Implementasi Kriptografi Untuk Pengamanan Data Nasabah

Pada koperasi Sinode Parbubu Dengan Metode

DES (Data Encription Standard)

**Mario Eko Tondi Hutapea \*,Faisal Taufik, S.Kom., M.Kom\*\*, Afdal Alhafiz S.Kom., M.Kom\*\***

\* Sistem Informasi, STMIK Triguna Dharma

\*\* Program Studi Dosen Pembimbing, STMIK Triguna Dharma

|  |  |  |
| --- | --- | --- |
| **Article Info** |  | **ABSTRACT** |
| **Article history:**  Received Jun 12th, 201x  Revised Aug 20th, 201x  Accepted Aug 26th, 201x |  | *Dalam sebuah koperasi memiliki data atau informasi yang sangat penting dan perlu di lakukan untuk menjaga kerahasiaan data tersebut. Dikarenakan data-data yang ada didalam Koperasi Sinode Parbubu itu sendiri bersifat pribadi dan rahasia sehingga jika data tersebut jatuh kepihak yang tidak semestinya akan berdampak buruk terhadap reputasi Koperasi itu sendiri dan bahkan berdampak buruk juga bagi nasabah Koperasi Sinode Parbubu, misalnya bocornya data diri nasabah yang bersifat rahasia dan apabila data tersebut disalah gunakan orang yang tidak bertaggung jawab akibatnya sangat fatal, seperti penipuan yang mengatas namakan Koperasi sehingga Koperasi kehilangan reputasinya.* |
| **Keyword:**  First keyword  Second keyword  Third keyword  Fourth keyword  Fifth keyword |
| *Copyright © 2021 STMIK Triguna Dharma.  All rights reserved.* |
| **Corresponding Author:** \*First Author  Nama :Mario Eko Tondi Hutapea  Program Studi  STMIK Triguna Dharma  Email: [Marioeko85@gmail.com](mailto:Marioeko85@gmail.com) | | |

1. **PENDAHULUAN**

Perkembangan yang sedemikian cepatnya membawa dunia memasuki era baru yang lebih cepat dari yang di bayangkan sebelumnya. Seperti komputer yang tidak hanya di gunakan sebagai pengolahan data saja, namun telah menjadi senjata utama dalam berkompetisi. Hal ini dikarenakan dengan adanya komputer dapat mempermudah dan mempercepat suatu pekerjaan dalam mengakses informasi. Dengan berkembangnya teknologi informasi secara tidak langsung berpengaruh terhadap bidang keamanan, tidak terkecuali pada kurangnya keamanan data di koperasi sinode parbubu yang terdampak bocornya data nasabah, sehingga dapat disalah gunakan oleh pihak yang tidak bertanggung jawab [1].

Pengamanan dan kerahasiaan data merupakan salah satu aspek penting dari suatu sistem informasi. Sehingga informasi hanya bisa diakses oleh pemilik informasi atau user yang telah ditentukan oleh pemilik informasi. Dalam hal ini ancaman kebocoran data biasanya terjadi melalui web atau email, tetapi juga dapat terjadi melalui perangkat penyimpanan data seluler, media optic, kunci USB dan Laptop, sehingga tanpa disadari bisa mengakibatkan hilangnya data. Dalam sebuah koperasi memiliki data atau informasi yang sangat penting dan perlu di lakukan untuk menjaga kerahasiaan data tersebut. Dikarenakan data-data yang ada didalam Koperasi Sinode Parbubu itu sendiri bersifat pribadi dan rahasia sehingga jika data tersebut jatuh kepihak yang tidak semestinya akan berdampak buruk terhadap reputasi Koperasi itu sendiri dan bahkan berdampak buruk juga bagi nasabah Koperasi Sinode Parbubu, misalnya bocornya data diri nasabah yang bersifat rahasia dan apabila data tersebut disalah gunakan orang yang tidak bertaggung jawab akibatnya sangat fatal, seperti penipuan yang mengatas namakan Koperasi sehingga Koperasi kehilangan reputasinyaUntuk dapat menghindari masalah ini, perusahaan Koperasi Sinode Parbubu harus melakukan pengamanan data nasabah agar data tersebut aman dan terjaga keakuratanya[2].

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Teknik dalam kriptografi ini memiliki suatu kunci tertentu dengan menggunakan pengolahan informasi awal (plain text) yang tidak dapat dibaca, baru (cipher text) suatu informasi menghasilkan enkripsi tertentu sehingga menjadi informasi awal (plain text) melalui tersebut dapat dikembalikan cipher text secara langsung sehingga orang lain tidak dapat mengenali data tersebut. Adapun proses penamaannya disebut proses Enkripsi. Data atau pesan yang asli sering disebut sebagai plaintext dan data yang telah dienkripsi disebut yang lebih tepat encipher[3].

