DESAIN SISTEM KEAMANAN KRIPTOGRAFI ELEKTRONIK *CODE BOOK* PADA DATA *LOGIN*

SKRIPSI

Diajukan untuk memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

OLEH :

NAMA : ABDI NUGE RAHANTA
NPM : 1514371036
PROGRAM STUDI : SISTEM KOMPUTER

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

ABSTRAK

ABDI NUGE RAHANTA
Desain Sistem Keamanan Kriptografi Elektronik Code Book Pada Data Login
Tahun 2019

Saat ini, keamanan terhadap data yang tersimpan sudah menjadi pesyaratan mutlak. Pengamanan terhadap jaringan komputer yang terhubung dengan basis data sudah tidak lagi aman karena kebocoran data dapat disebabkan oleh orang-orang yang tidak bertanggung jawab. Hasil program ini dapat digunakan untuk mengamankan data. Salah satu kriptografi yang cocok dalam pengamanan data tersebut adalah kriptografi algoritma Electronic Code Book karena kriptografi Electronic Code Book ini cocok untuk mengenkripsi password pada sebuah data login. Hasil program menunjukkan algoritma Electronic Code Book (ECB) cocok untuk pengamanan ini, karena dapat mengenkripsi password pada data login.

Kata kunci : kriptografi, electronic code book, ECB.

DAFTAR ISI

Halaman

KATA PENGANTAR

Puji Syukur Kehadirat Tuhan Yang Maha Esa karena dengan berkat dan kasih anugrahNya yang telah memberi kekuatan dan kesabaran kepada penulis, dimana hambatan selalu penulis hadapi, akan tetapi berkat izin Tuhan Yang Maha Esa dan berkat bimbingan, bantuan dan dorongan dari berbagai pihak, akhirnya penulis dapat melalui hambatan yang dihadapi hingga akhirnya dapat menyelesaikan penelitian dan penulisan Skripsi ini. Pada kesempatan ini penulis mengucapkan terimakasih kepada :

1. Kedua Orangtua, Istri dan Anak tercinta dimanapun berada. Terimakasih atas kasih sayang, Doa, dukungan dan dorongannya.
2. Rektor Universitas Pembangunan Panca Budi, Bapak Dr. H. Muhammad Isa Indrawan, SE, M.M
3. Rektor I, Bapak Ir. Bhakti Alamsyah, M.T, Ph.D
4. Dekan Fakultas Sains dan Teknologi, Ibu Sri Sindhi Indira, ST, M.Sc
5. Ketua Program Studi Sistem Komputer, Bapak Eko Hariyanto, S.Kom, M.Kom.
6. Dosen Pembimbing I, Bapak Dr. Muhammad Iqbal, S.Kom, M.Kom.
7. Dosen Pembimbing II, Bapak Zulham Sitorus, S.Kom, M.Kom.
8. Bapak/Ibu para Dosen dan Staf Admin Program Studi Sistem Komputer.
9. Bapak/Ibu para Staf Pegawai dan elemen-elemen di Universitas Pembangunan Panca Budi Medan.
10. Sumiratno, Mentari, Iin Puspita Dewi, Zaki Baginda Muhammad Amin, Adrya Rama, Daniel Panjaitan, Ibnu Gunawan, Ernanda Setiawan, Rizky Haryanto, Inon Fransdinata Kaban, Aldri Yaumar dan seluruh teman-teman seperjuangan Kelas Karyawan II J/S Program Studi Sistem Komputer di Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan, semuanya adalah teman-teman yang hebat dan sangat menginspirasi.

Penulis juga menyadari bahwa penyusunan Skripsi ini belum sempurna baik dalam penulisan maupun isi disebabkan keterbatasan kemampuan penulis. Oleh karena itu, Penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca untuk penyempurnaan Skripsi ini.

Medan, November 2019

Penulis

ABDI NUGE RAHANTA
1514371036

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Saat ini, keamanan terhadap data yang tersimpan dalam basis data sudah menjadi pesyaratan mutlak. Pengamanan terhadap jaringan komputer yang terhubung dengan basis data sudah tidak lagi aman, karena kebocoran data dapat disebabkan oleh orang-orang yang tidak bertanggung jawab atau pihak-pihak yang langsung berhubungan dengan basis data harus menemukan cara untuk mengamankan data tanpa campur tangan administrator basis data.

Kriptografi dapat digunakan untuk mengamankan data. Oleh karena itu, pengguna basis data membutuhkan bantuan untuk memenuhi kebutuhan keamanan akan data yang disimpannya.

Salah satu upaya pengamanan sistem informasi tersebut yang dapat dilakukan adalah kriptografi. Kriptografi adalah ilmu atau seni untuk menjaga keamanan pesan ketika pesan pesan dikirim dari satu tempat ke tempat lain. Salah satu aspek keamanan yang perlu dijamin dalam satu dokumen, baik konvensional maupun digital adalah kerahasiaannya (*confidentiality*). Kerahasiaan adalah layanan yang digunakan untuk menjaga informasi dari setiap pihak yang tidak berwenang untuk mengaksesnya. Dengan demikian informasi hanya akan dapt diakses oleh pihak-pihak yang berhak saja.

Teknik kriptografi dapat dimanfaatkan untuk menjamin keamanan dokumen. Salah satu yang dapat dimanfaatkan adalah enkripsi dan dekripsi data

atau dengan kata lain menyandikan data sehingga hanya orang yang bersangkutan saja yang mengetahui isi data tersebut. Kriptografi *Electronic Code Book* merupakan algoritma yang kuat dan sampai saat ini dinyatakan aman.

Mode ECB merupakan mode yang paling sederhana. ECB beroperasi dengan memecah teks asli berukuran *N x n* bit menjadi *N* blok dengan tiap blok berukuran *n* bit (sesuai dengan ukuran blok sistem penyandian), kemudian tiap blok disandi dengan kunci, dan algoritma enkripsi yang sama. Untuk dekripsi dilakukan hal yang sama hanya saja menggunakan algoritma dekripsi.

Berdasarkan berbagai pertimbangan tersebut maka dalam penyusunan skripsi, saya memilih judul **" Disain Sistem Keamanan Kriptografi *Electronic Code Book (ECB)* Pada Data *Login* "**.

1.2 Rumusan Masalah

Dalam pelaksanaan penelitian skripsi ini terdapat beberapa permasalahan yang menjadi titik utama pembahasan, diantaranya adalah sebagai berikut:

a. Bagaimana mengenkripsi *password* pada data *login* dengan menggunakan algoritma *Electronic Code Book* (*ECB*) ?
b. Bagaimana membangun aplikasi yang dapat mengenkripsi *password* pada data *login* menggunakan algoritma *Electronic Code Book* (*ECB*) menggunakan bahasa pemrograman ?

1.3 Batasan Masalah

Dari rumusan masalah yang ada maka dapat diberi batasan-batasan sehingga pembahasannya lebih terarah. Adapun batasan-batasan masalah menjadi acuan dalam pengerjaan skripsi ini adalah sebagai berikut:

a. Menganalisa penyandian *password* pada data *login* menggunakan kriptografi algoritma *Electronic Code Book* (*ECB*) sebagai layanan keamanan data.
b. Perancangan program *enkripsi password* pada data *login* menggunakan algoritma *Electronic Code Book* (*ECB*) menggunakan bahasa pemrograman *Microsoft Visual Studio 2008.*
c. Jumlah maximal panjang *character* dalam *user name* dan *password* adalah 13 *character*.
d. Data yang di dekripsi adalah *user name* atau *plainteks* berdasarkan kunci atau *password*.

1.4 Tujuan Penelitian

Sesuai uraian pada latar belakang di atas, maka yang menjadi tujuan skripsi ini adalah:

a. Menganalisa bagaimana cara kerja algoritma *Electronic Code Book* (*ECB*) dalam memberi layanan kerahasiaan pengamanan *password* pada data *login*.
b. Untuk membangun suatu aplikasi program yang dapat mengenkripsi *password* dalam menjaga keamanan data *login* pada basis data menggunakan algoritma *Electric Code Book* (*ECB*).

1.5 Manfaat Penelitian

Adapun manfaat yang bisa di dapat dari penelitian ini adalah sebagai berikut:

a. Agar dapat mengamankan data *password* sehingga tidak bisa diketahui oleh orang lain.

b. Untuk membantu mahasiswa maupun masyarakat dalam pengamanan data.

1.6 Metode Penelitian

Metodologi penelitian merupakan suatu cara atau teknik yang sistematik untuk mengerjakan suatu kasus. Langkah-langkah metodologi penelitian yang digunakan dalam pengerjaan Skripsi ini adalah sebagai berikut:

1. Studi *Literature*
   a. Mempelajari materi mata kuliah kriptografi dengan algoritma *Electronic Code Book* (*ECB*).
   b. Mempelajari pemrograman *Microsoft Visual Studio 2008.*
2. Pengumpulan Data
   a. Mengambil informasi tentang kriptografi dengan algoritma *Electronic Code Book* (*ECB*) melalui *internet*.
3. Melakukan analisa sistem yang digunakan dalam pembuatan aplikasi
   a. Perancangan sistem.
   b. Pengkodean.
   c. Impelentasi dan pengujian terhadap sistem yang dibangun dengan bahasa program *Microsoft Visual Studio 2008.*

1.7 Sistematika Penulisan

Adapun sistematika penulisan Skripsi ini terdiri dari lima bab yaitu:

BAB I PENDAHULUAN

Secara garis besar bab ini berisi tentang latar belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, Metode Penelitian dan Sistematika Penulisan.

BAB II LANDASAN TEORITIS

Bab ini menjelaskan Pengertian Kriptografi, Pengenalan kriptografi dan teori-teori yang dapat mendukung pembuatan dan penyelesaian Skripsi ini.

BAB III ANALISIS DAN PERANCANGAN

Pada bab ini menjelaskan secara umum tentang penganalisaan mengenai algoritma *Electronic Code Book (ECB)* dalam merahasiakan data dan perancangan aplikasi perangkat lunak dalam pengamanan data login pada basis data.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini akan dipaparkan sistem yang dibuat yang meliputi cara untuk menjalankan sistem dan hasil implementasi dari perancangan sistem pengamanan data login pada basis data dengan algoritma *Electronic Code Book (ECB).*

BAB V KESIMPULAN DAN SARAN

Bab ini berisikan tentang kesimpulan dan saran tentang hasil akhir dari pemecahan masalah yang telah diselesaikan atau terpecahkan.

BAB II

LANDASAN TEORI

2.1 Masalah Keamanan Informasi dan Data Komputer

2.1.1 Data dan Informasi

Data dapat di defenisikan sebagai kenyataan yang digambarkan oleh nilai-nilai, bilangan-bilangan, untaian karakter atau simbol-simbol yang membawa arti tertentu. Untuk mengetahui lebih dalam mengenai data, berikut pengertian data dari beberapa ahli:

1. Menurut *Antony* dan *Dearden*, "Data adalah bentuk jamak dari bentuk tunggal datum atau data-item".
2. Menurut Jogiyanto (Analisis dan Desain Sistem Informasi : 8) "Data merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata".

Informasi sendiri dapat di defenisikan sebagai hasil dari pengolahan data dalam bentuk yang lebih berguna bagi penerimanya, yang digunakan sebagai alat bantu dalam pengambilan keputusan. Untuk mengetahui lebih dalam mengenai informasi, berikut pengertian informasi dari beberapa ahli:

1. Menurut Jogiyanto HM (1999 : 692) "Informasi dapat didefenisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan".

2. Menurut Bodnar (2000 : 1) “Informasi adalah data yang diolah sehingga dapat disajikan dasar untuk mengambil keputusan yang tepat”.

2.1.2 Keamanan Komputer

Keamanan komputer adalah berhubungan dengan pencegahan diri dan deteksi terhadap tindakan pengganggu yang tidak dikenali dalam sistem komputer. Dalam keamanan sistem komputer yang perlu kita lakukan adalah untuk mempersulit orang lain untuk mengganggu sistem yang kita pakai, baik itu kita menggunakan komputer yang sifatnya *stand alone*, jaringan *local* maupun global. Kita harus memastikan sistem bisa berjalan dengan baik dan kondusif, selain itu program aplikasinya masih bisa dipakai tanpa masalah.

1. Beberapa hal yang menjadikan kejahatan komputer terus terjadi dan cenderung meningkat adalah sebagai berikut:
   a. Meningkatnya pengguna komputer dan *internet*.
   b. Banyaknya *software* yang pada awalnya digunakan untuk melakukan *audit* sebuah sistem dengan cara mencari kelemahan dan celah yang mungkin ada disalah gunakan untuk melakukan *scanning* sistem orang lain.
   c. Banyaknya *software-software* untuk melakukan penyusupan yang tersedia di internet yang bisa di *download* secara gratis.
   d. Meningkatnya kemampuan pengguna komputer dan *internet*.
   e. Semakin banyaknya perusahaan yang menghubungkan jaringan LAN mereka ke Internet.
   f. Meningkatnya aplikasi bisnis yang menggunakan internet.
   g. Banyaknya software yang mempunyai kelemahan (*bugs*).

2. Ada beberapa hal yang bisa menjawab pertanyaan mengapa kita perlu mengamankan sistem computer, antara lain:
   a. Menghindari resiko penyusupan, kita harus memastikan bahwa sistem tidak kemasukaan penyusup yang bisa membaca, menulis dan menjalankan program-program yang bisa mengganggu atau menghancurkan sistem kita.
   b. Mengurangi resiko ancaman, hal ini biasa berlaku di institusi dan perusahaan swasta.
   c. Melindungi sistem dari kerentanan, kerentanan akan menjadikan sistem kita berpotensi untuk memberikan akses yang tidak diizinkan bagi orang lain yang tidak berhak.
   d. Melindungi sistem dari gangguan alam seperti petir dan lain-lainnya.

2.1.3 Aspek-Aspek Keamanan Komputer

Inti dari keamanan komputer adalah melindungi komputer dan jaringannya dengan tujuan mengamankan informasi yang berada di dalamnya. Keamanan komputer sendiri meliputi beberapa aspek , antara lain:

1. *Privacy*, adalah sesuatu yang bersifat rahasia (*private*). Intinya adalah pencegahan agar informasi tersebut tidak diakses oleh orang yang tidak berhak. Contohnya adalah *email* atau *file-file* lain yang tidak boleh dibaca orang lain meskipun oleh *administrator*. Pencegahan yang mungkin dilakukan adalah dengan menggunakan teknologi *enkripsi*, jadi hanya pemilik informasi yang dapat mengetahui informasi yang sesungguhnya.
2. *Confidentiality*, merupakan data yang diberikan ke pihak lain untuk tujuan khusus tetapi tetap dijaga penyebarannya. Contohnya data yang bersifat

pribadi seperti: nama, alamat, no ktp, telpon dan sebagainya. *Confidentiality* akan terlihat apabila diminta untuk membuktikan kejahatan seseorang, apakah pemegang informasi akan memberikan infomasinya kepada orang yang memintanya atau menjaga klientnya.

