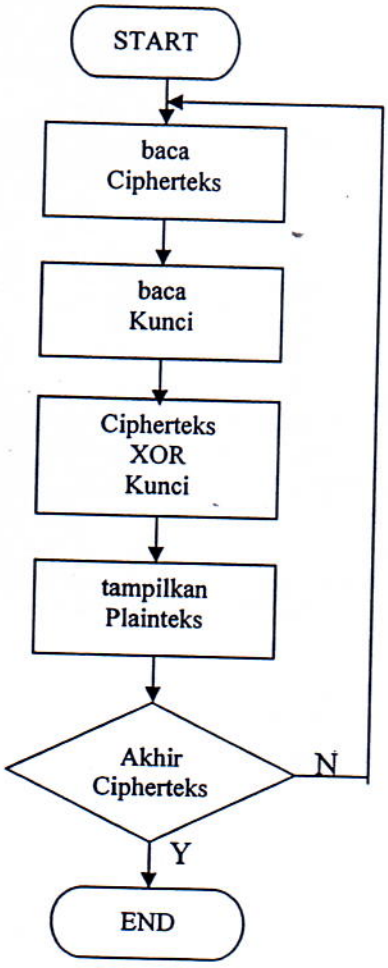


Gambar 5. Flowchart Enkripsi dengan operator XOR

Penjelasan:

1. Program akan membaca plainteks yang berbentuk deretan bilangan biner.
2. Program akan membaca kunci yang dihasilkan *Keystream Generator* yang juga berbentuk deretan bilangan biner.
3. Program akan melakukan operasi XOR antara plainteks dengan kunci yang akan menghasilkan cipherteks.
4. Program akan menyimpan cipherteks ke dalam database.
5. Program akan melakukan validasi apakah cipherteks sudah tersimpan dengan benar.

ISSN 1978-9483



Gambar 6. Flowchart Dekripsi dengan operator XOR

Penjelasan:

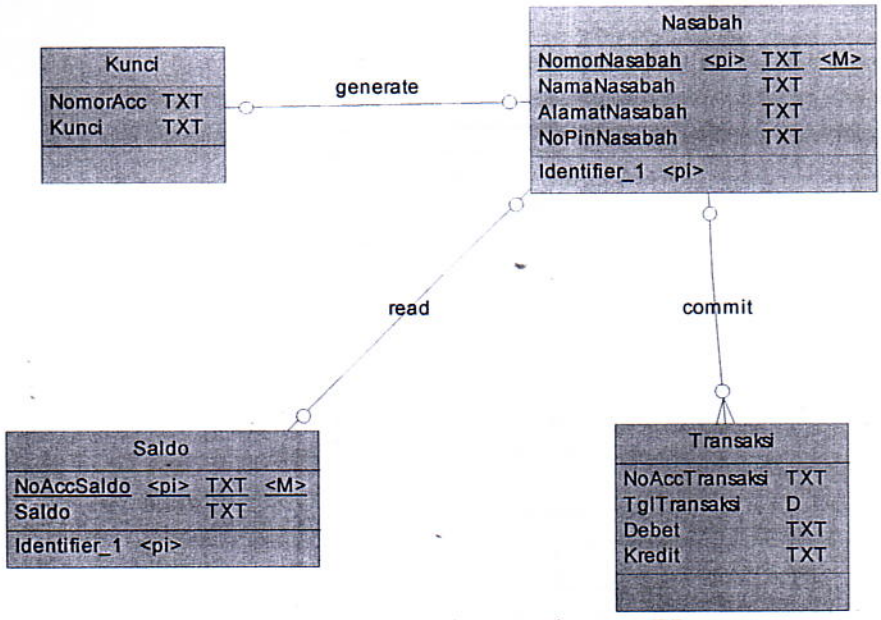
1. Program akan membaca seluruh cipherteks dari *file* database.
2. Program akan membaca kunci yang dihasilkan *Keystream Generator* yang juga berbentuk deretan bilangan biner.
3. Program akan melakukan operasi XOR antara cipherteks dengan kunci yang akan menghasilkan plainteks.
4. Program akan menampilkan hasil dekripsi berbentuk plainteks.
5. Program akan melakukan validasi apakah cipherteks seluruh cipherteks sudah dibaca dengan benar.