Ada beberapa metode dalam kriptografi, salah satunya yaitu Data Encryption Standard (DES). Data Encryption Standard (DES) adalah salah satu chiper block penyandian/kriptografi data yang populer dan telah dijadikan standard enkripsi kunci simetri sejak tahun 1976 dengan ukuran blok 64 bit dan ukuran kuncinya 56 bit. Data Encryption Standard (DES) adalah salah satu metode kriptografi cipher blok yang popular digunakan karena tingginya tingkat keamanan informasi dan dijadikan standard algoritma enkripsi kunci-simetri. DES adalah nama standard enkripsi simetri yang dahulu memiliki nama algoritma enkripsinya DEA (Data Encryption Algorithm), namun nama DES lebih populer dari pada DEA. Keamanan algoritma DES terletak pada banyaknya proses enkripsi dan dekripsi yang dilakukan sebanyak 16 kali putaran. Setiap putarannya akan menggunakan kunci internal yang berbeda. Hasil dari proses enchiper dipermutasi dengan matriks permutasi balikan (invers initial permutation) menjadi blok ciphertek[4].

Dari beberapa referensi kriptografi metode Data Encrytion Standard telah diterapkan untuk mengamankan data-data yang bersifat rahasia seperti keamanan informasi berbasis tanda tangan digital yang mana bertujuan untuk melakukan verifikasi apakah pesan atau informasi tersebut diterima dalam keadaan asli dari pengiriman atau telah dimodifikasi sehingga pesan atau informasi tersebut tidaklah asli dan DES juga mampu mengenkripsi dan dekripsi pesan atau informasi yang sangat rahasia dari orang-orang yang tidak bertanggung jawab dan tidak berkepentingan. Dari beberapa refrensi diatas dapat disimpulkan metode Data Encrytion Standard bisa dijadikan sebagai solusi untuk mengamankan data nasabah yang bersifat rahasia pada Koperasi Sinode Parbubu[5].

Harapannya sebuah sistem yang mengadopsi Data Encryption Standard dapat di implementasikan di Koperasi Sinode Parbubu. Sistem tersebut akan membantu pihak Koperasi Sinode Parbubu untuk mengamankan data nasabah. Sehingga data tersebut terjaga kerahasiaan dan keakuratan data tersebut. Berdasarkan deskripsi di atas maka penelitian ini diangkatlah sebuah judul IMPLEMENTASI KRIPTOGRAFI UNTUK PENGAMANAN DATA NASABAH PADA KOPERASI SINODE PARBUBU DENGAN METODE DES (DATA ENCRIPTION STANDARD).

1. **METODE PENELITIAN**

Metodologi Penelitian

Dalam teknik pengumpulan data dilakukan dengan dua tahapan, diantaranya yaitu:

**2.1 PENGUMPULAN DATA**

1. Observasi

Dalam penelitian ini dilakukan observasi pra-riset ter lebih dahulu untuk mencari masalah yang terjadi di koperasi sinode parbubu terkhusus dalam pengamanan data nasabah, dari masalah tersebut masalah akan dirumusakan dalam penelitian ini sehingga dapat menemukan rumusan apa saja yang perlu dipersiapkan untuk bagaimana cara menyelesaikan masalah tersebut.

2. Wawancara

Teknik wawancara ini dilakukan untuk mendapatkan informasi tambahan dari pihak-pihak yang memiliki wewenang dan berinteraksi langsung dengan dengan karyawan koperasi (Ferianto Sihombing).

Tabel 3. Tabel Data Nasabah

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NIK KTP | Nama Nasabah | Pekerjaan | Alamat | No. Telepon | Jumlah Pinjaman |
| 127116230888007 | Ramadani | Petani | Jl. Hela | 81377669066 | Rp. 5.000.000 |