3. *Integrity*, penekanannya adalah sebuah informasi tidak boleh diubah kecuali oleh pemilik informasi. Terkadang data yang telah terenkripsipun tidak terjaga integritasnya karena ada kemungkinan *ciphertext* dari enkripsi tersebut berubah. Contoh: Penyerangan Integritas ketika sebuah *email* dikirimkan ditengah jalan disadap dan diganti isinya, sehingga *email* yang sampai ketujuan sudah berubah.
4. *Autentication*, ini akan dilakukan sewaktu *user login* dengan menggunakan nama *user* dan *password*-nya, apakah cocok atau tidak, jika cocok diterima dan tidak akan ditolak. Ini biasanya berhubungan dengan hak akses seseorang, apakah dia pengakses yang sah atau tidak.
5. *Availability*, aspek ini berkaitan dengan apakah sebuah data tersedia saat dibutuhkan/diperlukan. Apabila sebuah data atau informasi terlalu ketat pengamanannya akan menyulitkan dalam akses data tersebut. Disamping itu akses yang lambat juga menghambat terpenuhnya aspek *availability*. Serangan yang sering dilakukan pada aspek ini adalah *denial of service* (DOS), yaitu penggagalan *service* sewaktu adanya permintaan data sehingga komputer tidak bisa melayaninya. Contoh lain dari *denial of service* ini adalah mengirimkan *request* yang berlebihan sehingga menyebabkan komputer tidak bisa lagi menampung beban tersebut dan akhirnya komputer *down*.

2.1.4 Ancaman Keamanan

Ada begitu banyak peristiwa pertukaran informasi setiap detik di internet. Pertukaran informasi tersebut tentu tak lepas dari terjadinya pencurian informasi oleh pihak-pihak yang tidak bertanggung jawab. Beberapa ancaman keamanan terhadap informasi adalah:

1. *Interruption*: Merupakan suatu ancaman terhadap ketersediaan informasi; data yang berada dalam sistem computer dirusak atau dihapus sehingga saat diperlukan, data atau informasi tersebut sudah tidak ada lagi.
2. *Interception*: Merupakan ancaman terhadap kerahasiaan (*secrecy*). Informasi yang ada disadap atau orang yang tidak berhak bisa mengakses computer tempat informasi tersebut disimpan.
3. *Modifikasi*: Merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil menyadap lalu lintas pengiriman informasi, lalu mengubahnya sesuai keinginan orang tersebut.
4. *Fabrication*: Merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil meniru (memalsukan) suatu informasi yang ada sehingga orang yang menerima informasi tersebut menyangka informasi tersebut berasal dari orang yang dikehendaki oleh si penerima informasi tersebut (Ariyus, 2009:10).

2.2 Kriptografi

Kriptografi berasal dari bahasa Yunani. Menurut bahasa tersebut kata kriptografi dibagi menjadi dua, yaitu *krypto* dan *graphia*. *Krypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Menurut terminologinya kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika dikirim dari suatu

tempat ke tempat yang lain. Dalam perkembangannya, kriptografi juga digunakan untuk mengidentifikasikan pengiriman pesan dengan tanda tangan digital dan keaslian pesan dengan sidik jari digital (*fingerprint*) (Ariyus, 2006:77).

2.2.1 Komponen Kriptografi

Pada dasarnya, kriptografi terdiri dari beberapa komponen seperti:

1. *Enkripsi: Enkripsi* merupakan hal yang sangat penting dalam *cryptography* sebagai pengamanan atas data yang dikirimkan agar rahasianya terjaga. Pesan aslinya disebut *plaintext* yang diubah menjadi kode-kode yang tidak dimengerti. *Enkripsi* bisa diartikan sebagai *chiper* atau kode. Seperti ketika kita tidak mengerti akan arti sebuah kata, kita bisa melihatnya di dalam kamus atau daftar istilah. Berbeda dengan *enkripsi*, untuk mengubah *plaintext* ke bentuk *ciphertext* digunakan algoritma yang bisa mengkodekan data yamg diinginkan.
2. *Dekripsi: Dekripsi* merupakan kebalikan dari *enkripsi*, pesan yang telah di *enkripsi* dikembalikan ke bentuk asalnya (*plaintext*), yang disebut *dekripsi* pesan. Algoritma yang digunakan untuk *dekripsi* tentu berbeda dengan yang digunakan untuk *enkripsi*.
3. Kunci: Kunci yang dimaksud di sini adalah kunci yang dipakai untuk melakukan *enkripsi* dan *dekripsi*. Kunci terbagi menjadi dua bagian, yaitu kunci pribadi (*private key*) dan kunci umum (*public key*).
4. *Chipertext:* Merupakan suatu pesan yang sudah melalui proses *enkripsi*. Pesan yang ada pada *chipertext* tidak bisa dibaca karena berisi karakter-karakter yang tidak memiliki makna (arti).

5. *Plaintext*: Sering juga disebut *cleartext*; merupakan suatu pesan bermakna yang ditulis atau diketik dan *plaintext* itulah yang akan diproses menggunakan algoritma *cryptography* agar menjadi *chipertext*.
6. Pesan: Pesan bisa berupa data atau informasi yang dikirim (melalui kurir, saluran komunikasi data, dsb) atau yang disimpan di dalam media perekaman (kertas, *storage*, dsb).
7. *Cryptanalysis*: Bisa diartikan sebagai analisis sandi atau suatu ilmu untuk mendapatkan *plaintext* tanpa harus mengetahui kunci secara wajar. Jika suatu *chipertext* berhasil menjadi *plaintext* tanpa menggunakan kunci yang sah, maka proses tersebut dinamakan *breaking code* yang dilakukan oleh para *cryptanalys*. Analisis sandi juga mampu menemukan kelemahan dari suatu algoritma *cryptography* dan akhirnya bisa menemukan kunci atau *plaintext* dari *chipertext* yang di *enkripsi* menggunakan algoritma tertentu (Ariyus, 2009:19).

2.2.2 Tujuan Kriptografi

Dari paparan awal dapat dirangkumkan bahwa kriptografi bertujuan untuk memberi layanan keamanan, Ada empat tujuan mendasar dari kriptografi yang juga merupakan aspek keamanan informasi, yaitu sebagai berikut:

1. Kerahasiaan, adalah aspek yang berhubungan dengan penjagaan isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka informasi yang telah dienkripsi.
2. Integritas data, adalah aspek yang berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang

tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

3. Autentikasi, adalah aspek yang berhubungan dengan identifikasi atau pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
4. *Non-repudiation* (menolak penyangkalan), adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman suatu informasi oleh yang mengirimkan, atau harus dapat membuktikan bahwa suatu pesan berasal dari seseorang, apabila ia menyangkal mengirim informasi tersebut.

2.2.3 Serangan Terhadap Kriptografi

Di bawah ini dijelaskan beberapa macam penyerangan terhadap pesan yang sudah dienkripsi, berdasarkan ketersediaan data yang ada, dan tingkat kesulitannya bagi penyerang, dimulai dari yang paling sulit adalah :

1. *Ciphertext only attack*, penyerang hanya mendapatkan *ciphertext* dari sejumlah pesan yang seluruhnya telah dienkripsi menggunakan algoritma yang sama. Sehingga, metode yang digunakan untukmemecahkannya adalah *exhaustive key search,* yaitu mencoba semua kemungkinan yang ada untuk menemukan kunci.
2. *Known plaintext attack,* dimana penyerang selain mendapatkan sandi, juga mendapatkan pesan asli. Terkadang disebut pula *clear-text attack.*

3. *Choosen plaintext attack*, sama dengan *known plaintext attack*, namun penyerang bahkan dapat memilih penggalan mana dari pesan asli yang akan disandikan. Serangan jenis ini lebih hebat daripada *known-plaintex*t attack, karena kriptoanalis dapat memilih *plainteks* tertentu untuk dienkripsikan, yaitu *plainteks-plainteks* yang lebih mengarahkan penemuan kunci
4. *Chosen ciphertext attack.* Pada tipe ini, kriptoanalis dapat memilih *cipherteks* yang berbeda untuk didekripsi dan memiliki akses atas *plaintext* yang didekripsi.
5. *Chosen key attack.* Kriptoanalis pada tipe penyerangan ini memiliki pengetahuan tentang hubungan antara kunci-kunci yang berbeda dan memilih kunci yang tepat untuk mendekripsi pesan.
6. *Rubber hose cryptanalysis*. Pada tipe penyerangan ini, kriptoanalis mengancam, menyiksa, memeras, memaksa, atau bahkan menyogok seseorang hingga mereka memberikan kuncinya. Ini adalah cara yang paling ampuh untuk mendapatkan kunci.
7. *Adaptive chosen plaintext attack,* Penyerangan tipe ini merupakan suatu kasus khusus *chosen-plaintext attack*. Kriptoanalis tidak hanya dapat memilih *plainteks* yang dienkripsi, ia pun memiliki kemampuan untuk memodifikasi pilihan berdasarkan hasil enkripsi sebelumnya. Dalam *chosen-plaintext attack*, kriptoanalis mungkin hanya dapat memiliki *plainteks* dalam suatu blok besar untuk dienkripsi; dalam *adaptive-chosen-plaintext attack* ini ia dapat memilih blok *plainteks* yang lebih kecil dan kemudian memilih yang lain berdasarkan

hasil yang pertama, proses ini dapat dilakukannya terus menerus hingga ia dapat memperoleh seluruh informasi.

Berdasarkan bagaimana cara dan posisi seseorang mendapatkan pesan-pesan dalam saluran komunikasi, penyerangan dapat dikategorikan menjadi:

1. *Spoofing*, Penyerang misalnya Angga bisa menyamar menjadi Adi. Semua orang dibuat percaya bahwa Angga adalah Adi. Penyerang berusaha meyakinkan pihak-pihak lain bahwa tak ada salah dengan komunikasi yang dilakukan, padahal komunikasi itu dilakukan dengan sang penipu/penyerang. Contohnya jika orang memasukkan PIN ke dalam mesin ATM palsu yang benar-benar dibuat seperti ATM asli tentu sang penipu bisa mendapatkan PIN-nya dan copy pita magentik kartu ATM milik sang nasabah. Pihak bank tidak tahu bahwa telah terjadi kejahatan.
2. *Man in the middle,* Jika *spoofing* terkadang hanya menipu satu pihak, maka dalam skenario ini, saat Adi hendak berkomunikasi dengan Badu, Angga di mata Adi seolah-olah adalah Badu, dan Angga dapat pula menipu Badu sehingga Angga seolah-olah adalah Adi. Angga dapat berkuasa penuh atas jalur komunikas ini, dan bisa membuat berita fitnah.
3. *Sniffing*, secara harfiah berarti mengendus, tentunya dalam hal ini yang diendus adalah pesan (baik yang belum ataupun sudah dienkripsi) dalam suatu saluran komunikasi. Hal ini umum terjadi pada saluran *public* yang tidak aman. Sang pengendus dapat merekam pembicaraan yang terjadi.

4. *Replay attack*, Jika seseorang bisa merekam pesan-pesan *handshake* (persiapan komunikasi), ia mungkin dapat mengulang pesan-pesan yang telah direkamnya untuk menipu salah satu pihak.

Beberapa metode penyadapan data yang biasanya dilakukan oleh penyerang:

1. *Electromagnetic eavesdropping,* Penyadap mencegat data yang ditransmisikan melalui saluran *wireless*, misalnya radio dan *microwive*.
2. *Acoustic Eavesdropping*, Menangkap gelombang suara yang dihasilkan oleh suara manusia.
3. *Wiretapping,* Penyadap mencegat data yang ditransmisikan pada saluran kabel komunikasi dengan menggunakan sambungan perangkat keras.

Jenis-jenis serangan berdasarkan teknik yang digunakan untuk menemukan kunci:

1. *Brute force attack* atau *Exhaustive attack,* Serangan *brute force* adalah sebuah teknik serangan yang menggunakan percobaan terhadap semua kunci yang mungkin untuk mengungkap *plainteks*/kunci.
2. *Analytical attack,* Pada jenis serangan ini, kriptanalis tidak mencoba-coba semua kemungkinan kunci tetapi menganalisis kelemahan algoritma kriptografi untuk mengurangi kemungkinan kunci yang tidak mungkin ada.

2.2.4 Kompleksitas Serangan

Kompleksitas serangan kriptografi dapat diukur dengan beberapa cara, yaitu:

1. Kompleksitas data (*data complexity*)

   Jumlah data (*plainteks* dan *cipherteks*) yang dibutuhkan sebagai masukan untuk serangan. Semakin banyak data yang dibutuhkan untuk melakukan serangan, semakin kompleks serangan tersebut, yang berarti semakin bagus sistem kriptografi tersebut.

2. Kompleksitas waktu (*time complexity*)

   Waktu yang dibutuhkan untuk melakukan serangan. Semakin lama waktu yang dibutuhkan untuk melakukan serangan, berarti semakin bagus kriptografi tersebut.

3. Kompleksitas ruang memori (*space/storage complexity*)

   Jumlah memori yang dibutuhkan untuk melakukan serangan. Semakin banyak memori yang dibutuhkan untuk melakukan serangan, berarti semakin bagus sistem kriptografi tersebut.

2.2.5 Prinsip Menentukan Algoritma Kriptografi

Pengetahuan mengenai serangan terhadap kriptografi sangatlah penting untuk meningkatkan efektifitas dan kualitas algoritma penyandian yang digunakan. Prinsip yang dipakai dalam menentukan penggunaan suatu algoritma kriptografi adalah :

1. Persamaan matematis yang menggambarkan operasi algoritma kriptografi yang dibuat sangat kompleks sehingga algoritma tidak mungkin dipecahkan secara analitik.
2. Biaya untuk memecahkan *ciphertext* melampaui nilai informasi yang terkandung di dalam *ciphertext* tersebut.

3. Waktu yang diperlukan untuk memecahkan *ciphertext* tersebut melampaui lamanya waktu informasi tersebut harus dijaga kerahasiaannya.

2.2.6 Algoritma Kriptografi Klasik

Sebelum komputer ada, kriptografi dilakukan dengan menggunakan pensil dan kertas. Algoritma kriptografi (*cipher*) yang digunakan saat itu dinamakan juga algoritma klasik, adalah berbasis karakter, yaitu enkripsi dan dekripsi dilakukan pada setiap karakter pesan. Semua algoritma klasik termasuk ke dalam sistem kriptografi simetris dan digunakan jauh sebelum kriptografi kunci *public* ditemukan. Kriptografi klasik memiliki beberapa ciri:

1. Berbasis karakter
2. Menggunakan pena dan kertas saja, belum ada komputer
3. Termasuk ke dalam kriptografi kunci simetris

Pada dasarnya, algoritma kriptografi klasik dapat dikelompokkan ke dalam dua macam *cipher*, yaitu:

1. *Cipher* substitusi (*substitution cipher*)

   Di dalam *cipher* substitusi setiap unit *plainteks* diganti dengan satu unit *cipherteks*. Satu "unit" di sini berarti satu huruf, pasangan huruf, atau dikelompokkan lebih dari dua huruf. Algoritma substitusi tertua yang diketahui adalah *Caesar cipher* yang digunakan oleh kaisar Romawi, *Julius Caesar* (sehingga dinamakan juga *Caesar cipher*), untuk pesan yang dikirimkan kepada gubernurnya.

Caesar cipher mudah dipecahkan dengan *exhaustive key* karena jumlah kuncinya sangat sedikit (hanya ada 26 kunci). *Caesar cipher* termasuk *cipher* abjad tunggal.