Alasan pemilihan enkripsi dengan cipher aliran

Penulis memilih algoritma cipher aliran karena algoritma cipher aliran merupakan salah satu algoritma kriptografi modern yang sederhana sehingga mudah dimengerti sebagai dasar pemahaman kriptografi modern.

Diagram ER (Entity Relationship)

Untuk memudahkan perancangan database, maka perlu dibuat suatu diagram hubungan antar-entitas (Entity Relationship Diagram):



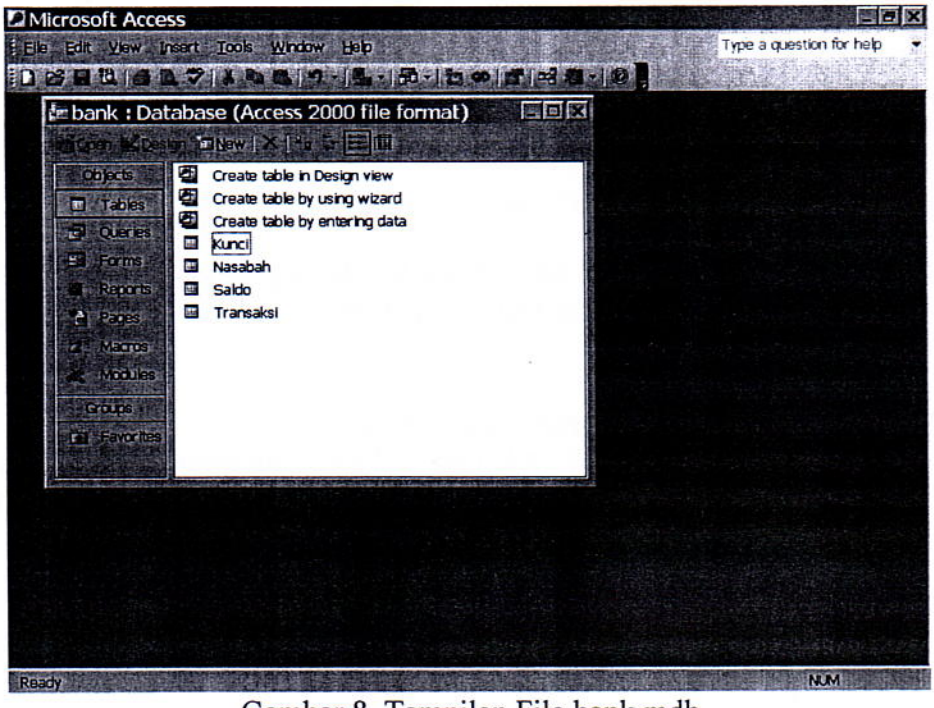
Gambar 7. Diagram ER

Perancangan Database dengan Ms Access

Setelah mengetahui hubungan antar-entitas, maka dibuat beberapa tabel yang mewakili entitas-entitas yang ada. Tabel-tabel tersebut meliputi:

1. Tabel Kunci: *field* NomorAcc, Kunci.
2. Tabel Nasabah: *field* NomorAcc, Nama, Alamat, NomorPin. *Field* NomorPin inilah yang akan diproses melalui enkripsi atau dekripsi.
3. Tabel Saldo: *field* NomorAccount, Saldo.
4. Tabel Transaksi: *field* NomorAccount, TanggalTransaksi, Debet, Kredit.

Semua tabel tersebut disimpan dalam file *bank.mdb*.

