APLIKASI ALGORITMA DES
DAN *CRYPTANALYSIS* MENGGUNAKAN TEORI PROBABILITAS
PADA KARTU ATM

Oleh

SUDARMONO

M 0198080

SKRIPSI

Ditulis dan diajukan untuk memenuhi sebagian persyaratan

Memperoleh gelar Sarjana Sains Matematika

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SEBELAS MARET

SURAKARTA

2006

SKRIPSI

APLIKASI ALGORITMA DES

DAN *CRYPTANALYSIS* MENGGUNAKAN TEORI PROBABILITAS

PADA KARTU ATM

yang disiapkan dan disusun oleh

SUDARMONO

M 0198080

dibimbing oleh

Pembimbing I,	Pembimbing II,
DR. Sutanto, SSi, DEA	Sri Kuntari, M.Si
NIP. 132 149 079	NIP. 132 240 173

telah dipertahankan di depan Dewan Penguji

pada hari Senin, tanggal 9 Oktober 2006

dan dinyatakan telah memenuhi syarat.

Anggota Tim Penguji:	Tanda Tangan
1. Drs. Bambang Harjito, M.App.Sc NIP. 131 947 675	1. .....
2. Drs. Sugiyanto, M.Si NIP. 132 000 804	2. .....
3. Irwan Susanto, S.Si, DEA NIP. 132 134 694	3. .....

Surakarta, 9 Oktober 2006

Disahkan oleh

Fakultas Matematika dan Ilmu Pengetahuan Alam

Dekan,	Ketua Jurusan Matematika,
Drs. H. Marsusi, M.S.	Drs. Kartiko, M.Si
NIP. 130 906 776	NIP. 131 569 203

ABSTRAK

Sudarmono. 2006. **APLIKASI ALGORITMA DES DAN *CRYPTANALYSIS* MENGGUNAKAN TEORI PROBABILITAS PADA KARTU ATM**, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret.

Kriptografi adalah seni atau ilmu pengetahuan yang digunakan untuk menjaga keamanan dan kerahasiaan data atau informasi agar tidak diketahui oleh orang yang tidak berhak. Kebalikan dari kriptografi adalah *cryptanalysis*, yaitu seni atau ilmu untuk memecahkan rahasia suatu penyandian tanpa melalui cara yang seharusnya. Algoritma DES merupakan salah satu algoritma kriptografi simetri yang digunakan untuk melindungi data dalam bidang perbankan, yaitu untuk membuat penyandian dalam kartu ATM.

Tujuan dari penulisan skripsi ini adalah dapat mengetahui proses penentuan PIN sebuah kartu ATM menggunakan algoritma DES dan dapat memecahkan rahasia penyandian algoritma DES pada kartu ATM menggunakan teori probabilitas. ATM yang dibahas dalam skripsi ini adalah ATM Eurocheque. Metode penelitian yang digunakan dalam penulisan skripsi ini adalah studi literatur dan simulasi. Adapun langkah-langkah yang dilakukan adalah dengan terlebih dahulu menjabarkan proses enkripsi dan dekripsi data menggunakan algoritma DES kemudian mengaplikasikan algoritma DES dalam sistem keamanan ATM yaitu untuk menentukan empat digit PIN suatu ATM. Selanjutnya menggunakan teori-teori dalam probabilitas dilakukan *cryptanalysis* terhadap sistem keamanan ATM.

Dari hasil pembahasan dalam skripsi ini dapat disimpulkan algoritma DES yang digunakan dalam sistem keamanan ATM masih membuka peluang bagi *cryptanalys* untuk melakukan *cryptanalysis* terhadap sistem tersebut. Dengan mengetahui data pada *magnetic stripe* sebuah ATM, seorang *cryptanalys* dapat menebak empat digit PIN yang dipakai oleh nasabah pengguna ATM menggunakan teori probabilitas.

ABSTRACT

Sudarmono. 2006. **APPLICATION OF DES ALGORITHM AND CRYPTANALYSIS BY USING PROBABILITY THEORY AT ATM CARD, Faculty of Mathematics and Natural Sciences, Sebelas Maret University.**

Cryptography is an art or science used to take care of security and data secret or information in order not to know by one who have no business. Reverse of cryptography is cryptanalysis, that is art or science to solve encoding secret without passing the way of which ought to. DES algorithm represent one of the symmetry cryptography algorithm used to protect data in the field of banking, that is to make encoding in card of an ATM.

The Objective of this thesis writing is to know the determination process of PIN in ATM card which is using DES algorithm and can solve encoding secret of DES algorithm at ATM card using probability theory. ATM which is discussed in this thesis is ATM Eurocheque. Research method which is used in writing of this thesis is literature study and simulation. As for stages, steps taken is beforehand formulate process of encryption and decryption data use DES algorithm later, then application of DES algorithm in security system of ATM that is to determine four PIN digit in ATM. The next step is using theorys in probability conducted cryptanalysis in security system of ATM.