2. Cipher transposisi (*transposition cipher*)

   Pada cipher transposisi, huruf-huruf di dalam *plainteks* tetap saja, hanya saja urutannya diubah. Dengan kata lain algoritma ini melakukan *transpose* terhadap rangkaian karakter di dalam teks. Nama lain untuk metode ini adalah permutasi atau pengacakan (*scrambling*), karena *transpose* setiap karakter di dalam teks sama dengan mempermutasikan karakter-karakter tersebut.

2.2.7 Algoritma Kriptografi Modern

Algoritma kriptografi modern umumnya beroperasi dalam mode bit ketimbang mode karakter (seperti yang dilakukan pada *cipher* substitusi atau *cipher* transposisi dari algoritma kriptografi klasik).

Operasi dalam mode bit berarti semua data dan informasi (baik kunci, *plainteks*, maupun *cipherteks*) dinyatakan dalam rangkaian (*string*) bit biner, 0 dan 1.

Algoritma enkripsi dan dekripsi memproses semua data dan informasi dalam bentuk rangkaian bit. Rangkaian bit yang menyatakan *plainteks* dienkripsi menjadi *cipherteks* dalam bentuk rangkaian bit, demikian sebaliknya.

Perkembangan algoritma kriptografi modern berbasis bit didorong oleh pengguna komputer digital yang merepresentasikan data dalam bentuk biner.

Algoritma kriptografi yang beroperasi dalam mode bit dapat dikelompokkan menjadi dua katagori:

1. *Cipher* aliran (*stream cipher*)

   Algoritma kriptografi beroperasi pada *plainteks/cipherteks* dalam bentuk bit tunggal, yangdalam hal ini rangkaian bit dienkripsikan/didekripsikan bit per bit.

2. *Cipher* blok (*block cipher)*

   Algoritma kriptografi beroperasi pada *plainteks/cipherteks* dalam bentuk blok bit, yang dalam hal ini rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya.

2.3 Algoritma *Electronic Code Book* (ECB)

Mode ECB merupakan mode yang paling sederhana. ECB beroperasi dengan memecah teks asli berukuran *N x n* bit menjadi *N* blok dengan tiap blok berukuran *n* bit (sesuai dengan ukuran blok sistem penyandian), kemudian tiap blok disandi dengan kunci, dan algoritma enkripsi yang sama. Untuk dekripsi dilakukan hal yang sama hanya saja menggunakan algoritma dekripsi.

Pada mode operasi ECB, jika teks asli memiliki ukuran yang bukan tepat kelipatan ukuran blok sistem penyandian maka diperlukan *padding*. *Padding* merupakan penembahan beberapa *byte* pada blok terakhir teks asli agar memiliki ukuran yang tepat kelipatan ukuran blok. Larik *byte* yang digunakan untuk *padding* bisa berupa *byte* kosong atau sebuah larik dengan konstan misalnya *byte* 80 yang diikuti *byte* 00.

Cara kedua untuk menggenapi panjang teks asli sehingga berukuran tepat kelipatan ukuran blok adalah teknik *cipherteks stealing*. Dengan teknik ini ukuran teks sandi sama dengan ukuran teks asli tanpa perlu tambahan *padding*. Misalnya

2 blok terakhir teks awal adalah *PN-1*, dan *PN*. Ukuran blok adalah *n* bit, dan ukuran blok terakhir (*PN*) adalah *m* bit dengan $m < n$ lakukan hal berikut:

1. Lakukan $X = \boldsymbol{enc}k(Pn\text{-}1)$ (enkripsi blok *PN-1*).
2. Tetapkan *CN* = *headm*(X) (Fungsi *headm*(X) adalah mengambil *m* bit terdepan X).
3. Hitung $Y = PN|tail$n-m(X) (fungsi *head*n-m(X) adalah mengambil *n-m* bit terbelakang *Y*).

Tetapkan *CN-1* = **enc***k* (Y) (Sadikin, 2012:194).

2.3.1 Pertimbangan Keamanan

Algoritma enkripsi mode operasi ECB beroperasi pada *N* blok secara terpisah dengan kunci rahasia *K* yang sama. Tidak ada masukan berasal dari blok sebelumnya terhadap penyandian blok ke-*i*. Oleh karena itu, bila suatu blok memiliki kesalahan bit maka kesalahan bit itu hanya berpengaruh pada blok itu.

Namun ada 2 hal yang menjadi kelemahan utama mode operasi ECB, yaitu:

1. Pola pada teks asli tetap bertahan pada teks sandi. Hal ini disebabkan hasil enkripsi suatu blok teks asli yang sama menghasilkan blok teks sandi yang sama.
2. Karena blok teks asli yang sama memiliki blok teks sandi sama, seorang penyadap dapat menggunakan ulang blok teks sandi untuk keperluannya.

Karena kelemahan-kelemahannya itu, mode operasi ECB tidak direkomendasikan untuk digunakan sebagai mode operasi penyandian pesan yang berukuran lebih dari pada ukuran blok sistem penyandian pada jaringan yang melalui kanal yang tidak aman (Sadikin, 2012:196).

2.4 Login

Login disebut juga "*logon*" atau "*sign in*" adalah istilah dalam hal keamanan komputer, yakni berupa proses pintu masuk bagi pengguna untuk mengakses sistem komputer. *Login* dimaksudkan untuk mengatur proses *identifikasi*. Dengan teknologi terkini, proses *login* semakin diperketat dengan *enkripsi* secara *hardware*, seperti proses *scan* sidik jari dan retina mata.

Login juga bisa disebut proses untuk mengakses komputer dengan memasukkan identitas dari *account* pengguna dan kata sandi guna mendapatkan hak akses menggunakan sumber daya komputer tujuan. Untuk melakukan *log* masuk ke sistem biasanya membutuhkan *account* pengguna yang digunakan sebagai identitas berupa runtutan karakter yang secara unik merujuk ke pengguna tertentu, dan kata sandi yang merupakan runtutan karakter berupa kunci yang dijaga kerahasiaannya terhadap orang lain.

2.5 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah bahasa untuk menentukan, visualisasi, mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hamper dalam semua bidang yang membutuhkan pemodelan. Adapun bagian-bagian dari UML antara lain:

1. *View, View* digunakan untuk melihat sistem yang dimudelkan dari beberapa aspek.
2. *Uses case view,* Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*.
3. *Logical view,* Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class, object* dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu.
4. *Component view,* Mendeskripsikan implementasi dan ketergantungan modul.
5. *Concurrency view,* Membagi sistem ke dalam proses dan prosesor.
6. *Deployment view,* Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya.
7. *Diagram, Diagram* berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem.

2.6 Activity Diagram

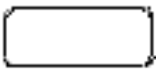
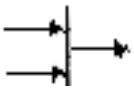
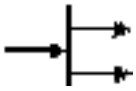
Activity Diagram merupakan *state* diagram khusus dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal proces*sing). Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram aktivitas menunjukan aktivitas sistem dalam bentuk kumpulan aksi-aksi.

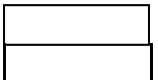
Ketika digunakan dalam pemodelan software, diagram aktivitas merepsentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar, seperti pemasangan atau kejadian – kejadian

internel. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behavior* dalam kondisi tertentu

Menurut Whitten et.al (2004,442) *diagram activity* digunakan untuk menggambarkan urutan aliran kegiatan-kegiatan dari sebuah proses bisnis atau sebuah *use case*. Diagram ini juga dapat digunakan untuk memodelkan aksi dan hasil ketika operasi berlangsung. Berikut tabel yang menyajikan notasi *Activity Diagram*.

Tabel 2.1 Simbol Activity Diagram

No	Gambar	Nama	Keterangan
1		*Start Point*	Untuk mengawali suatu kegiatan.
2		*End Point*	Untuk mengahiri suatu kegiatan
3		aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kaata kerja.
4		*Fork*	Percabangan
5		*Join*	Penghubung

6		*Decision*	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
7		*Swimlane*	Sebuah cara untuk mengelompokan activity berdasarkan actor.

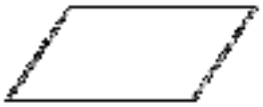
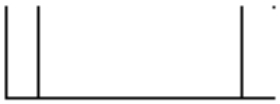
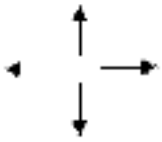
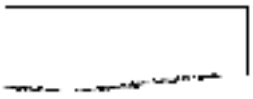
2.7 Flowchart

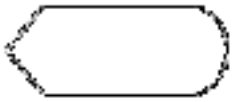
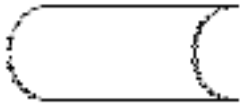
Sumber: http://bopungumn.blogspot.com/2012/03/activity-diagram

Flowchart adalah bagan (diagram) alir yang merupakan sekumpulan simbol-simbol atau skema yang menunjukkan kegiatan-kegiatan program dari awal sampai akhir. Diagram ini menggambarkan urutan-urutan atau langkah-langkah pengerjaan dari suatu algoritma. Berikut adalah penjelasan arti lambang-lambang *flowchart* atau diagram alir data.

Tabel 2.2 Simbol Flowchart

Simbol	Fungsi
	Terminal, untuk memulai atau mengakhiri program.
	Proses, suatu simbol yang menunjukkan setiap pengolahan yang dilakukan.

	*Input-output*, untuk memasukkan data ataupun menunjukkan hasil dari suatu program.
	*Decision*, suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan.
	*Predefined process*, proses suatu simbol untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur.
	*Conector*, suatu prosedur akan masuk atau keluar melalui simbol ini lembar yang sama.
	*Oof-page connector*, merupakan simbol masuk atau keluar suatu prosedur pada kertas yang lain.
Simbol	**Fungsi**
	Arus *flow* dari pada prosedur yang dapat dilakukan atas, bawah ke atas, kiri ke kanan, dan kanan ke kiri
	*Document*, merupakan simbol untuk data yang berbentuk kertas maupun untuk informasi.

	*Display*, simbol untuk *output*, ditunjukkan ke suatu *device* seperti *printer* dan sebagainya.
	Penyimpanan *file* secara sementara.

Sumber : http://id.wikipedia.org/wiki/Diagram_alir

BAB III

METODOLOGI PENELITIAN

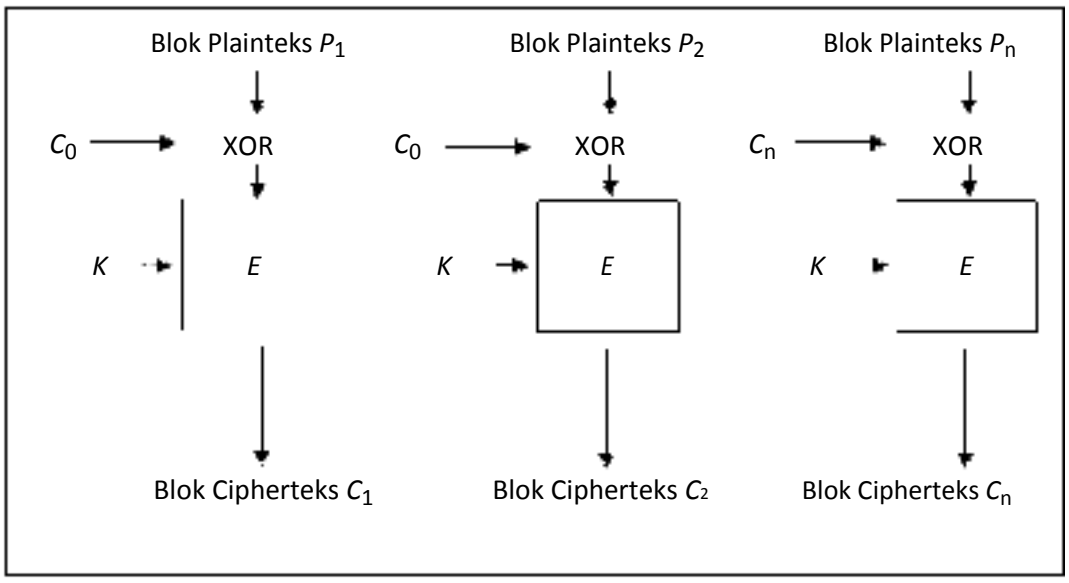
3.1 Analisis Permasalahan

Pada bagian pembahasan ini dijelaskan secara umum bagaimana cara kerja dari algoritma *Electronic Code Book (ECB)* dalam melakukan penyandian pesan *(message).* Algoritma *Electronic Code Book (ECB)* memiliki keuntungan, Karena tiap blok *plainteks* dienkripsi secara *independent*, maka kita tidak perlu mengenkripsi *file* secara *linear*. Kita dapat mengenkripsi 5 blok pertama, kemudian blok-blok di akhir, dan kembali ke blok-blok di tengah dan seterusnya. Mode *Electronic Code Book (ECB)* cocok untuk mengenkripsi arsip (file) yang diakses secara acak, misalnya arsip-arsip basis data. Jika basis data dienkripsi dengan mode *Electronic Code Book (ECB)* , maka sembarang *record* dapat dienkripsi atau didekripsi secara *independent* dari *record* lainnya (dengan asumsi setiap *record* terdiri dari sejumlah blok diskrit yang sama banyaknya). Kesalahan 1 atau lebih bit pada blok *cipherteks* hanya mempengaruhi *cipherteks* yang bersangkutan pada waktu *dekripsi*. Blok-blok *cipherteks* lainnya bila didekripsi tidak terpengaruh oleh kesalahan bit *cipherteks* tersebut.

3.2 Algoritma Sistem

Pada mode ini, setiap blok plainteks dienkripsi secara individual dan independen. Secara matematis, *enkripsi* dengan algoritma *Electronic Code Book (ECB)* dinyatakan sebagai $C_i = E_K(P_i)$ dan dekripsi sebagai $P_i = D_K(C_i)$. Yang

dalam hal ini, P_i dan C_i masing-masing blok *plainteks* dan *cipherteks* ke-*i*. Gambar 3.1 memperlihatkan *enkripsi* dua buah blok *plainteks*, P_1 dan P_2 dengan algoritma *Electronic Code Book (ECB)*, yang dalam hal ini *E* menyatakan fungsi *enkripsi* yang melakukan *enkripsi* terhadap blok *plainteks* dengan menggunakan kunci *K*.