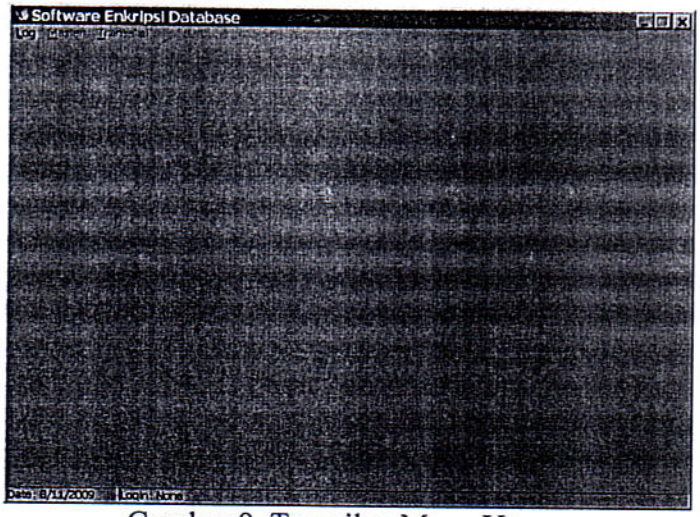


Gambar 8. Tampilan File bank.mdb

Perancangan Software Ekripsi Database

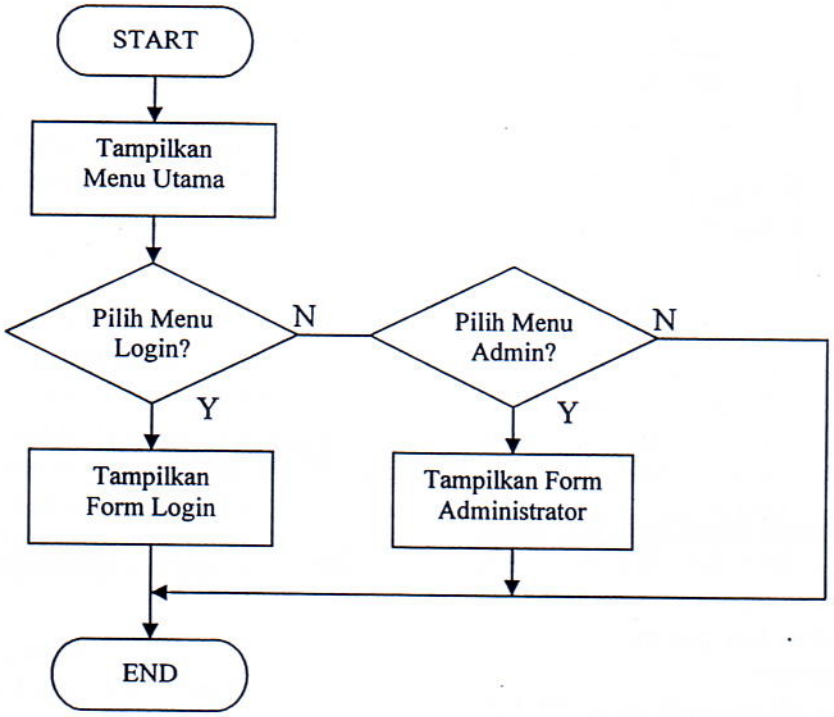
Perancangan tampilan program terdiri dari perancangan menu utama, menu login, form administrator, form nasabah dan form setor tunai.

Interface Menu Utama



Gambar 9. Tampilan Menu Utama

Flowchart Form Utama



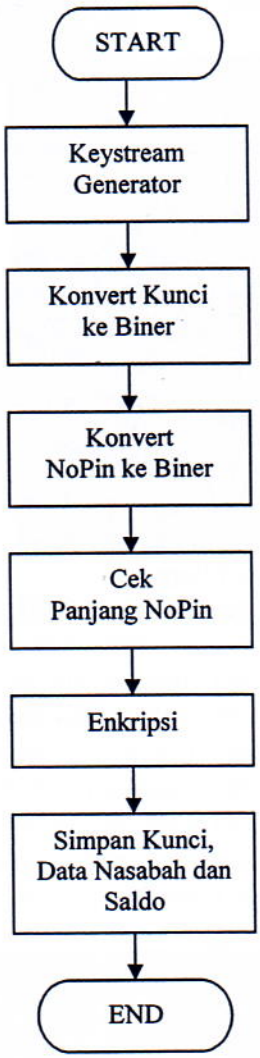
Gambar 10. Flowchart menu Utama

Penjelasan:

1. Program menampilkan Menu Utama yang berisi Menu Login dan Menu Administrator.
2. Apakah user memilih Menu Login?
   Jika ya maka program akan menampilkan Form Login. Jika tidak maka apakah user memilih Menu Admin? Jika ya maka program akan menampilkan Form Admin.