From the analysis in this thesis, it can be concluded that DES algorithm which is used in the security system of ATM is still open opportunity to cryptanalys to conduct cryptanalysis on that system. By knowing the magnetic stripe data in an ATM, a cryptanalys can guess four PIN digit which used by the consumer of an ATM by using probability theory.

MOTO

Dan katakanlah, "Bekerjalah kamu, maka Allah dan Rasul-Nya serta orang-orang mukmin akan melihat pekerjaanmu itu, dan kamu akan dikembalikan kepada (Allah) Yang Maha Mengetahui yang ghaib dan yang nyata, lalu diberitakan-Nya kepada kamu apa yang telah kamu kerjakan."

(Q.S. At Taubah : 105)

Orang yang cerdas adalah yang senantiasa mengoreksi dirinya dan mempersiapkan amal untuk kehidupan setelah kematian. Adapaun orang yang bodoh adalah yang memperturutkan hawa nafsunya dan berpanjang angan-angan kepada Alloh.

(HR. Tirmidzi)

Orang lain tak akan pernah mengerti berapa lama sesuatu itu dikerjakan. Yang mereka ketahui hanyalah seberapa sempurnanya sesuatu itu dikerjakan.

(Nancy Hanks)

Seseorang yang enggan mengerjakan suatu hal maka orang tersebut akan selalu memiliki sedikit waktu untuk mengerjakannnya.

(Penulis)

PERSEMBAHAN

Karya sederhana ini kupersembahkan untuk

- *Alloh dan Rosul-Nya yang mulia serta para Mujahid Dakwah di jalan-Nya*
- *Ayah dan Ibunda tercinta yang telah mengajari bagaimana memberi tanpa mengharap balasan…*
- *Adik-adikku tersayang : Haryanto, Purwanto dan Ratih*
- *Semua ustadz dan guruku yang mulia serta semua sahabatku*
- *Almamater, Universitas Sebelas Maret Surakarta*

KATA PENGANTAR

Assalamu ‘alaikum Wr. Wb.

Segala persembahan hanya bagi Alloh, pemilik segala ilmu dan penggenggam segala jiwa makhluk-Nya. Sholawat dan salam semoga tercurah bagi guru segala manusia, nabi akhir zaman, Rosulullo Muhammad SAW.

Alhamdulillah, berkat rahmat Alloh penulis dapat menyelesaikan skripsi berjudul **“Aplikasi Algoritma DES dan *Cryptanalysis* Menggunakan Teori Probabilitas pada Kartu ATM”.** Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Sains Matematika pada Fakultas Matematika dan Ilmu Pengetahuan Alam Jurusan Matematika Universitas Sebelas Maret Surakarta.

Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam penulisan skripsi ini, baik secara langsung maupun tak langsung, terutama kepada :

1. Bapak DR. Sutanto, S.Si, DEA, Pembimbing I yang dengan sabar memberikan bimbingan dan arahan kepada penulis dalam penyusunan skripsi;
2. Ibu Sri Kuntari, M.Si, Pembimbing II yang dengan sabar telah memberikan bimbingan dan dorongan kepada penulis dalam penyusunan skripsi ini;
3. Bapak Drs. Sutrima, M.Si, Pembimbing Akademik yang telah memberikan bimbingan dan kepercayaan kepada penulis selama kuliah;
4. Bapak Drs. Kartiko, M.Si, Kajur Matematika Fakultas MIPA Universitas Sebelas Maret Surakarta yang telah memberikan izin kepada penulis untuk menyelesaikan skripsi ini;
5. Bapak Drs. Marsusi, M.S, Dekan Fakultas MIPA Universitas Sebelas Maret Surakarta yang telah memberi kesempatan kepada penulis untuk menyusun skripsi ini;
6. Seluruh staf pengajar Jurusan Matematika Fakultas MIPA Universitas Sebelas Maret Surakarta, yang telah banyak menularkan ilmunya pada penulis;

7. Kedua orang tuaku tercinta, kakek, nenek dan adik-adikku tersayang yang telah mencurahkan semua perhatian dan kasih sayangnya kepada penulis. Tidak ada yang bisa membalas kasih sayang dan pengorbanan mereka kecuali Alloh SWT. Ya Alloh....sayangi dan rahmatilah mereka;
8. Bapak H. Sabar Mulyanto, STP sekeluarga yang telah memberikan dukungan baik moril maupun materiil kepada penulis, semoga Alloh membalas kebaikanya;
9. *Kelompok belajar* yang senantiasa mengevaluasi perkembangan skripsi penulis setiap pekan. *Jazakumulloh* atas semua perhatian dan pembinaanya. *Antum* semua telah mengajari penulis indahnya hidup dalam naungan cinta;
10. Semua ustadz dan guruku yang mulia, engkaulah sarana penyampai ilmu Alloh SWT kepadaku. Semoga Alloh membalas semua kebaikan dan keikhlasanmu dengan balasan yang lebih baik;
11. Keluarga besar Komunitas Tarbiyah Surakarta yang selalu memberikan kesempatan kepada penulis untuk belajar menjadi yang terbaik.
12. Teman–teman seperjuangan dalam dakwah kampus di Syiar Kegiataan Islam (SKI) FMIPA UNS, HIMATIKA FMIPA UNS, KAMMI Daerah Solo, FKM Ma'had Al Bina Surakarta serta teman-teman dalam aktivitas dakwah sekolah di Surakarta. Kalian telah mengajarkan makna ***ukhuwah*** dan ***amal jama'i*** kepada penulis. Tidak kuasa penulis menyebut nama ***antum*** satu persatu;
13. Takmir, Reisma dan Jamaah Masjid Ar Rohman di Palur yang selalu menemani hari-hari penulis selama menyelesaikan skrispi ini. Kalianlah penghibur dan pelipur lara dalam proses pematangan menuju dewasa;
14. Teman-teman Jurusan Matematika angkatan 98 yang telah menemani penulis selama kuliah, semoga persahabatan ini senantiasa terjalin selamanya;
15. Semua pihak yang belum penulis sebutkan satu persatu dalam tulisan ini.