Gambar 3.1. Skema *enkripsi* dengan algoritma *ECB*

Diketahui:

Plainteks (dalam *character*): ABDI NUGE

Kunci (dalam *character*): PANCABUDI

C_0: 00110100 (4 dalam *character*)

Tabel 3.1 Tabel Plainteks

PLAINTEKS	BINARY
A	**01000001**
B	**01010010**
D	**01001001**
I	**01000101**
(spasi)	**00100000**
N	**01001001**
U	**01010011**
G	**01001011**
E	**01000001**

Tabel 3.2 Tabel Kunci

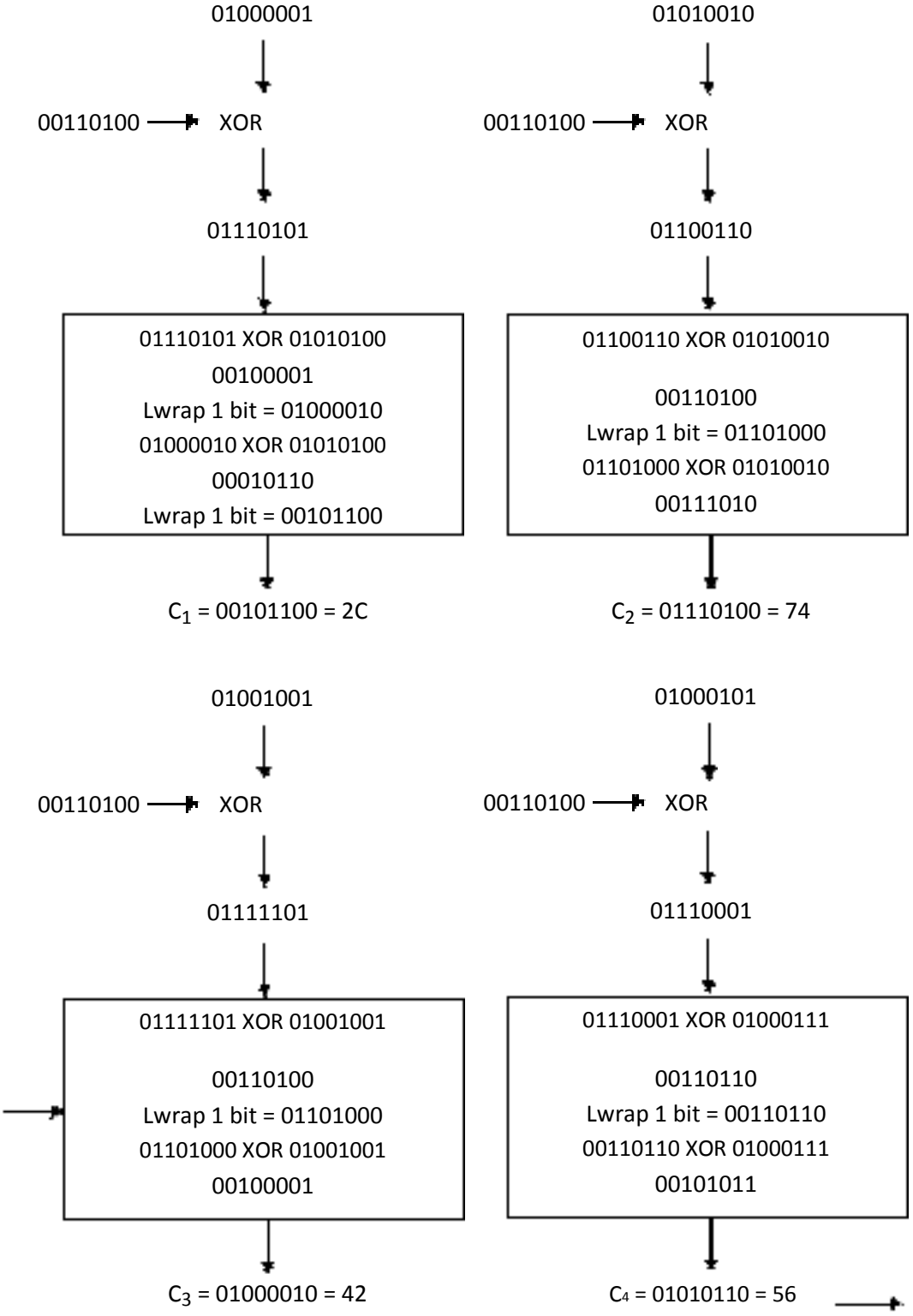
PLAINTEKS	BINARY
U	**01010100**
N	**01010010**
P	**01001001**
A	**01000111**
B	**01010101**

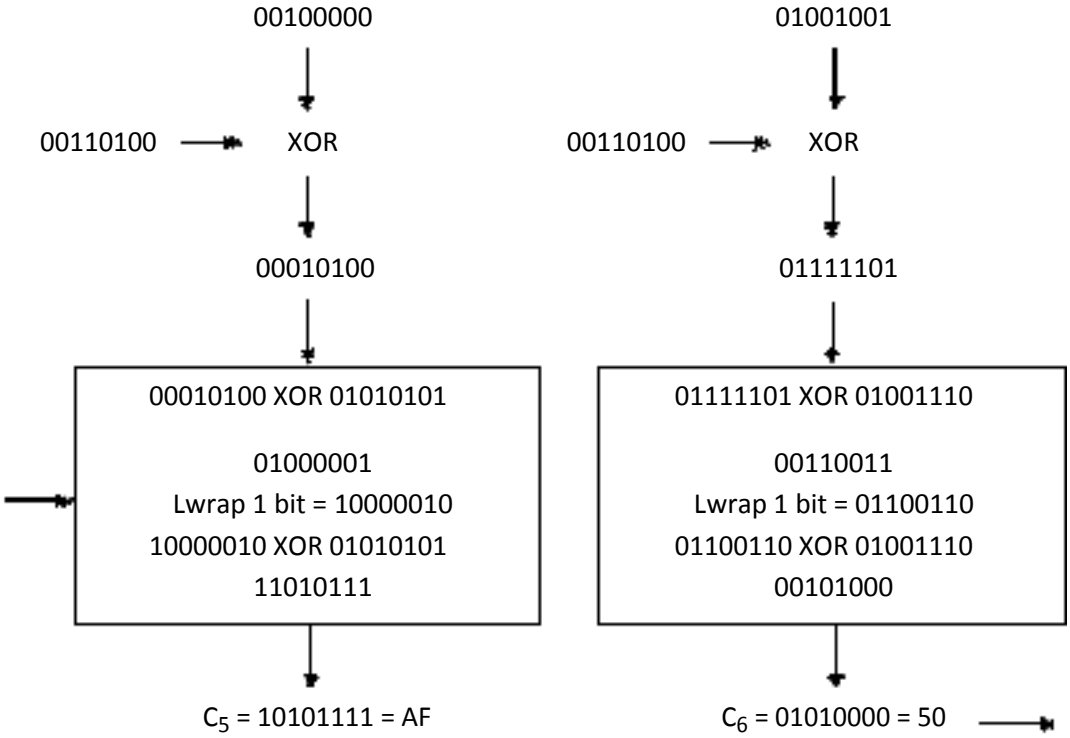
Penyelesaian:

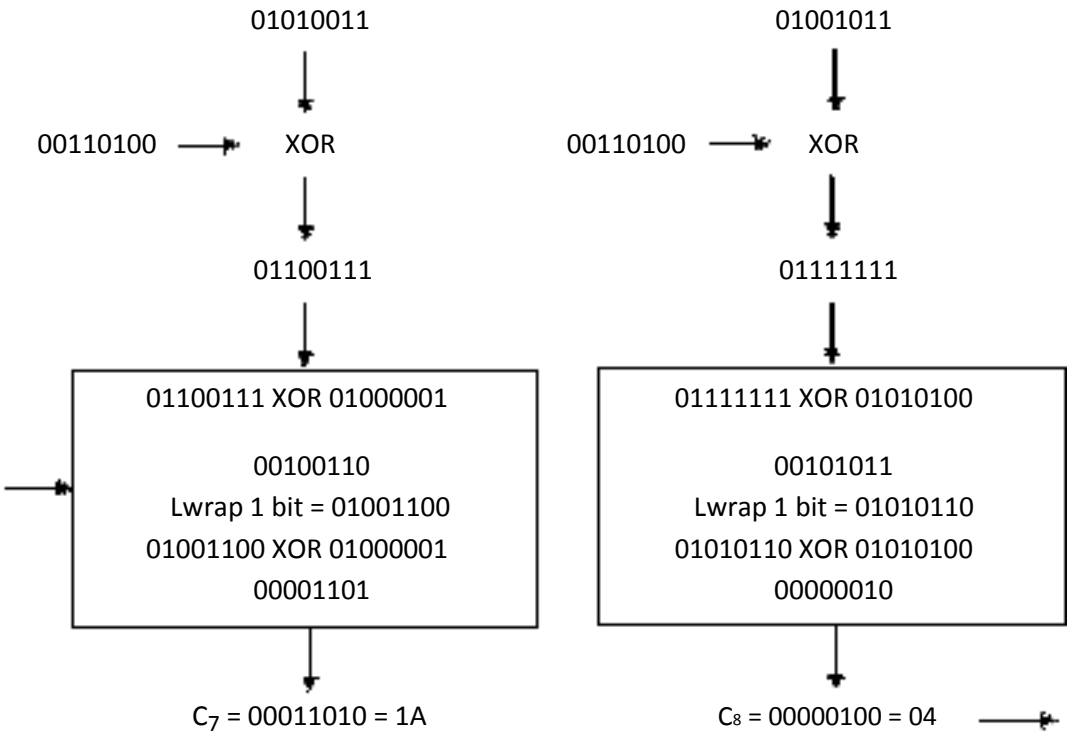
Fungsi *enkripsi E* yang digunakan adalah dengan menjadikan blok-blok *plainteks* menjadi 8 bit dan menjadikan kunci juga 8 bit dan kemudian meng-XOR-kan blok

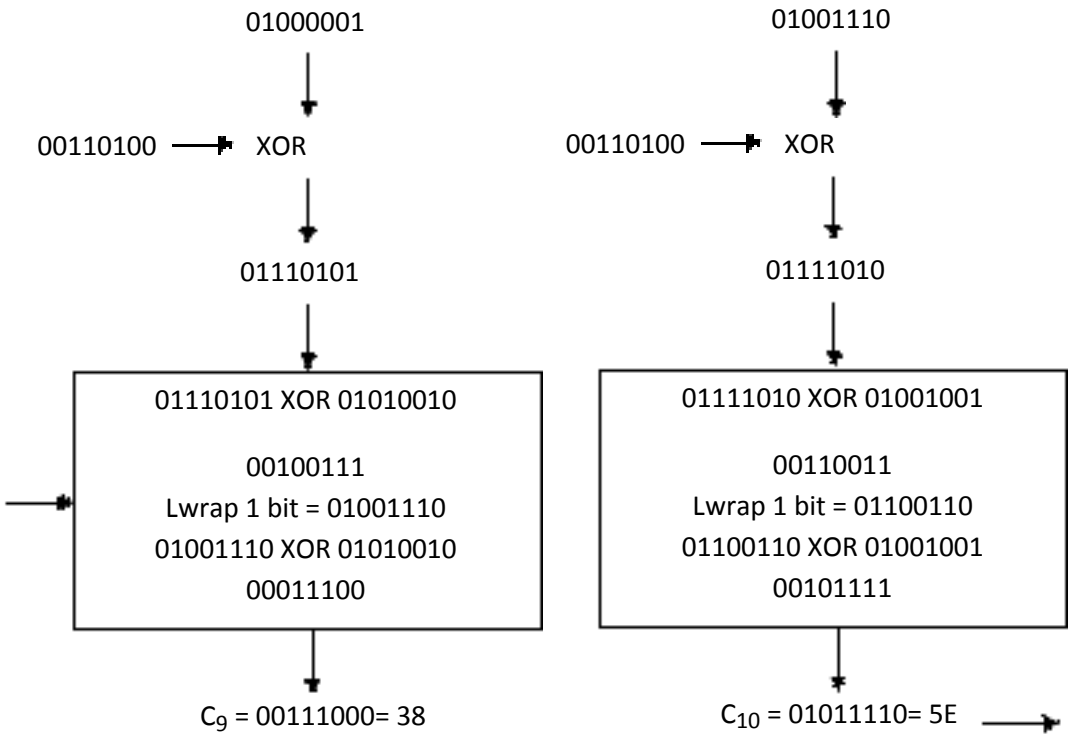
plainteks P_i dengan K, kemudian geser secara *wrapping* bit-bit dari $P_i \oplus K$ satu posisi ke kiri dan hasil *wrapping* di XOR kan kembali dengan kunci awal.

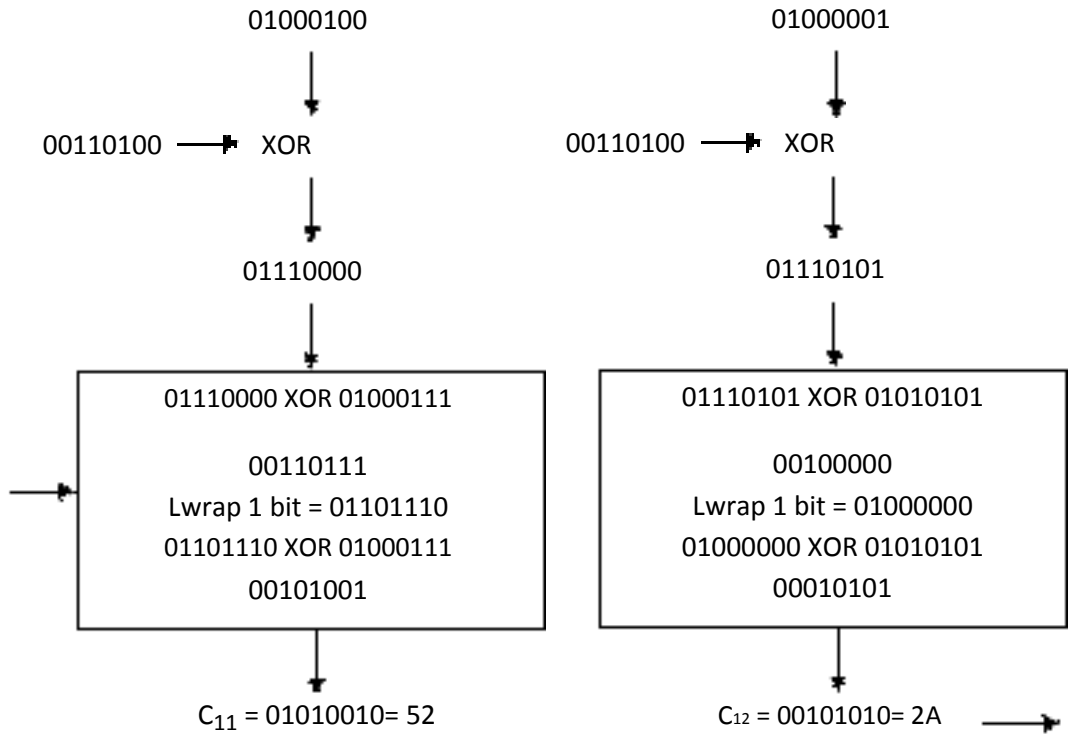
Proses enkripsi untuk setiap blok digambarkan sebagai berikut:

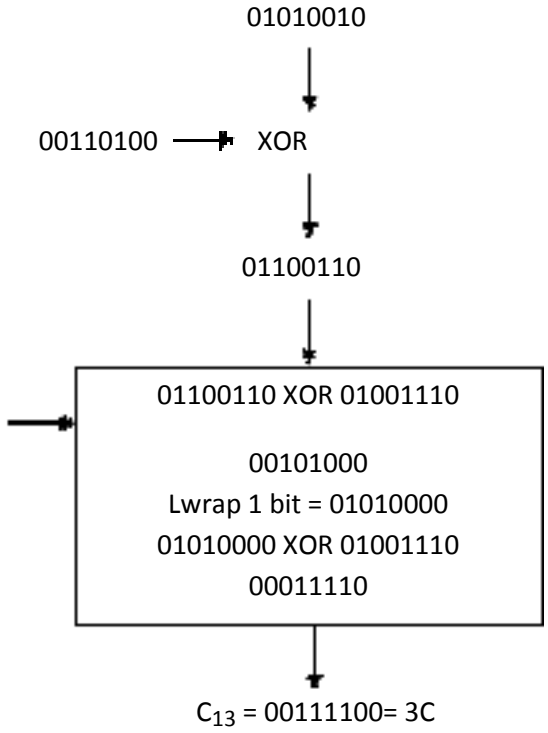



00100000
00110100
XOR
00010100
00010100 XOR 01010101
01000001
Lwrap 1 bit = 10000010
10000010 XOR 01010101
11010111
C5 = 10101111 = AF
01001001
00110100
XOR
01111101
01111101 XOR 01001110
00110011
Lwrap 1 bit = 01100110
01100110 XOR 01001110
00101000
C6 = 01010000 = 50