**2.2. Studi Kepustakaan (Study of Literature)**

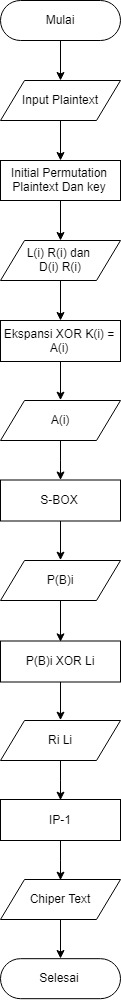
Di dalam studi literatur, penelitian ini banyak menggunakan jurnal-jurnal baik jurnal nasional, jurnal lokal maupun buku sebagai sumber referensi. Dari komposisi yang ada jumlah literatur yang digunakan sebanyak 32 sumber referensi. Diharapkan dengan literatur tersebut dapat membantu peneliti dalam menyelesaikan permasalahan yang terjadi di koperasi sinode parbubu terkait pengamanan data nasabah. Dikarenakan dalam penelitian ini menggunakan konsep pendekatan eksperimental maka berikut ini adalah tahapan penelitian yaitu sebagai berikut:

**2.2. Algoritma Sistem**

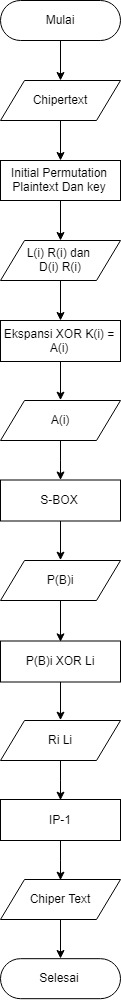
Algoritma sistem merupakan penjelasan langkah-langkah dalam penyelesaian masalah dalam perancangan sistem keamanan data nasabah dengan menggunakan algoritma DES. Hal ini dilakukan untuk meningkatkan keamanan data nasabah tersebut.

**2.3 Flowchart Dari Metode Penyelesaian**

Berikut ini adalah flowchart dari proses enkripsi dan dekripsi dari algoritma DES yaitu sebagai berikut:



Gambar 3. 2 Flowchart Proses Enkripsi



Gambar 3. 3 *Flowchart* proses dekripsi

**2.4 Dekripsi Data Penelitian**

Berikut ini adalah data nasabah yang di dapat dari koperasi sinode parbubu, yang akan diamankan. Dalam pengujiannya, sebagai contoh data yang digunakan sebagai sampel dalam penelitian ini yaitu sebagai berikut:

Tabel 3. Sampel Data nasabah koperasi sinode parbubu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NIK KTP | Nama Nasabah | Pekerjaan | Alamat | No. Telepon | Jumlah Pinjaman |
| 127116230888007 | Ramadani | Petani | Jl. Hela | 81377669066 | Rp. 5.000.000 |

### 2.4.1 Proses Enkripsi

Proses enkripsi adalah mengubah suatu data plaintext ke chippertext. Dalam proses enkripsi terdapat beberapa langkah-langkah berikut:

1. Mengubah *Plaintext* dan *Key* Menjadi Bilangan Biner

Mengubah *plaintext* kedalam biner berdasarkan tabel ASCII.

Tabel 3. Konversi plaintext ke biner

|  |  |  |  |
| --- | --- | --- | --- |
| **PLAINTEXT** | | | |
|  | **Dec** | **hexa** | **Biner** |
| R | 82 | 52 | 01010010 |
| a | 97 | 61 | 01100001 |
| m | 109 | 6D | 01101101 |
| a | 97 | 61 | 01100001 |
| d | 100 | 64 | 01100100 |
| a | 97 | 61 | 01100001 |
| n | 110 | 6E | 01101110 |
| i | 105 | 69 | 01101001 |

Mengubah *key* ke dalam biner berdasarkan tabel ASCII

Tabel 3. Konversi key ke biner

|  |  |  |  |
| --- | --- | --- | --- |
| **KEY** | | | |
|  | **Dec** | **Hexa** | **Biner** |
| M | 77 | 4D | 01001101 |
| A | 65 | 41 | 01000001 |
| R | 82 | 52 | 01010010 |
| I | 73 | 49 | 01001001 |
| O | 79 | 4F | 01001111 |
| E | 69 | 45 | 01000101 |
| K | 75 | 4B | 01001011 |
| O | 79 | 4F | 01001111 |

1. *Initial Permutation Plaintext*

Lakukan *initial permutation* (IP) pada bit plaintext menggunakan tabel IP seperti berikut:

Tabel 3. Initial permutation

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PLAINTEXT (X)** | | | | | | | |  |  | **IP1** | | | | | | | |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |  |  | 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |  |  | 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |  | | 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |  |  | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |  |  | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |  |  | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

Keterangan pada tabel *initial permutation* dan tabel IP(X):

Angka 0 dan 1 merupakan bilangan biner

Angka **1,2,3** dan seterusnya yang menggunakan penebalan adalah urutan posisi bit

Urutan bit ke-58 pada tabel *plaintext* (X), diletakan pada posisi 1 pada tabel IP,

Urutan bit ke-50 pada tabel *plaintext* (X), diletakan pada posisi 2 pada tabel IP,

Urutan bit ke-42 pada tabel *plaintext* (X), di letakan pada posisi 3 pada tabel IP,

Demikian seterusnya dan menghasilkan Tabel IP(X).