Perancangan Algoritma Enkripsi

Flowchart Algoritma Enkripsi



Gambar 11. Flowchar Algoritma Enkripsi

Baris Program Algoritma Enkripsi

Dari flowchart algoritma enkripsi di atas, kita dapat menuangkannya menjadi baris program seperti berikut ini:

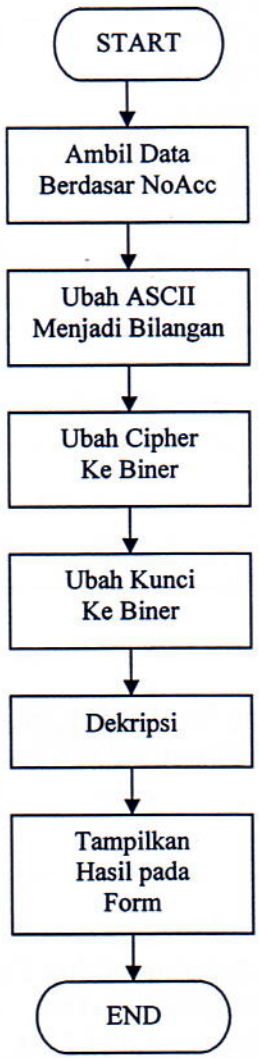
```
1:   //random key generator
2:   Randomize;
3:   Kunci:=RandomRange(128,255);

4:   //ubah kunci ke biner
5:   binKunci:=DecToBin(inttostr(Kunci));

6:   //ubah nopin ke biner
7:   for i:=1 to length(MaskEditNoPin.Text) do
8:     begin
9:     huruf:=(MaskEditNoPin.text[i]);
10:    kode:=ord(huruf);
11:    binPlain:=DecToBin(inttostr(kode));
```

```
12:  //cek apakah nopin 8 bit
13:    if length(binPlain)=1 then Insert('0000000',binPlain,1)
14:    else if length(binPlain)=2 then Insert('000000',binPlain,1)
15:    else if length(binPlain)=3 then Insert('00000',binPlain,1)
16:    else if length(binPlain)=4 then Insert('0000',binPlain,1)
17:    else if length(binPlain)=5 then Insert('000',binPlain,1)
18:    else if length(binPlain)=6 then Insert('00',binPlain,1)
19:    else if length(binPlain)=7 then Insert('0',binPlain,1);

20:    //enkripsi
21:    for j:=1 to length(binkunci) do
22:  Begin
23:  k:=strtoint(binkunci[j]) xor strtoint(binplain[j]);
24:  cipher:=cipher+inttostr(k);
25:  end;
26:    dec:=BinToDec(cipher);
27:    z:=Chr(dec);
28:    hasil:=hasil+z;
29:    cipher:='';      //reset cipher
30:    end;

31:  //simpan kunci
32:  ADOTableKunci.Open;
33:  ADOTableKunci.Insert;
34:  ADOTableKunci.FieldByName('kunci').AsString:=IntToStr(Kunci);
35:  ADOTableKunci.FieldByName('NomorAcc').AsString:=Edit2.Text;
36:  ADOTableKunci.Post;

37:  //simpan data nasabah
38:  ADOTableNasabah.Open;
39:  ADOTableNasabah.Insert;
40:  ADOTableNasabah.FieldByName('nomoracc').AsString:=Edit2.Text;
41:  ADOTableNasabah.FieldByName('nama').AsString:=Edit3.Text;
42:  ADOTableNasabah.FieldByName('alamat').AsString:=Edit4.Text;
43:  ADOTableNasabah.FieldByName('nomorpin').AsString:=hasil;
44:  ADOTableNasabah.Post;

45:  //simpan saldo=0
46:  ADOTableSaldo.Open;
47:  ADOTableSaldo.Insert;
48:  ADOTableSaldo.FieldByName('NomorAccount').AsString:=Edit2.Text;
49:  ADOTableSaldo.FieldByName('saldo').AsString:='0';
50:  ADOTableSaldo.Post;
51:  ADOTableSaldo.Close;
52:  ShowMessage('Data tersimpan');
53:  Edit2.SetFocus;
54:  Edit2.Clear;
55:  Edit3.Clear;
56:  Edit4.Clear;
```

```
57:  MaskEditNoPin.Clear;
58:  ButtonSimpan.Enabled:=False;
```