01010011
00110100
XOR
01100111
01100111 XOR 01000001
00100110
Lwrap 1 bit = 01001100
01001100 XOR 01000001
00001101
C7 = 00011010 = 1A
01001011
00110100
XOR
01111111
01111111 XOR 01010100
00101011
Lwrap 1 bit = 01010110
01010110 XOR 01010100
00000010
C8 = 00000100 = 04


01000001
00110100
XOR
01110101
01110101 XOR 01010010
00100111
Lwrap 1 bit = 01001110
01001110 XOR 01010010
00011100
C9 = 00111000= 38
01001110
00110100
XOR
01111010
01111010 XOR 01001001
00110011
Lwrap 1 bit = 01100110
01100110 XOR 01001001
00101111
C10 = 01011110= 5E


01000100
00110100
XOR
01110000
01110000 XOR 01000111
00110111
Lwrap 1 bit = 01101110
01101110 XOR 01000111
00101001
C11 = 01010010= 52
01000001
00110100
XOR
01110101
01110101 XOR 01010101
00100000
Lwrap 1 bit = 01000000
01000000 XOR 01010101
00010101
C12 = 00101010= 2A



Jadi, hasil *enkripsi plainteks*

01000001 01010010 01001001 01000101 00100000 01001001 01010011

01001011 01000001 01001110 01000100 01000001 01010010

(ABDI NUGE dalam *character*)

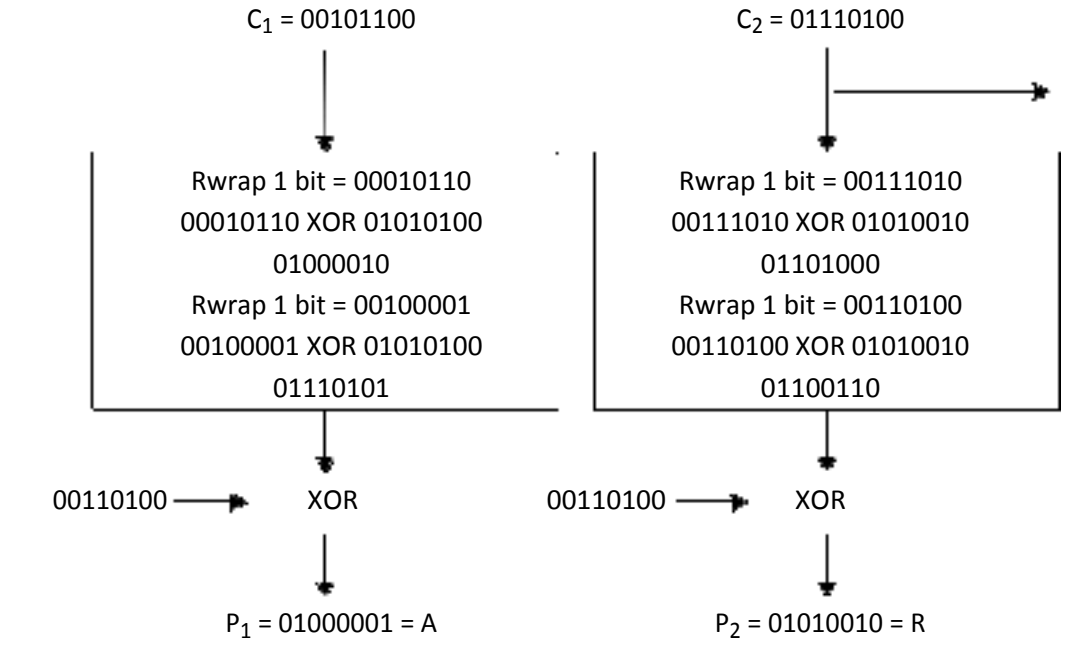
Adalah

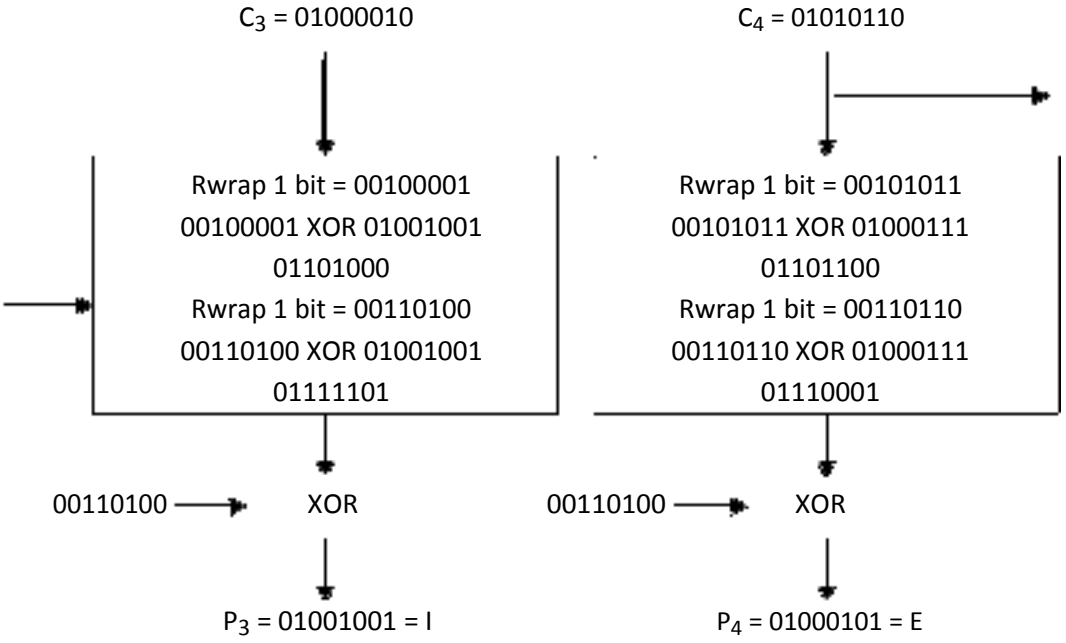
00101100 01110100 01000010 01010110 10101111 01010000 00011010

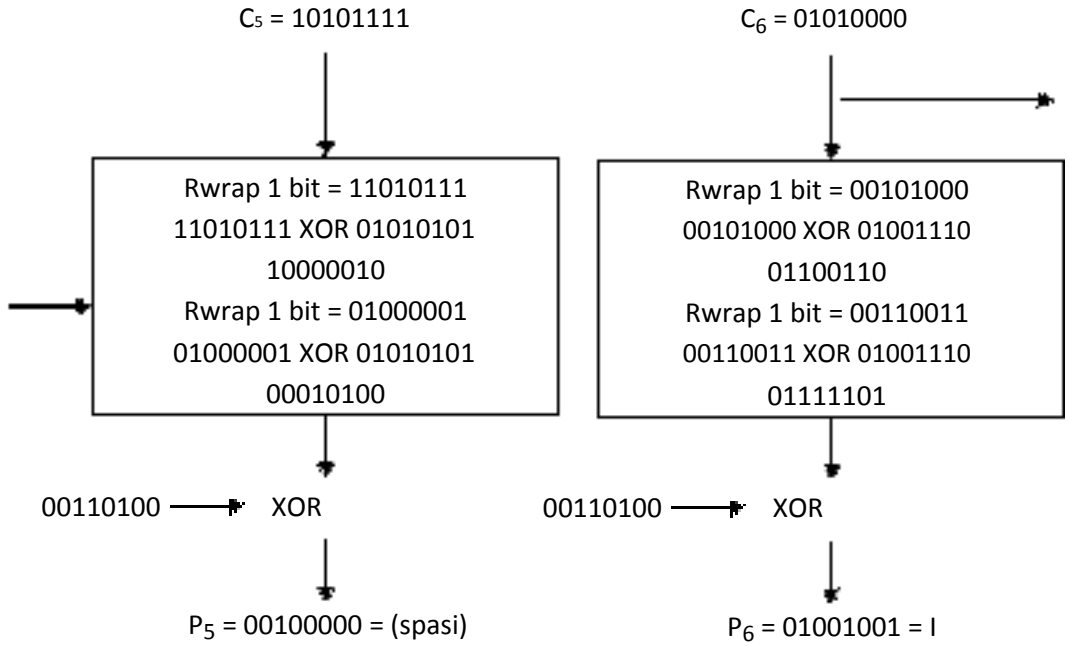
00000100 00111000 01011110 01010010 00101010 00111100

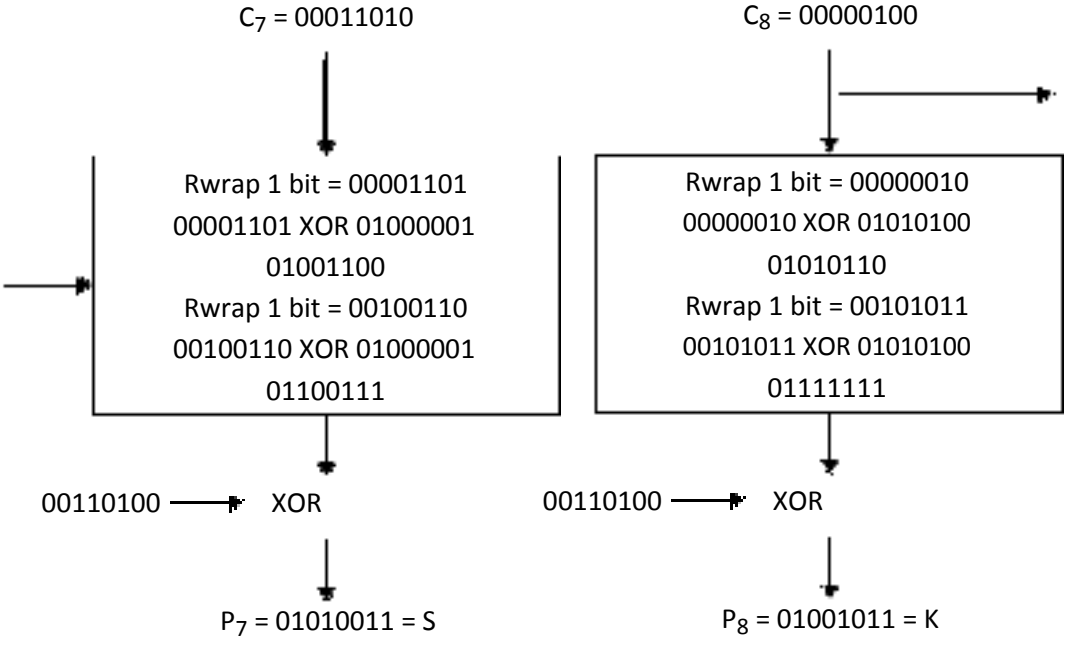
(2C744256AF501A04385E522A3C dalam notasi *HEX*)

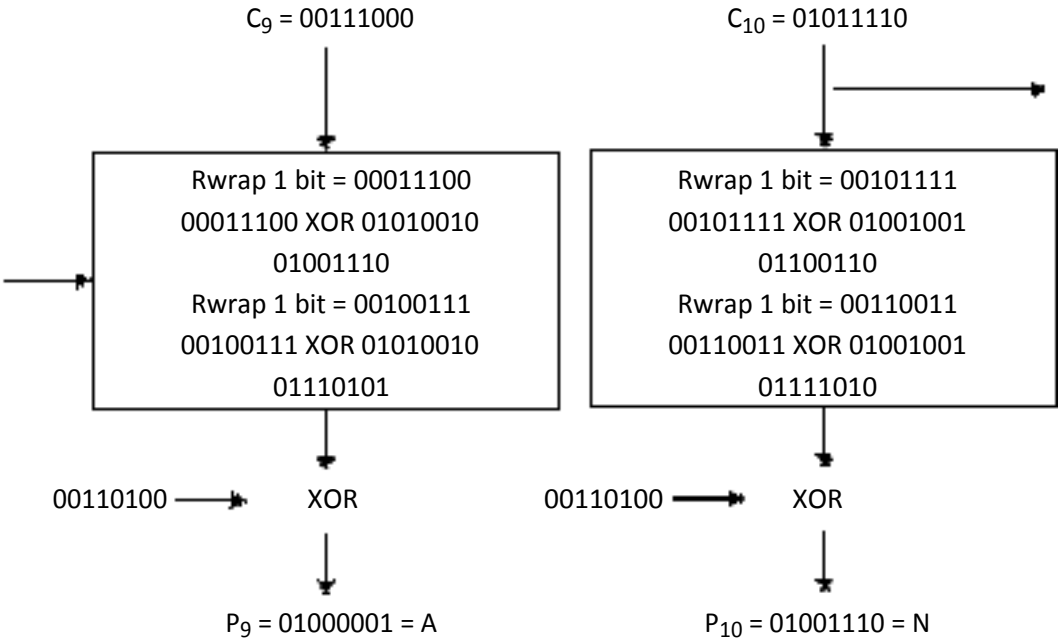
Setelah hasil *enkripsi* berhasil maka untuk mengembalikan ke *plainteks* semula dengan cara mendekripsikan hasil *enkripsi*.