Tabel 3. IP(X)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IP(X)** | | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **L0** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **R0** |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

Selanjutnya bit pada IP(X) di pecah menjadi dua bagian yaitu L0 dan R0 sehingga hasilnya dapat di lihat pada tabel 3.5.

1. Melakukan Permutasi *Key* Kompresi PC-1

Kunci yang sudah diubah menjadi bilangan biner, lalu di permutasikan dengan menggunakan tabel permutasi kompresi PC-1, pada langkah ini terjadikompresi 64 bit menjadi 56 bit.

Tabel 3. Permutasi Kompresi PC-1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **KEY** | | | | | | | |  |  | **PC1** | | | | | | |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |  |  | 57 | 49 | 41 | 33 | 25 | 17 | 9 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |  |  | 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |  | | 10 | 2 | 59 | 51 | 43 | 35 | 27 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 63 | 55 | 47 | 39 | 31 | 23 | 15 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |  |  | 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |  |  | 14 | 6 | 61 | 53 | 45 | 37 | 29 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |  |  | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

Keterangan pada tabel Permutasi Kompresi PC-1

Angka 0 dan 1 merupakan bilangan biner

Angka **1,2,3** dan seterusnya yang menggunakan penebalan adalah urutan posisi bit

Urutan bit ke-57 pada tabel *key*, diletakan pada posisi 1 pada Tabel PC-1,

Urutan bit ke-49 pada tabel *key*, diletakan pada posisi 2 pada Tabel PC-1 dst, dan hasil permutasi *key* dapat di lihat pada tabel 3.8.

Tabel 3. PC-1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TABEL PC-1** | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | **C0** |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | **D0** |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |

Selanjutnya bit pada Tabel hasil permutasi PC-1 di pecah menjadi dua bagian yaitu C0 dan D0 sehingga hasilnya sebagai berikut.

C0: 0000000 0111111 1100000 0000000

D0: 1101010 0101100 0111011 0010100

1. Melakukan Pergeseran Kiri

Lakukan pergeseran kiri (*left Shift Operation*) pada C0 dan D0 sebanyak satu atau dua kali berdasarkan putaran yang ada pada tabel putaran sebagai berikut:

Tabel 3. Left shif

|  |  |
| --- | --- |
| **Putaran ke – i** | **Jumlah Pergeseran Bit (Left Shift)** |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 2 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 2 |
| 9 | 1 |
| 10 | 2 |
| 11 | 2 |
| 12 | 2 |
| 13 | 2 |
| 14 | 2 |
| 15 | 2 |
| 16 | 1 |

Keterangan:

Untuk putaran ke-1, dilakukan pergeseran 1bit ke kiri,

Untuk putaran ke-2, dilakukan pergeseran 1 bit ke kiri,

Untuk putaran ke-3, dilakukan pergeseran 2 bit ke kiri, dan seterusnya hingga putaran yang ke-16.

Berikut hasil dari *left shift*:

Putaran ke-1, di geser 1 bit ke kiri.

C1: 0000000 1111111 1000000 0000000

D1: 1010100 1011000 1110110 0101001

Putaran ke-2, di geser 1 bit ke kiri.

C2: 0000001 1111111 0000000 0000000

D2: 0101001 0110001 1101100 1010011

Putaran ke-3, di geser 2 bit ke kiri.

C3: 0101001 0110001 1101100 1010011

D3: 0000111 1111100 0000000 0000000 dst.

C16: 0000000 0111111 1100000 0000000

D16: 1101010 0101100 0111011 0010100

Setiap hasil putaran digabungkan kembali menjadi CiDi dan diinput kedalam tabel *Permutation Compression* 2 (PC-2) dan terjadi kompresi data CiDi 56 bit menjadi CiDi 48 bit dan mengkasilkan Ki.