Perancangan Algoritma Dekripsi

Flowchart Algoritma Dekripsi



Gambar 12. Flowchart Algoritma Deskripsi

Baris Program Algoritma Dekripsi

Dari flowchart algoritma dekripsi di atas, kita dapat menuangkannya menjadi baris program seperti berikut ini:

```
1:  //ambil data berdasar nomoracc
2:  ADOQueryNasabah.Close;
3:  ADOQueryNasabah.SQL.Clear;
4:  ADOQueryNasabah.SQL.Add('select nomorpin,nama,alamat from  nasabah where
    nomoracc="'+nomor+'"');
5:  ADOQueryNasabah.Open;
6:  cipher:=ADOQueryNasabah.FieldByName('nomorpin').AsString;

7:  //ubah karakter ascii menjadi bilangan
8:  for i:=1 to length(cipher) do
```

```
9:      begin
10:     k:=cipher[i];
11:     bil:=Ord(k);
12:     //bilangan[i]:=bil;


13:     //ubah cipher ke biner
14:     binCipher:=dectobin(inttostr(bil));

15:     //ambil kunci dari tabel
16:     ADOQueryKunci.Close;
17:     ADOQueryKunci.SQL.Clear;
18:     ADOQueryKunci.SQL.Add('select kunci from kunci where nomoracc="'+nomor+'"');
19:     ADOQueryKunci.Open;
20:     kunci:=ADOQueryKunci.FieldByName('kunci').AsString;

21:     //ubah kunci ke biner
22:     binKunci:=DecToBin(kunci);

23:     //dekripsi
24:     for j:=1 to length(binKunci) do
25:      begin
26:      a:=StrToInt(binCipher[j]) xor StrToInt(binKunci[j]);
27:      plain:=plain+inttostr(a);
28:      end;
29:     b:=BinToDec(plain);
30:     hasil:=hasil+chr(b);
31:     plain:='';     //reset plain
32:     end;

33:   //tampilkan hasil pada form
34:   EditPin.Text:=hasil;
35:   EditNama.Text:=ADOQueryNasabah.FieldByName('nama').AsString;
EditAlamat.Text:=ADOQueryNasabah.FieldByName('alamat').AsString;
```

Perancangan Fungsi Biner To Decimal

Fungsi Biner To Decimal (disingkat BinToDec) digunakan untuk mengubah bentuk suatu bilangan biner menjadi bilangan desimal. Kode programnya adalah sebagai berikut:

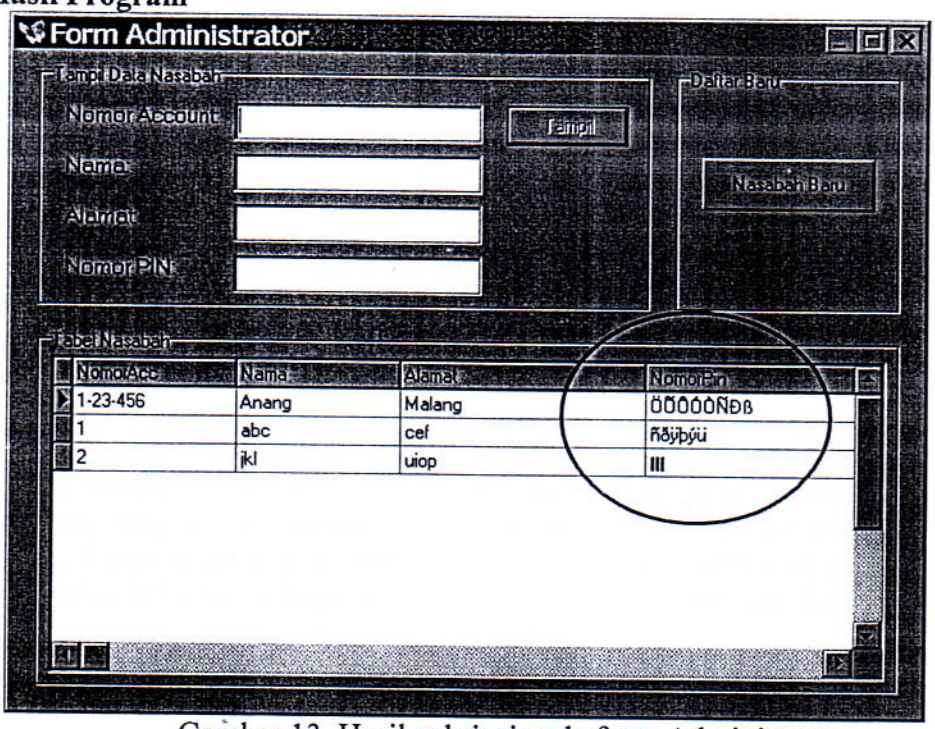
```
1:  function BinToDec(Input:string):Integer;
2:  var p,hasil,dua,i,j,bil:integer;
3:  begin
4:    hasil:=0;dua:=1;
5:    if StrToInt(RightStr(Input,1))=1 then
6:     begin
7:     bil:=1;
8:     p:=Length(input)-1;
9:     for i:=p downto 1 do
10:     begin
11:     if StrToInt(Input[i])=1 then
12:      begin
13:      for j:=0 to p-i do
14:       begin
15:       if StrToInt(Input[i])<>0 then dua:=dua*2;;
16:       end;
17:      hasil:=hasil+dua;
18:      dua:=1;
19:      end;
20:    end
21:   end
22:   else
23:   begin
24:   bil:=0;
25:   p:=Length(Input)-1;
26:   for i:=p downto 1 do
27:    begin
28:    if StrToInt(Input[i])=1 then
29:     begin
30:     for j:=0 to p-i do
31:      begin
32:      if StrToInt(Input[i])<>0 then dua:=dua*2;
33:      end;
34:     hasil:=hasil+dua;
35:     dua:=1;
36:     end;
37:    end
38:  end;
39: hasil:=hasil+bil;
40: BinToDec:=hasil;
      end;
```