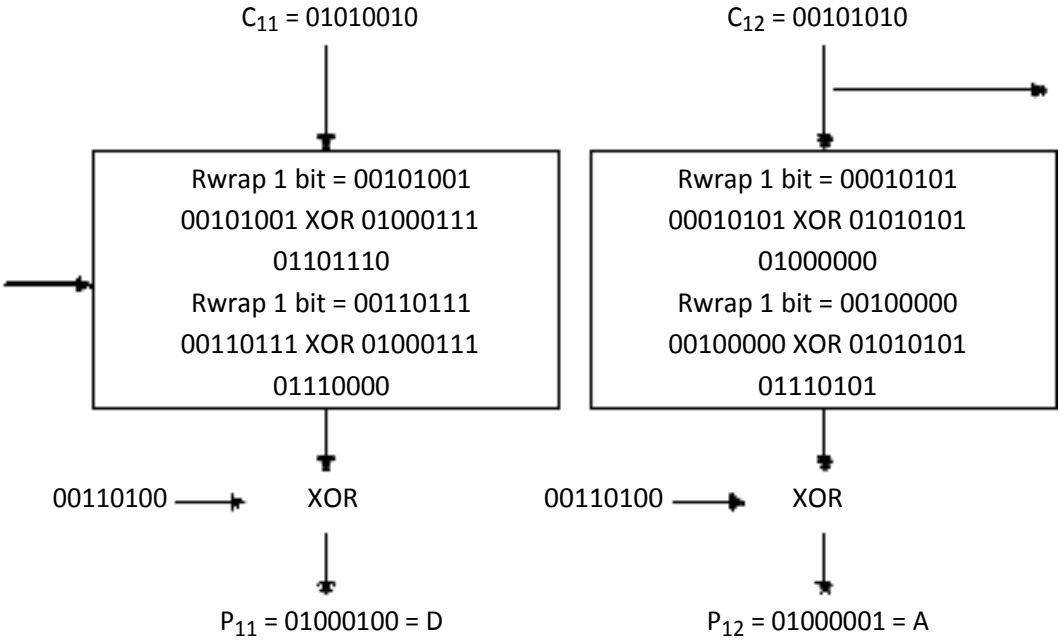

C1 = 00101100
Rwrap 1 bit = 00010110
00010110 XOR 01010100
01000010
Rwrap 1 bit = 00100001
00100001 XOR 01010100
01110101
00110100
XOR
P1 = 01000001 = A
C2 = 01110100
Rwrap 1 bit = 00111010
00111010 XOR 01010010
01101000
Rwrap 1 bit = 00110100
00110100 XOR 01010010
01100110
00110100
XOR
P2 = 01010010 = R

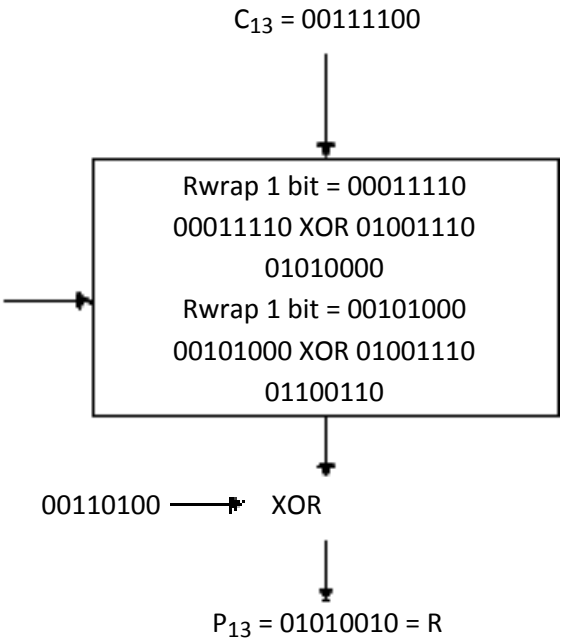

C3 = 01000010
Rwrap 1 bit = 00100001
00100001 XOR 01001001
01101000
Rwrap 1 bit = 00110100
00110100 XOR 01001001
01111101
00110100
XOR
P3 = 01001001 = I
C4 = 01010110
Rwrap 1 bit = 00101011
00101011 XOR 01000111
01101100
Rwrap 1 bit = 00110110
00110110 XOR 01000111
01110001
00110100
XOR
P4 = 01000101 = E


C5 = 10101111
Rwrap 1 bit = 11010111
11010111 XOR 01010101
10000010
Rwrap 1 bit = 01000001
01000001 XOR 01010101
00010100
00110100
XOR
P5 = 00100000 = (spasi)
C6 = 01010000
Rwrap 1 bit = 00101000
00101000 XOR 01001110
01100110
Rwrap 1 bit = 00110011
00110011 XOR 01001110
01111101
00110100
XOR
P6 = 01001001 = I


C7 = 00011010
Rwrap 1 bit = 00001101
00001101 XOR 01000001
01001100
Rwrap 1 bit = 00100110
00100110 XOR 01000001
01100111
00110100
XOR
P7 = 01010011 = S
C8 = 00000100
Rwrap 1 bit = 00000010
00000010 XOR 01010100
01010110
Rwrap 1 bit = 00101011
00101011 XOR 01010100
01111111
00110100
XOR
P8 = 01001011 = K


C9 = 00111000
C10 = 01011110
Rwrap 1 bit = 00011100
00011100 XOR 01010010
01001110
Rwrap 1 bit = 00100111
00100111 XOR 01010010
01110101
Rwrap 1 bit = 00101111
00101111 XOR 01001001
01100110
Rwrap 1 bit = 00110011
00110011 XOR 01001001
01111010
00110100
XOR
00110100
XOR
P9 = 01000001 = A
P10 = 01001110 = N


C11 = 01010010
C12 = 00101010
Rwrap 1 bit = 00101001
00101001 XOR 01000111
01101110
Rwrap 1 bit = 00110111
00110111 XOR 01000111
01110000
Rwrap 1 bit = 00010101
00010101 XOR 01010101
01000000
Rwrap 1 bit = 00100000
00100000 XOR 01010101
01110101
00110100
XOR
00110100
XOR
P11 = 01000100 = D
P12 = 01000001 = A



Jadi, hasil *dekripsi cipherteks*

00101100 01110100 01000010 01010110 10101111 01010000 00011010 00000100 00111000 01011110 01010010 00101010 00111100

(2C744256AF501A04385E522A3C dalam notasi *HEX*) Adalah

01000001 01010010 01001001 01000101 00100000 01001001 01010011 01001011 01000001 01001110 01000100 01000001 01010010

(ABDI NUGE dalam *character*)

Kata "*code book*" di dalam algoritma *Electronic Code Book (ECB)* muncul dari fakta bahwa karena blok *plainteks* yang sama selalu dienkripsi menjadi blok *cipherteks* yang sama, maka secara teoritis dimungkinkan membuat buku kode *plainteks* dan *cipherteks* yang berkoresponden.

3.3 Flowchart Enkripsi dan Dekripsi

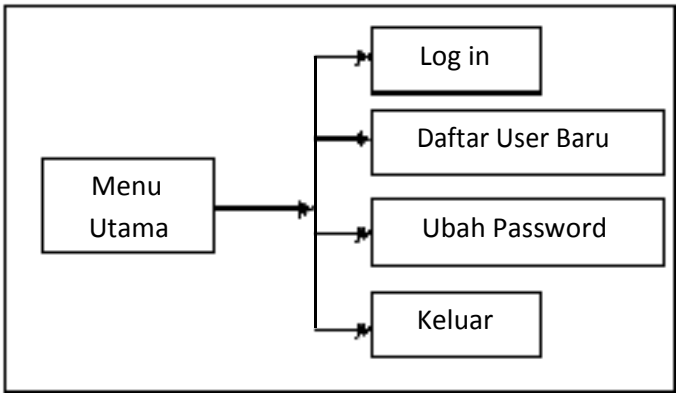
Adapun *flowchart* dari *enkripsi* dan *dekripsi* dengan menggunakan algoritma *Electronic Code Book (ECB)* dapat dilihat pada gambar sebagai berikut ini.

Gambar 3.2. *Flowchart Enkripsi*

Gambar 3.3. *Flowchart Dekripsi*

3.3.1 Struktur Menu

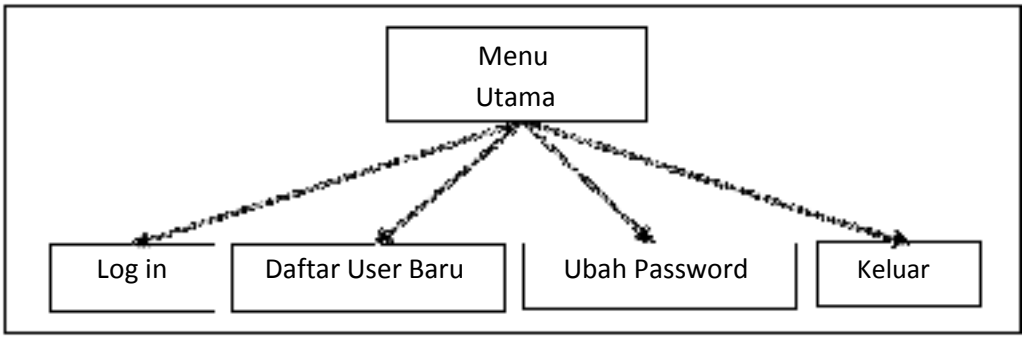
Struktur menu ini digunakan untuk menampilkan detail informasi menu yang digunakan dalam pembuatan program.



Gambar 3.4 Struktur Menu

3.3.2 Struktur Chart

Struktur *chart* ini digunakan untuk menampilkan detail informasi menu yang digunakan dalam pembuatan program.



Gambar 3.5 Sruktur Chart

3.3.3 Diagram Use Case

Model *Use Case* adalah sebuah kumpulan dari diagram dan teks yang mendeskripsikan bagaimana keinginan pengguna berinteraksi dengan sistem. Diagram *Use Case* mengidentifikasikan fungsionalitas yang disediakan oleh

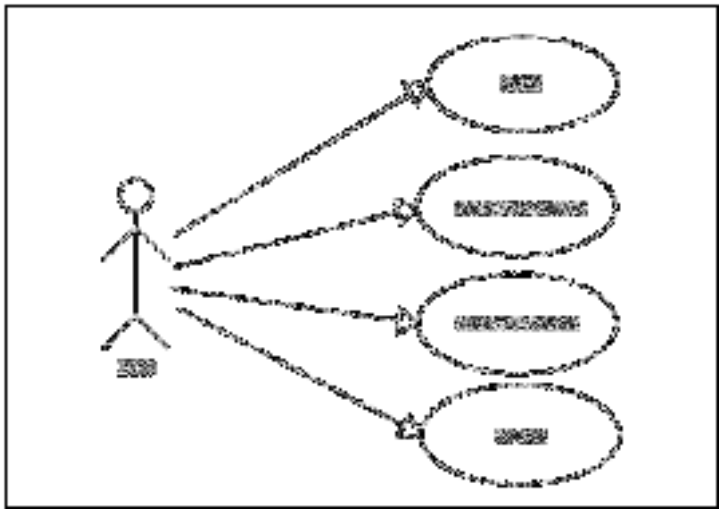
sistem (*Use Case*), dan pengguna yang berinteraksi dengan sistem (aktor), dan gabungan antara pengguna dan fungsionalitas.

Dalam perancangan *use case* diagram program, terdapat diagram *use case* program utama, diagram *use case log in*, diagram *use case* daftar *user* baru, dan diagram *use case* ubah *password*.

3.3.3.1 Use Case Menu Utama

Use Case program utama memiliki langkah–langkah sebagai berikut:

1. Pengguna dapat memilih mode *log in* dengan memilih *log in.*
2. Pengguna dapat memilih mode daftar *user* baru dengan memilih daftar *user* baru.
3. Pengguna dapat memilih mode ubah *password* dengan memilih daftar ubah *password.*
4. Pengguna dapat memilih keluar program dengan memilih *Keluar.*

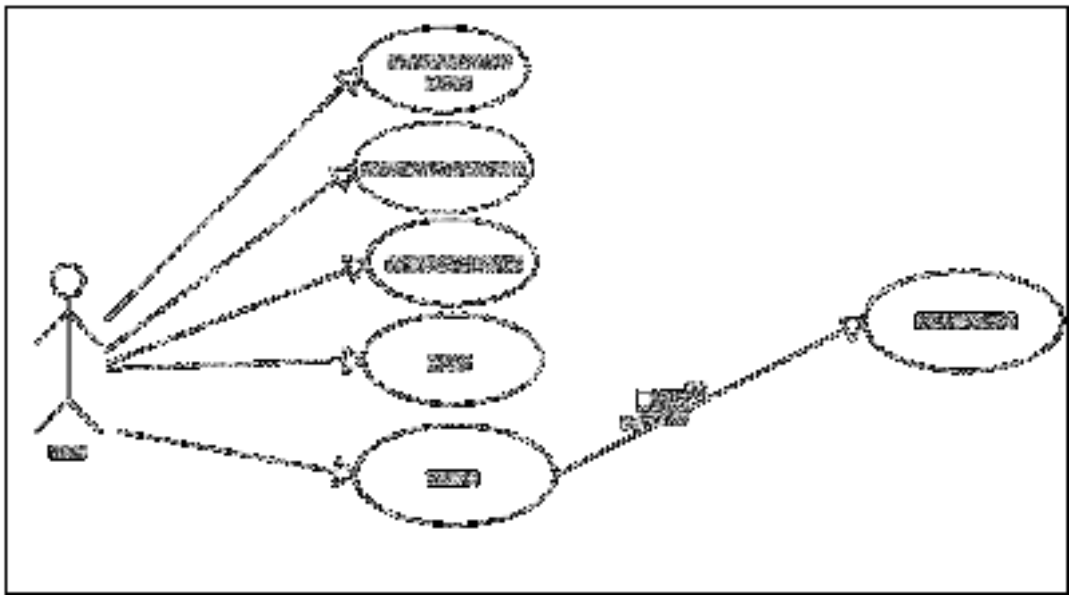


Gambar 3.6. *Use Case* Menu Utama

3.3.3.2 Use Case Log In

Use Case log in mempunyai langkah–langkah:

1. Pengguna diharuskan untuk mengisi nama *user* (*user name*) terlebih dahulu dengan mengisi *text box*.
2. Pengguna diharuskan untuk memasukkan *password* terlebih dahulu dengan mengisi *text box*.
3. Pengguna diharuskan untuk memasukkan angka pelindung terlebih dahulu dengan mengisi *text box*.
4. Pengguna dapat langsung *log in* dengan memilih *log in.*
5. Pengguna dapat keluar dari program dengan memilih keluar.

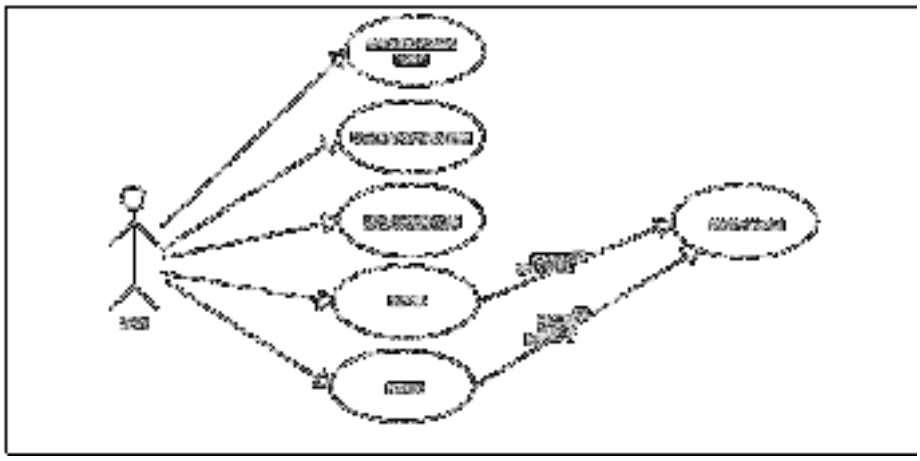


Gambar 3.7. *Use Case Log in*

3.3.3.3 Use Case Daftar User Baru

Use Case daftar *user* baru mempunyai langkah–langkah:

1. Pengguna diharuskan untuk memasukkan nama *user* (*user name*) terlebih dahulu dengan mengisi *text box*.
2. Pengguna diharuskan untuk memasukkan *password* terlebih dahulu dengan mengisi *text box*.
3. Pengguna diharuskan untuk memasukkan angka pelindung terlebih dahulu dengan mengisi *text box*.
4. Pengguna dapat langsung daftar dengan memilih daftar.
5. Pengguna dapat keluar dari program dengan memilih keluar.