Tabel 3. Permutation Compression 2 (PC-2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PC-2** | | | | | |
| 14 | 17 | 11 | 24 | 1 | 5 |
| 3 | 28 | 15 | 6 | 21 | 10 |
| 23 | 19 | 12 | 4 | 26 | 8 |
| 16 | 7 | 27 | 20 | 13 | 2 |
| 41 | 52 | 31 | 37 | 47 | 55 |
| 30 | 40 | 51 | 45 | 33 | 48 |
| 44 | 49 | 39 | 56 | 34 | 53 |
| 46 | 42 | 50 | 36 | 29 | 32 |

Keterangan:

Urutan bit pada CiDi yang ke-14, diletakan di posisi 1 pada tabel PC-2,

Urutan bit pada CiDi yang ke-17, diletakan di posisi 2 pada tabel PC-2,

Urutan bit pada CiDi yang ke-11, diletakan di posisi 3 pada tabel PC-2, dan seterusnya.

Berikut hasil *output*nya:

C1D1: 0000000 1111111 1000000 0000000 1010100 1011000 1110110 0101001

K1: 101000 001001 001001 000010 001010 001111 101101 000110

C2D2: 0000001 1111111 0000000 0000000 0101001 0110001 1101100 1010011

K2: 101000 000001 001001 010010 010111 100000 100100 111001

C3D3: 0000111 1111100 0000000 0000000 0100101 1000111 0110010 1001101 dst.

C16D16: 0000000 0111111 1100000 0000000 1101010 0101100 0111011 0010100

K16: 101000 001001 001000 100010 010100 110101 111010 100011

1. Melakukan Ekspansi Data

Pada langkah ini, kita akan meng-ekspansi data Ri-1 32 bit menjadi Ri 48 bit sebanyak 16 kali putaran dengan nilai perputaran 1>= i <=16 menggunakan Tabel Ekspansi (E).

Tabel 3. Ekspansi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabel Ekspansi** | | | | | |
| 32 | 1 | 2 | 3 | 4 | 5 |
| 4 | 5 | 6 | 7 | 8 | 9 |
| 8 | 9 | 10 | 11 | 12 | 13 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 17 | 18 | 19 | 20 | 21 |
| 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 28 | 29 | 30 | 31 | 32 | 1 |

Hasil E(Ri-1) kemudian di XOR dengan Ki dan menghasilkan Vektor Matriks Ai. Berikut hasil *output*nya:

**Iterasi 1**

E((R1)-1) : 100000 000001 011111 111101 011000 001000 001000 000010

K1 : 101000 001001 001001 000010 001010 001111 101101 000110

**--------------------------------------------------------------------------------------------**XOR

A1 : 001000 001000 010110 111111 010010 000111 100101 000100

Pada iterasi satu (1) diatas didapat A1 dari hasil XOR E(R1-1) dan K1, setelah itu maka proses selanjutnya langsung ke langkah ke-6 terlebih dahulu, dimana Ai, akan dimasukan ke dalam S-BOX dan menghasilkan PB1 yang kemudian di XOR kan dengan L0 dan menghasilkan nilai Ri. Nilai Ri ini digunakan untuk melanjutkan iterasi ke-2.

1. Memasukan Data ke S-BOX

A1 : 001000 001000 010110 111111 010010 000111 100101 000100

Tabel 3. Subtitusi S1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0000** | **0001** | **0010** | **0011** | **0100** | **0101** | **0110** | **0111** | **1000** | **1001** | **1010** | **1011** | **1100** | **1101** | **1110** | **1111** |
| **00** | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| **01** | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| **10** | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| **11** | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

Kemudian kita ambil sampel blok bit pertama yaitu **001000**, pisahkan menjadi 2 blok yaitu:

1. Bit pertama dan terakhir yaitu 1 dan 0, digabungkan menjadi 00
2. Bit kedua hingga kelima yaitu 0100

Selanjutnya dibandingkan dengan memeeriksa perpotongan antara kedua di dapatkan nilai 2 (warna kuning) lalu dibinerkan menjadi **0010**

Tabel 3. Subtitusi S2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0000** | **0001** | **0010** | **0011** | **0100** | **0101** | **0110** | **0111** | **1000** | **1001** | **1010** | **1011** | **1100** | **1101** | **1110** | **1111** |
| **00** | 15 | 1 | 8 | 14 | 6 | 11 | 3 | 4 | 9 | 7 | 2 | 13 | 12 | 0 | 5 | 10 |
| **01** | 3 | 13 | 4 | 7 | 15 | 2 | 8 | 14 | 12 | 0 | 1 | 10 | 6 | 9 | 11 | 5 |
| **10** | 0 | 14 | 7 | 11 | 10 | 4 | 13 | 1 | 5 | 8 | 12 | 6 | 9 | 3 | 2 | 15 |
| **11** | 13 | 8 | 10 | 1 | 3 | 15 | 4 | 2 | 11 | 6 | 7 | 12 | 0 | 5 | 14 | 9 |