Perancangan Fungsi Decimal To Biner

Fungsi Decimal To Biner (disingkat DecToBin) digunakan untuk mengubah bentuk suatu bilangan desimal menjadi bentuk biner. Kode programnya adalah sebagai berikut:

```
1:   function DecToBin(input:string):string;
2:   var kata,tot,bin:string;
3:       bil,i,sisa,hasil:integer;
4:   begin
5:    bil:=StrToInt(input);
6:    hasil:=2;
7:    while hasil >= 2 do
8:    begin
9:     hasil:=bil div 2;
10:    sisa:=bil mod 2;
11:    bil:=hasil;
12:    kata:=kata+IntToStr(sisa);
13:   end;
14:   tot:=kata+IntToStr(hasil);
15:   for i:=Length (tot) downto 1 do
16:   bin:=bin+tot[i];
17:   DecToBin:=bin;
    end;
```

Hasil Program



Gambar 13. Hasil enkripsi pada form Administrator

Hasil dari program adalah mendapatkan *field* NomorPin menjadi bentuk rahasia.

4. Kesimpulan

Dari hasil dan pembahasan yang diperoleh dari bab sebelumnya, maka dapat ditarik suatu kesimpulan:

1. Keamanan informasi yang tersimpan dalam database dapat ditingkatkan dengan adanya enkripsi yang diimplementasikan dalam suatu sistem informasi.
2. Informasi yang bersifat rahasia hanya dapat dibaca oleh pihak yang berkepentingan melalui proses dekripsi.

Daftar Pustaka

[1] Adi Nugroho. 2005. *Analisis Dan Perancangan Sistem Informasi Dengan Metodologi Berorientasi Obyek.* Bandung: Penerbit Informatika.

[2]Djoko Pramono. 1996. *Belajar Sendiri Pemrograman Delphi.* Jakarta: PT ELEX MEDIA KOMPUTINDO.

[3]Faried Irmansyah. 2003. *Pengantar Database*, (Internet), (http://www.ilmukomputer.com, diakses pada 11 Desember 2003)

[4] Rieysha · Published: June 14, 2009 · Category: Algoritma, Pemograman, Tool (http://ilmukomputer.org/?p=8036)

[5] Rinaldi Munir. 2006. *Kriptografi.* Bandung: Penerbit Informatika.

[6] Teddy Marcus, Agus Prijono, Joseph Widiadhi. 2002. *Pemrograman Delphi Dengan ADOExpress: Mengakses Basisdata MS. Access.* Bandung: CV. Informatika.