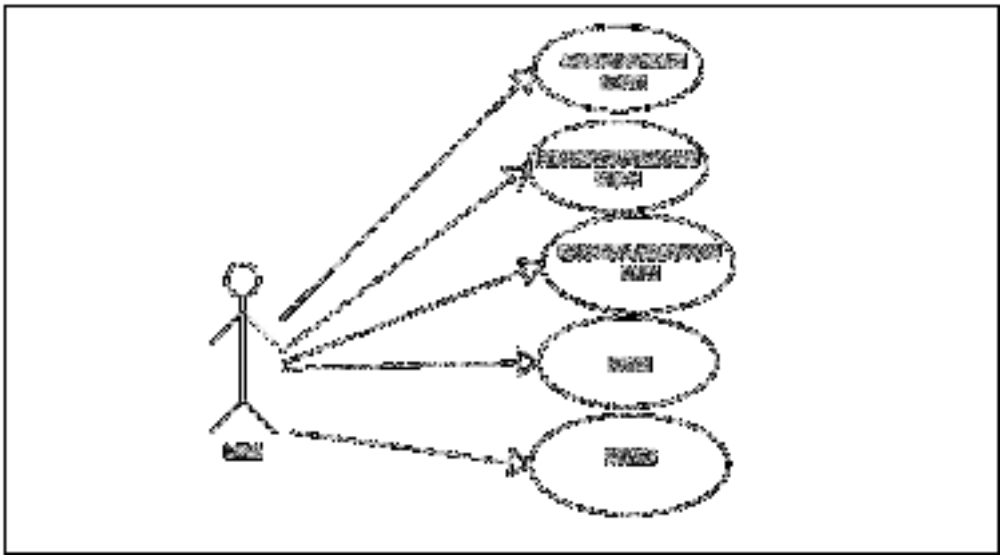
Gambar 3.8. *Use Case* Daftar *User* Baru

3.3.3.4 Use Case Ubah Password

Use Case ubah *password* baru mempunyai langkah–langkah:

1. Pengguna diharuskan untuk memasukkan nama *user* (*user name*) terlebih dahulu dengan mengisi *text box*.

2. Pengguna diharuskan untuk memasukkan *password* lama anda terlebih dahulu dengan mengisi *text box*.
3. Pengguna diharuskan untuk memasukkan *password* baru anda yang ingin diubah dengan mengisi *text box*.
4. Pengguna diharuskan untuk memasukkan angka pelindung terlebih dahulu dengan mengisi *text box*.
5. Pengguna dapat langsung daftar dengan memilih ubah.
6. Pengguna dapat keluar dari program dengan memilih keluar.



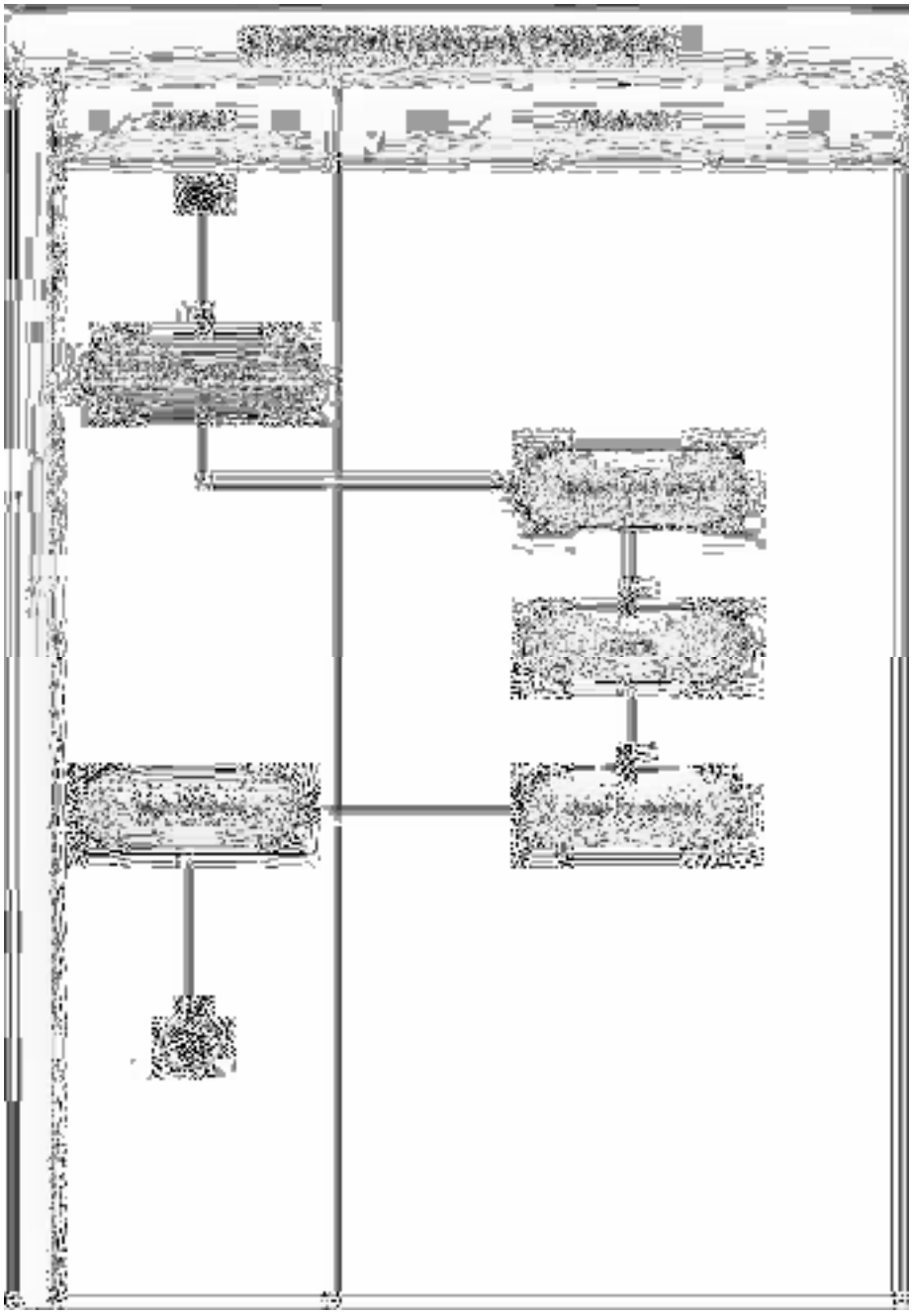
Gambar 3.9. *Use Case* Ubah *Password*

3.3.3.5 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state*

adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing).*

Dari *activity diagram* berikut ini terlihat bagian dari proses *enkripsi* keamanan data.



Gambar 3.10. *Activity Diagram*

Keterangan :

1.	Menu Utama	:	Tampilan menu utama dari program.
2.	Daftar *User* Baru	:	Tampilan *form* daftar *user* baru.
3.	*Log In*	:	Tampilan *form* untuk melakukan *log in*.
4.	Ubah *Password*	:	Tampilan form untuk mengubah password.
5.	Menu Keluar	:	Keluar dari program.

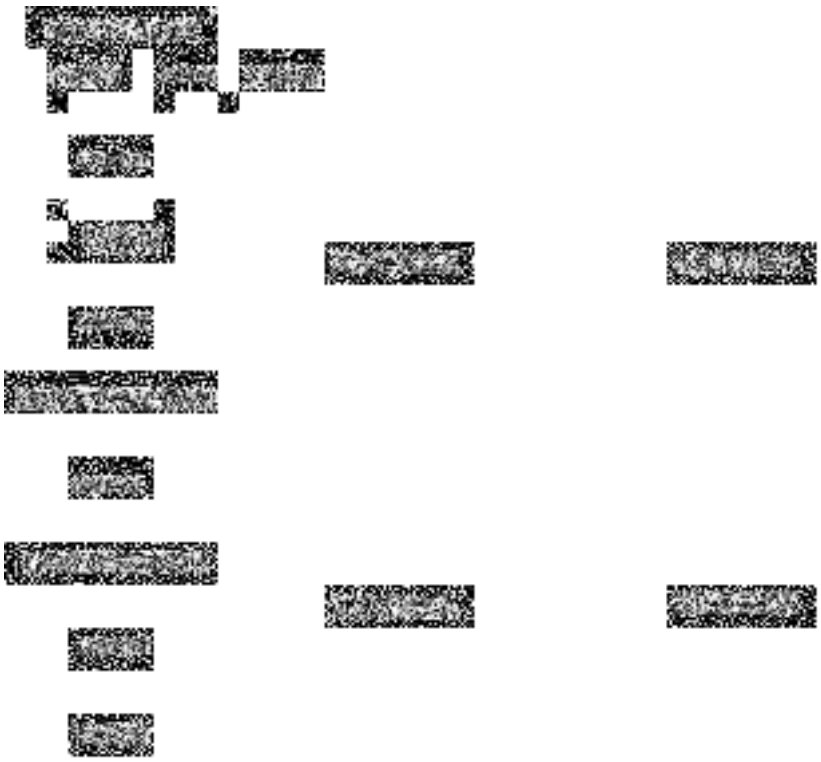
3.4 Pemodelan/Perancangan Sistem

Aplikasi akan dirancang dalam bentuk aplikasi yang bisa diimplementasikan. *Interface* disediakan untuk memudahkan pengguna dalam memberikan *input* berupa beberapa parameter yang diperlukan, serta menampilkan hasilnya, termasuk di dalamnya proses *enkripsi* maupun *dekripsi*, yang dalam hal ini menggunakan algoritma *Electronic Code Book (ECB).*

Untuk *User Interface* akan dirancang kedalam empat *form* tampilan, yang terdiri dari:

3.4.1 Form Menu Utama

Form menu utama ini merupakan tampilan dimana *user* akan memilih ke arah mana akan memulai menggunakan program ini. Berikut adalah rancangan *form* menu utama:



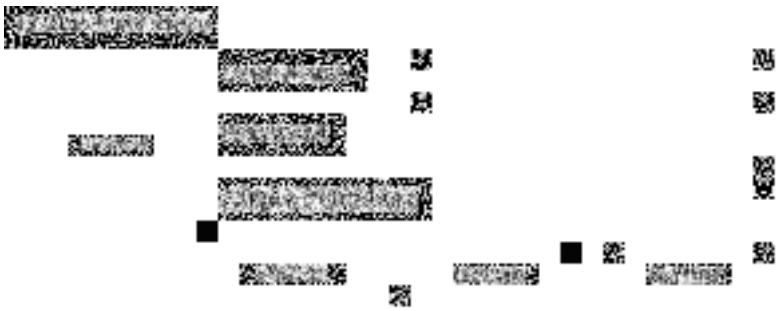
Gambar 3.11. Rancangan *Form Menu* Utama

Keterangan Gambar 3.11 Rancangan *Form Menu* Utama adalah sebagai berikut:

Form Menu Utama terdiri dari 4 buah *button* yang masing-masing memiliki fungsi yang berbeda sesuai kebutuhan program. Jika *button login* di klik, maka akan muncul menu *login*. Jika *button* daftar *user* baru di klik, maka akan muncul menu daftar *user* baru. Jika *button* keluar di klik, maka akan keluar dari program.

3.4.2 *Form* Daftar *User* Baru

Form Daftar *User* Baru ini merupakan tampilan dimana *user* bisa mendaftar atau membuat akun agar bisa masuk ke menu selanjutnya. Berikut adalah rancangan *form* daftar *user* baru:



Gambar 3.12. Rancangan *Form* Daftar *User* Baru

Keterangan Gambar 3.12 Rancangan *Form* Daftar *User* Baru adalah sebagai berikut:

Form Daftar *User* Baru terdiri dari 3 buah *text box* dan 3 buah *button* yang masing-masing memiliki fungsi yang berbeda sesuai kebutuhan program. *Text Box user name* berguna sebagai tempat pengetikan nama pendaftar. *Text Box password* berguna sebagai tempat pengetikan *password* pendaftar. *Text Box* angka pelindung berguna sebagai tempat pengetikan angka pelindung pendaftar. Jika *button* daftar di klik, maka akan otomatis terdaftar dan disimpan ke *databased.* Jika *button* bersih di klik, maka data yang ada di *text box* akan kembali bersih atau kosong. Jika *button* keluar di klik, maka akan keluar dari *form* daftar *user* baru.

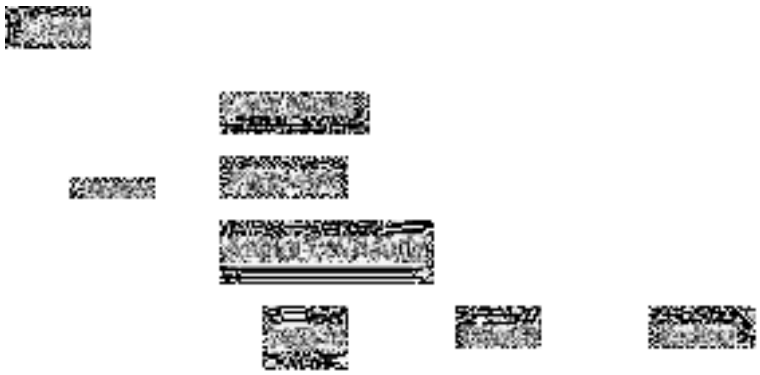
Setelah kita mendaftar maka Nama dan *Password* akan tersimpan pada tabel *login* yang secara otomatis akan tersimpan ke *database*, dan kemudian *password* secara otomatis juga akan terenkripsi ke dalam kalimat yang tidak berarti, sehingga tidak ada yang tahu *password* yang sesungguhnya. Adapun tampilan tabelnya adalah sebagai berikut:

Tabel 3.3. Tabel *Login*

NAMA	PASSWORD
Abdi Nuge	XXXXXXXX
Unpab	XXXXXXXX
Admin Nuge	XXXXXXXX

3.4.3 Form Login

Form login ini merupakan tampilan dimana *user* akan mengisi *input* data berupa nama *user* dan *password* sebagai pintu masuk menuju program/aplikasi. Berikut adalah rancangan *form login*:



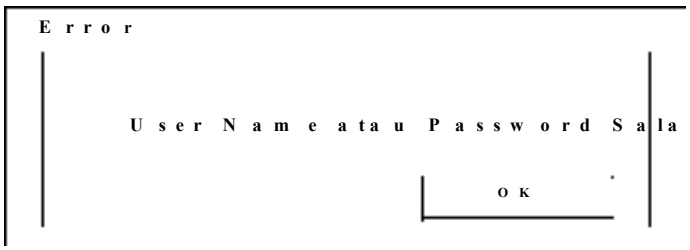
Gambar 3.13. Rancangan *Form Login*

Keterangan Gambar 3.13. Rancangan *Form Login* adalah sebagai berikut:

Form Login terdiri dari 3 buah *text box* dan 3 buah *button* yang masing-masing memiliki fungsi yang berbeda sesuai kebutuhan program. *Text Box User Name* berguna sebagai tempat pengetikan *user name* pengguna. *Text Box Password* berguna sebagai tempat pengetikan *password* pengguna. *Text Box* angka pelindung berguna sebagai tempat pengetikan angka pelindung pengguna.