Kemudian kita ambil sampel blok bit kedua yaitu **001000**, pisahkan menjadi 2 blok yaitu:

1. Bit pertama dan terakhir yaitu 0 dan 0, digabungkan menjadi 0
2. Bit kedua hingga kelima yaitu 0100

Selanjutnya dibandingkan dengan memeeriksa perpotongan antara kedua di dapatkan nilai 6 (warna kuning) lalu dibinerkan menjadi **0110.**

Dan seterusnya untuk blok ketiga hingga blok kedelapan dibandingkan dengan S3 dan S8. Berdasrkan cara diatas diperoleh hasil sebagai berikut:

B1 = 00100110 01111110 01010010 11011000

1. Memutasikan Bit Vektor Bi

Setelah didapatkan nilai vector Bi, langkah selanjutnya adalah memutasikan bit vektor Bi menggunakan tabel P-*BOX*, lalu dikelompokkan menjadi 4 blok dimana setiap blok memiliki 32 bit data

Tabel 3. Matrik Permutasi P (P-box)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **P-BOX** | | | | | | | |
| 16 | 7 | 20 | 21 | 29 | 12 | 28 | 17 |
| 1 | 15 | 23 | 26 | 5 | 8 | 31 | 10 |
| 2 | 8 | 24 | 14 | 32 | 27 | 3 | 9 |
| 19 | 13 | 30 | 6 | 22 | 11 | 4 | 25 |

Sehingga hasil yang didapatkan sebagai berikut:

P(B1) : 01101110 01110101 00010010 01010101

Hasil P(Bi) kemudian di XOR kan dengan Li-1untuk mendapatkan nilai Ri. Sedangkan nilai Li sendiri diperoleh dari nilai Ri-1 untuk nilai 1>=i<=16.

L0 : 01101110011101010001001001010101

R0 : 11111111000000010101010010101110

P(B1) : 01101110 01110101 00010010 01010101

L0 : 01101110011101010001001001010101

**--------------------------------------------------------------------------------------------**XOR

R1 : 10010001011101000100011011111011

Untuk mencari R2 sampai R16, lakukan langkah yang sama dari langkah 5 sampai 7 dan dituliskan dalam bentuk itrasi, sehingga pada iterasi ke-16 didapatkan hasil sebagai berikut.

**Iterasi 16**

E (R15) : 000111111011110110100110101011111001010011111100

K16 : 101000001001001000100010010100110101111010100011

**--------------------------------------------------------------------------------------------**XOR

A16 : 101111110010111110000100111111001100101001011111

B16 : 00100110011010111001001011001110

P(B16) : 11100000101000011101110011111111

L(16)-1 : 11000110110010100100111000110001

**--------------------------------------------------------------------------------------------**XOR

R16 : 11000110110010100100111000110001

L(16)-1 : 11000110110010100100111000110001

1. Menggabungkan R16 dan L16

Langkah terakhir adalah menggabungkan R16 dengan L16 kemudian dipermutasikan dengan tabel *initial permutation* (IP-1).

Tabel 3. Permutasi R16 dan L16 dengan Tabel IP­-1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **R16 dan L16** | | | | | | | | |  |  | **TABEL IP-1** | | | | | | | |
| **R16** | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |  |  | 40 | 8 | 48 | 16 | 56 | 24 | 64 | 32 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |  |  | 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |  |  | 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |  | | 37 | 5 | 45 | 13 | 53 | 21 | 61 | 29 |
| **L16** | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 36 | 4 | 44 | 12 | 52 | 20 | 60 | 28 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |  |  | 35 | 3 | 43 | 11 | 51 | 19 | 59 | 27 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |  |  | 34 | 2 | 42 | 10 | 50 | 18 | 58 | 26 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |  |  | 33 | 1 | 41 | 9 | 49 | 17 | 57 | 25 |

Tabel 3. Chipertext

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CHIPERTEKS** | | | | | | | |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

Menghasilkan *output*:

*Chiper* dalam biner : **10100001 01110110 11001110 10011110 10101011 10100001 01011100 01110010**

Atau dalam *chiper* hexa : **A1 76 CE 9E AB A1 5C 72**

### 3.3.4 Proses Dekripsi

Untuk dapat mengetahui isi pesan sebenarnya, perlu dilakukan konversi *ciphertext* menjadi bentk biner untuk mendapatkan bit *chipertext*. Dekripsi dapat dilakukan sebagai berikut:

1. Melakukan Permutasi Terhadap *Chiper*

*Chiper* dalam biner : **10100001 01110110 11001110 10011110 10101011 10100001 01011100 01110010**

Atau dalam *chiper* hexa : **A1 76 CE 9E AB A1 5C 72**

Tabel 3. Initial permutation chipper (IP)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Cipher*text** | | | | | | | |  |  | **Tabel IP** | | | | | | | |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |  |  | 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |  |  | 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |  |  | 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |  |  | 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |  |  | 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |  |  | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |  |  | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |  |  | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

Tabel 3. Hasil *initial permutation chipper* (IP)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IP(*Cipher*)** | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | **L0** |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | **R0** |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

Selanjutnya bit pada IP (*Chiper*) dipecah menjadi 2 bagian yaitu L0 dan R0,

Sehingga menghasilkan sebagai berikut:

L0 : 11000110 11001010 01001110 00110001

R0 : 00111101 10110011 01011100 10011110

**Iterasi 16**

P(B16) : 00100110011010111001001011001110

L15 : 11000110110010100100111000110001

**--------------------------------------------------------------------------------------------**XOR

R16 : 11100000101000011101110011111111

Lakukan iterasi 15-1 sehingga didapatkan, pada iterasi pertama sebagai berikut.

**Iterasi 1**

P(B1) : 01001010 00000101 00011111 10011100

L0 : 01001010 11111010 00011111 10011001

**--------------------------------------------------------------------------------------------**XOR

**R1 : 11111111000000010101010010101110**

**L1 : 00000000111111101100010001000001**

1. Melakukan Permutasi R1 dan L1 dengan Tabel IP-1

Kemudian R1 dan L1 di permutasikan kembali dengan tabel *inverse initial permutation* sehingga menghasilkan *output*:

*Plaintext* dalam biner : **01010010 01100001 01101101 01100001 01100100 01100001 01101110 01101001**

Atau dalam bentuk hexa: **52 61 6D 61 64 61 6E 69**

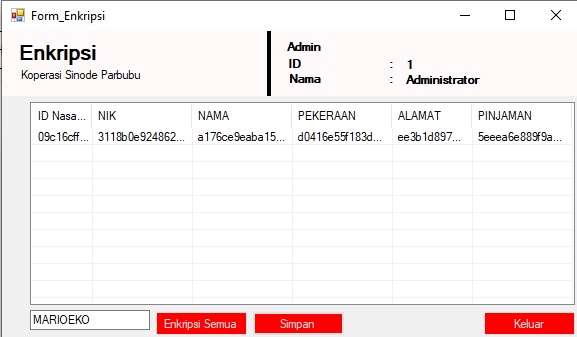
Atau dalam bentuk char: **Ramadani**

1. **ANALISA DAN HASIL**

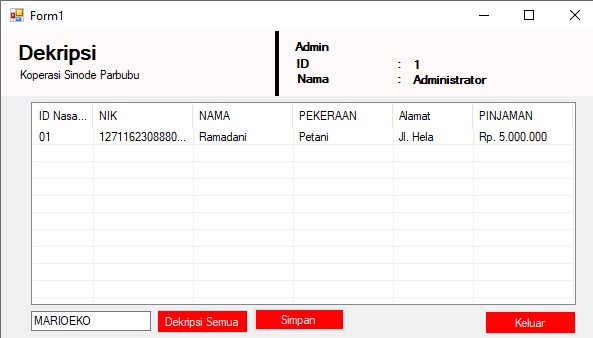
## 3.1 Pengujian Sistem

Uji coba sistem bertujuan untuk membuktikan bahwa *input*, *proses*, *output* yang dihasilkan oleh sistem aplikasi *Visual Studio* 2012telah benar dan sesuai dengan yang diinginkan. Pengujian sistem dengan cara memasukkan data ke dalam sistem dan memperhatikan *output* yang dihasilkan. Jika *input,* proses dan *output* telah sesuai, maka sistem telah benar. Berikut merupakan tahapan untuk pengujian sistem yaitu:

1. Melakukan *input* data nasabah yang kemudian sistem akan menampilkan data nasabah yang tersimpan di *database*.
2. Menggunakan bahasa pemrograman *Microsoft Visual Studio* 2012dalam pengolahan data yang disimpan dalam *database Microsoft Office Access* 2010 *.* Penggunaan sistem pengamanan data nasabah pada PT.Cogindo, agar dapat berjalan dengan baik *file* aplikasi *Visual Studio* 2012harus ditempatkan pada satu *folder* dan dilengkapi dengan *input* data dari analisa sistem. Lokasi *folder* yang telah ditentukan adalah tempat untuk menyimpan *file*-*file* yang telah dikumpulkan, untuk menghindari kesalahan sebaiknya data tidak diletakkan kedalam *folder* yang berbeda. Selanjutnya untuk menerapkan metode dalam mengamankan data nasabah, maka data tersebut akan *diinput* ke aplikasi lalu simpan data tersebut ke dalam *database Access*. Jalankan aplikasi *Visual Studio* 2012yang telah terinstal dikomputer. Berikut ini merupakan hasil pengujian yang dilakukan pada sistem.



Gambar 5. Pengujian untuk data nasabah enkripsi



Gambar 5. Pengujian untuk data nasabah dekripsi

1. **KESIMPULAN**

Berdasarkan pembahasan dan evaluasi dari bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut :

1. Dalam menganalisa masalah yang terjadi terkait dengan pengamanan data nasabah di Koperasi Sinode Parbubu didapat data dari Koperasi Sinode parbubu berupa data-data nasabah yang sangat penting dan oleh sebab itu maka dilakukan proses enkripsi untuk data nasabah terkait dengan menggunakan algoritma DES *(Data Encryption Standard)* untuk menajaga kerahasiaan dan keamanan data nasabah pada Koperasi Sinode Parbubu.
2. Perancang sistem kriptografi yang mengadopsi algoritma DES (*Data Encryption Standard)* dengan metode sistem *Block Cipher* di dalam menyelesaikan masalah terkait pengamanan data nasabah di Koperasi Sinode Parbubu menggunakan pemrograman yang berbasis desktop.
3. Pengujian sistem ini dilakukan sebelum nantinya dapat digunakan untuk membantu Koperasi Sinode Parbubu terkait di dalam pengamanan data nasabah.

**UCAPAN TERIMA KASIH**

Terimakasih buat sahabat yang selalu menemani dan memberikan suport kepada saya selama dalam menyelesaikan artikel ilmiah ini terkhusus buat senior dan teman teman kos F4 yang tiada henti-hentinya memberikan masukan-masukan dan saran dalam penyusunan karya ilmiah ini.

**REFERENSI**

1. G. Putrodjojo, J. H. Purba, and J. Candra, “Aplikasi Algoritma Des (Data Encryption Standard) Untuk Pengaman Data,” *CCIT J.*, vol. 10, no. 1, pp. 62–74, 2017, doi: 10.33050/ccit.v10i1.518.
2. B. Rahardjo, *Keamanan Informasi Berbasis Internet*, vol. 0. 1999.
3. P. Pahrizal and D. Pratama, “Implementasi Algoritma Rsa Untuk Pengamanan Data Berbentuk Teks,” *Pseudocode*, vol. 3, no. 1, pp. 44–49, 2016, doi: 10.33369/pseudocode.3.1.44-49.
4. I. M. Arrijal, R. Efendi, and B. Susilo, “Penerapan Algoritma Kriptografi Kunci Simetris Dengan Modifikasi Vigenere Cipher Dalam Aplikasi Kriptografi Teks,” *Pseudocode*, vol. 3, no. 1, pp. 69–82, 2016, doi: 10.33369/pseudocode.3.1.69-82.
5. R. Munir, “Slide Kuliah Pengantar Kriptografi,” 2019.

**BIBLIOGRAFI PENULIS**

|  |  |
| --- | --- |
|  | Nama : Mario Eko Tondi Hutapea  Nirm : 2017020743  Program Studi : Sistem Informasi  Deskripsi : Anak Desa yang seharinya kerja sambil kuliah dan salah satu Mahasiswa Stmik Triguna Dharma Stambuk 2017 yang masuk dalah kuliah malam dan sekarang sedang berfokus kepada bidnag keahlian desain grafis yang dimana suatu saat nanti setelah mendapat gelar alumni bisa membuka salah satu ruang pekerjaan dalam bidang desainer. |
|  | Nama : Faisal Taufik  NIDN : 0104038603  Program Studi : Sistem Informasi  Deskripsi : Dosen Tetap STMIK Triguna Dharma yang aktif mengajar dan fokus pada bidang keilmuan pemrograman komputer. |
|  | Nama : Afdal Al Hafiz  NIDN. : 011405930  Deskripsi : Dosen Tetap STMIK Triguna Dharma yang aktif mengajar dan fokus pada bidang keilmuan sistem kendali |