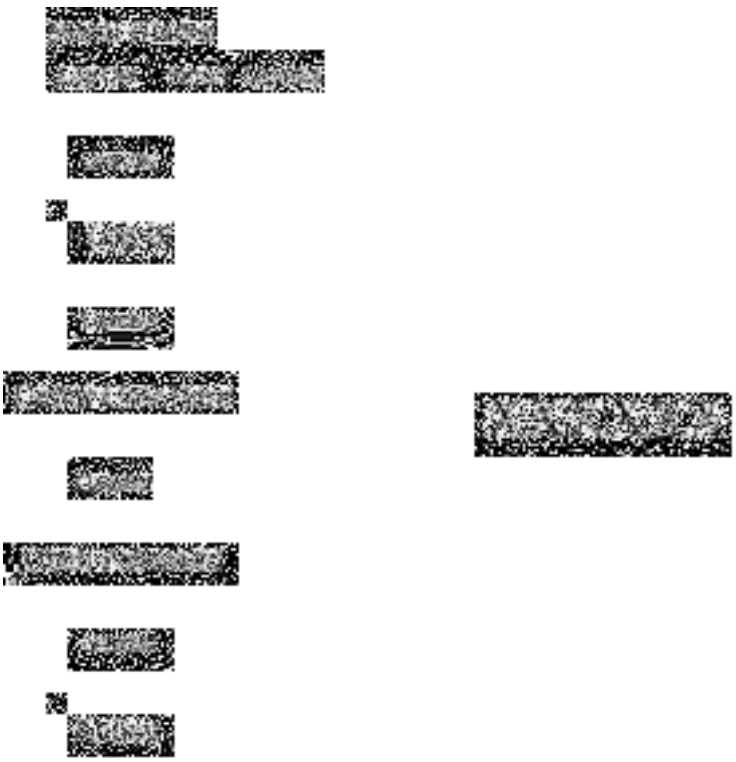
Jika *button login* di klik, maka akan masuk ke menu selanjutnya. Jika *button* bersih di klik, maka data yang ada di *text box* akan kembali bersih atau kosong. Jika *button* keluar di klik, maka akan keluar dari *form login*.

Dan apabila kita salah memasukkan *password* atau *user name* maka akan ada timbul pesan seperti berikut ini:



Gambar 3.14. Rancangan *Form* Salah Password atau User Name

Dan apabila kita telah login, maka akan muncul tampilan tampilan seperti berikut ini:



Gambar 3.15. Rancangan *Form* Tampilan Setelah *Login*

3.4.4 Form Ubah Password

Form ubah password ini merupakan tampilan dimana *user* bisa mengubah *password* lamanya menjadi *password* baru dengan catatan si pemilik *account* harus *login* terlebih dahulu untuk dapat mengubah passwodrnya. Berikut adalah rancangan *form* ubah *password*:

Gambar 3.16. Rancangan *Form* Ubah *Password*

Form ubah *password* terdiri dari 4 buah *text box* dan 2 buah *button* yang masing-masing memiliki fungsi yang berbeda sesuai kebutuhan program. *Text Box user name* berguna sebagai tempat pengetikan nama pengguna. *Text Box password* lama berguna sebagai tempat pengetikan *password* lama sebelum diubah. *Text Box password* baru berguna sebagai tempat pengetikan *password* baru setelah diubah. Jika *button* ubah di klik, maka akan otomatis *password* lama akan berubah menjadi *password* baru dan disimpan ke *databased.* Jika *button* bersih di klik, maka data yang ada di *text box* akan kembali bersih atau kosong. Jika *button* keluar di klik, maka akan keluar dari *form* ubah *password*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Sistem

Dalam penerapan kriptografi *Electronic Code Book* pada pengamanan data *login* ini tidak lepas dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Adapun *hardware* dan *software* yang dibutuhkan sebagai berikut:

1. Perangkat Keras (*Hardware*)
    a. *Processor* intel pentium IV 2.6 GHz atau lebih tinggi
    b. *Memory Ram* 2 GB
    c. *Harddisk* 80 GB
    d. O/S Windows 7
    e. Monitor
    f. *Keyboard* dan *mouse*
2. Perangkat Lunak (*Software*)
    a. *Microsoft Visual Basic* 2008
    b. *Microsoft Office Access* 2007

4.2 Implementasi Sistem

Implementasi adalah pelaksanakan sebuah aplikasi. Dalam implementasi pengamanan data *login* ini akan menampilkan implementasi rancangan antar

muka. Berikut ini merupakan implementasi rancangan antar muka dari sistem yang dibuat:

1. *Form* Utama

Form utama adalah *form* yang menampilkan menu-menu pilihan pada pengamanan data *login*. *Form* ini berisi menu *login*, daftar *user* baru , ubah *password* dan keluar. Adapun tampilan dari form utama adalah sebagai berikut:

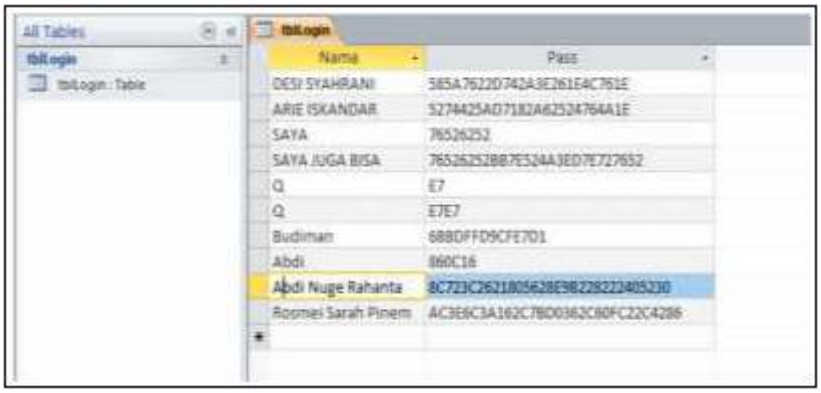
Gambar 4.1 *Form* Menu Utama

2. *Form* Daftar *User* Baru

Form daftar *user* baru adalah *form* yang digunakan untuk membuat sebuah *account* untuk dapat *login* atau masuk kedalam aplikasi. Pada *form* ini kita diharuskan menginput Nama, *Pasword* dan angka pelindung. Adapun tampilan dari *form* daftar *user* baru adalah sebagai berikut :

Gambar 4.2 *Form* Daftar *User* Baru

Setelah kita mendaftar maka Nama dan *Password* akan tersimpan pada tabel *login* yang secara otomatis akan tersimpan ke *database*, dan kemudian *password* secara otomatis juga akan terenkripsi ke dalam kalimat yang tidak berarti, sehingga tidak ada yang tahu *password* yang sesungguhnya. Adapun tampilan tabelnya adalah sebagai berikut:



Gambar 4.3 *Database* Tabel *Login*

3. *Form Login*

Form login adalah *form* yang ditampilkan untuk memastikan apakah kita telah sukses masuk ke dalam aplikasi. Pada *form* ini *user* diharuskan menginputkan *User Name*, *Password* dan angka pelindung yang merupakan kunci untuk masuk ke dalam aplikasi. Adapun tampilan dari *form login* adalah sebagai berikut:

Gambar 4.4 *Form Login*

Dan apabila kita salah memasukkan *password* atau *user name* maka akan ada tibul pesan seperti berikut ini:

Gambar 4.5 Salah *Password* atau *User Name*

Dan apabila kita telah login, maka akan muncul tampilan tampilan seperti berikut ini:

Gambar 4.6 Tampilan Setelah *Login*

4. *Form* Ubah *Password*

Form ubah *password* adalah *form* untuk mengubah *password* lama menjadi *password* baru yang kita inginkan, tetapi untuk merubah *password* tersebut kita harus *login* terlebih dahulu untuk memastikan apakah *account* kita terdaftar di *database* atau tidak. Pada *form* ini *user* diharuskan menginputkan *User Name*, *Password* lama, password baru dan angka pelindung untuk merubah. Adapun tampilan dari *form* ubah *password* adalah sebagai berikut:

Gambar 4.7 *Form* Ubah *Password*

4.3 Pengujian

Pengujian sistem bertujuan untuk membuktikan bahwa *input*, proses dan hasil *enkripsi* yang dihasilkan oleh sistem telah benar dan sesuai dengan yang diinginkan. Pengujian sistem dilakukan dengan cara memasukkan *user name* dan *password*. Berikut merupakan tahapan untuk pengujian sistem yaitu:

1. Mendaftarkan *user* baru pada *form* daftar *user* baru yang akan disimpan di *database.*
2. Melakukan *login* untuk memastikan kita telah masuk ke dalam aplikasi tersebut atau tidak dengan menggunakan *account* yang telah didaftarkan.

3. Apabila kita ingin mengubah *password* lama kita, kita dapat mengubahnya di *form* ubah *password* dengan catatan kita harus *login* terlebih dahulu.

4.4 Kelemahan dan Kelebihan Sistem

Adapun kelemahan dari sistem yang dirancang adalah sebagai berikut:

1. Sistem hanya dapat mengenkripsi *password* pengguna yang telah di daftarkan pada *form* daftar *user* baru
2. Tampilan aplikasi yang kurang sempurna.
3. Jumlah banyak *character* dalam *user name* dan *password* telah dibatasi yaitu 13 *character.*

Sedangkan kelebihan sistem yang dirancang adalah sebagai berikut:

1. Sistem ini dirancang untuk mengenkripsi *password* menjadi kalimat yang tidak dimengerti maksud dan artinya.
2. Sistem menyediakan *form login*.
3. Sistem dapat sewaktu-waktu menambah parameter-parameter dalam pendaftaran *user* baru.
4. *Database* hanya dapat dilihat oleh seorang *administrator* yang mengolah sistem ini, sehingga pengguna lain tidak bisa melihatnya.

BAB V
PENUTUP

5.1 Kesimpulan

Setelah melakukan penelitian dan pengujian, maka penulis mendapatkan suatu kesimpulan yang dapat digunakan sebagai garis besar dari keseluruhan rangkuman skripsi ini.

Adapun kesimpulan dari penelitian ini adalah sebagai berikut;

a. *Password* pada data *login* ini dienkripsi dengan cara, menjadikan blok-blok *plainteks* menjadi 8 bit dan menjadikan kunci juga 8 bit dan kemudian meng-XOR-kan blok *plainteks* P_i dengan K, kemudian geser secara *wrapping* bit-bit dari $P_i \oplus K$ satu posisi ke kiri dan hasil *wrapping* di XOR kan kembali dengan kunci awal.
b. Aplikasi pengamanan data *login* ini dibuat atau dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual Studio 2008*.
c. Aplikasi ini mengubah *password* asli pengguna ke dalam kalimat yang tidak dimengerti artinya, karena telah dienkripsi oleh sistem. Sehingga orang yang berniat membobol *password* si pengguna sangat kecil kemungkinannya karna *password* aslinya telah dirubah.
d. Aspek utama pada pengamanan data *login* ini terletak pada *user name* dan baris kuncinya.

5.2 Saran

Dalam penelitian ini, ada beberapa saran yang nantinya dapat digunakan sebagai pengembangan yang lebih baik untuk kedepannya.

Adapun beberapa saran dari skripsi ini adalah:

a. Agar kerahasiaan pada data *password* dapat terjaga dengan baik, maka sebaiknya hanya si pemilik/*user* saja yang tahu dan jangan sampai lupa, karena data yang tersimpan dalam *database* adalah data yang telah dienkripsi.
b. Untuk menjaga kerahasiaan pada *password login*, sebaiknya dipilih kata yang unik, mudah diingat dan bersifat pribadi.
c. Hindari *password* yang berasal dari kata yang bersifat umum karena akan memiliki kemungkinan yang besar untuk ditebak.

DAFTAR PUSTAKA

Rifki Sadikin. (2012). ***Kriptografi Untuk Keamanan Jaringan dan Implementasinya Dalam Bahasa Java.*** Yogyakarta: Andi.

Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." IT Journal Research and Development 2.1 (2017): 1-11.

Batubara, S., Wahyuni, S., & Hariyanto, E. (2018, September). Penerapan Metode Certainty Factor Pada Sistem Pakar Diagnosa Penyakit Dalam. In Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 81-86).

Mariance, U. C. (2018). Analisa dan Perancangan Media Promosi dan Pemasaran Berbasis Web Menggunakan Work System Framework (Studi Kasus di Toko Mandiri Prabot Kota Medan). Jurnal Ilmiah Core IT: Community Research Information Technology, 6(1).

Sarif, M. I. (2017). Penemuan Aturan yang Berkaitan dengan Pola dalam Deret Berkala (Time Series).

Putri, N. A. (2018). Sistem Pakar untuk Mengidentifikasi Kepribadian Siswa Menggunakan Metode Certainty Factor dalam Mendukung Pendekatan Guru. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 78-90.

Hendrawan, J. (2018). Rancang Bangun Aplikasi Mobile Learning Tuntunan Shalat. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 44-59.

Dony Ariyus. (2006). ***Computer Security***. Yogyakarta: Andi,.

Hera Napit (11 Januari 2010). Algoritma Kritptografi Modern, dari http://heranapit.blogspot.com/2010/11/algoritma-kriptografi-modern.html

Jogiyanto HM (1999 : 692). Pengertian Informasi Menurut Para Ahli Defenisi, dari http://www.sarjanaku.com/2012/11/pengertian-informasi-menurut-para-ahli.html

Fachri, B. (2018). Perancangan Sistem Informasi Iklan Produk Halal Mui Berbasis Mobile Web Menggunakan Multimedia Interaktif. Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika), 3, 98-102.

Fachri, B. (2018, September). APLIKASI PERBAIKAN CITRA EFEK NOISE SALT & PAPPER MENGGUNAKAN METODE CONTRAHARMONIC MEAN FILTER. In Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 87-92).

George H. Bodnar (2000 : 1). Pengertian Informasi Menurut Para Ahli Defenisi, dari http://www.sarjanaku.com/2012/11/pengertian-informasi-menurut-para-ahli.html

Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. Int. J. Recent Trends Eng. Res, 3(8), 58-64.

Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. Int. J. Secur. Its Appl, 10(8), 173-180.

Robert N. Antony dan John Dearden. Pengertian Data dan Informasi, dari http://nitannia.blogspot.com/2012/11/pengertian-data-dan-informasi-menurut_19.html

Jogyanto, Pengertian Data dan Informasi, dari http://nitannia.blogspot.com/2012/11/pengertian-data-dan-informasi-menurut_19.html

M. Nishom (16 Desember 2012). Pengertian Keamanan Komputer, dari http://www.isomwebs.com/2012/pengertian-keamanan-komputer/

Noerpedia (24 April 2011). Membongkar dan Menghancurkan Kriptografi, dari http://noerpedia.blogspot.com/2011/04/membongkar-dan-menghancurkan.html

Hadi Wibowo (28 Juli 2008). Serangan Terhadap Kriptografi, dari http://hadiwibowo.wordpress.com/2008/07/28/serangan-terhadap-kriptografi/

Andre Satria Al Jamby (7 Desember 2009). Tujuan Kriftografi, darihttp://cyb3rbl4ck.blogspot.com/2009/12/tujuan-kriptografi.html

Naila Fitria (13506036). Jenis-Jenis Serangan Terhadap Kriptografi

Kurnia, D., Dafitri, H., & Siahaan, A. P. U. (2017). RSA 32-bit Implementation Technique. Int. J. Recent Trends Eng. Res, 3(7), 279-284.

Ruwaida, D., & Kurnia, D. (2018). Rancang Bangun File Transfer Protocol (FTP) dengan Pengamanan Open SSL pada Jaringan VPN Mikrotik di SMK Dwiwarna. CESS (Journal of Computer Engineering, System and Science), 3(1), 45-49.

Khairul, K., Haryati, S., & Yusman, Y. (2018). Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. Jurnal Teknologi Informasi dan Pendidikan, 11(1), 1-6.

Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol

application for LAN and Wi-Fi communication. Int. J. Eng. Technol., 7(2.13), 345-347.

Hariyanto, E., & Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. International Journal of Science and Research (IJSR), 5(10), 1363-1365.