Akhirnya semoga skripsi ini dapat bermanfaat bagi pembaca dan dapat memberikan sumbangan kebaikan pada perkembangan ilmu pengetahuan. Amin.
Wassalamu 'alaikum Wr. Wb.

Surakarta, Oktober 2006

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

Halaman

BAB I
PENDAHULUAN

1.1 Latar belakang Masalah

Informasi saat ini sudah menjadi sebuah komoditi yang sangat penting. Bahkan ada yang mengatakan bahwa saat ini sudah berada di sebuah *"information-based society"* atau masyarakat berbasis informasi. Kemampuan untuk mengakses dan menyediakan informasi secara cepat dan akurat menjadi sangat esensial bagi sebuah organisasi, baik yang berupa organisasi komersial, perguruan tinggi, lembaga pemerintahan, maupun individual. Sangat pentingnya nilai sebuah informasi menyebabkan seringkali informasi hanya boleh diakses oleh orang-orang tertentu. Jatuhnya informasi ke pihak lain dapat menimbulkan kerugian bagi pemilik informasi, sehingga masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi.

Menurut Rahardjo [8], keamanan informasi *(information security)* adalah bagaimana mencegah penipuan *(cheating)*, pemalsuan, atau, paling tidak, dapat mendeteksi adanya kejahatan di sebuah sistem yang berbasis informasi. Untuk itu diperlukan sebuah sistem yang menjamin keamanan dan kerahasiaan data atau informasi agar tidak diketahui oleh orang yang tidak berhak.

Salah satu cara untuk menjaga supaya informasi aman (tidak diketahui orang lain) adalah dengan teori penyandian yang biasa disebut dengan teori kriptografi *(cryptography).* Kriptografi yaitu ilmu yang mempelajari kemungkinan untuk melakukan suatu komunikasi antara dua orang melalui saluran yang tidak terjamin keamanannya [12], sehingga pihak yang tidak berkepentingan tidak dapat memahami apa yang sedang dikomunikasikan. Sedangkan menurut Menezes ***et al.*** [5], kriptografi adalah ilmu yang menggunakan matematika sebagai teknik untuk menyelesaikan masalah yang berhubungan dengan aspek keamanan informasi.

Kriptografi pada dasarnya sudah dikenal sejak lama. Menurut Wibowo [15], kriptografi sudah digunakan oleh bangsa Mesir sejak 4000 tahun yang lalu oleh raja-raja Mesir pada saat perang untuk mengirimkan pesan rahasia kepada

panglima perangnya melalui kurir-kurirnya. Seiring dengan perkembangan teknologi, algoritma kriptografi pun mulai berubah menuju ke arah algoritma kriptografi yang lebih rumit dan kompleks. Kriptografi diakui atau tidak mempunyai peranan yang paling penting dalam peperangan sehingga algoritma kriptografi berkembang cukup pesat pada saat Perang Dunia I dan Perang Dunia II. Beberapa algoritma kriptografi yang pernah digunakan dalam peperangan, diantaranya adalah ADFVGX yang dipakai oleh Jerman pada Perang Dunia I, *Sigaba/M-134* yang digunakan oleh Amerika Serikat pada Perang Dunia II, *Typex* oleh Inggris, dan *Purple* oleh Jepang. Selain itu Jerman juga mempunyai mesin legendaris yang dipakai untuk memecahkan sandi yang dikirim oleh pihak musuh dalam peperangan yaitu, *Enigma*.

Sebelum tahun 1970, teknologi kriptografi digunakan terbatas hanya untuk tujuan militer dan diplomatik. Kemudian bidang bisnis dan perorangan mulai menyadari pentingnya melindungi informasi yang berharga sehingga menggunakan teknologi kriptografi ini. Kriptografi merupakan salah satu metode pengamanan data yang digunakan untuk menjaga kerahasiaan data, keaslian data, serta keaslian pengirim. Metode ini bertujuan agar informasi yang bersifat rahasia yang dikirim melalui telekomunikasi umum seperti LAN atau internet, tidak dapat diketahui atau dimanfaatkan oleh orang yang tidak berkepentingan [10].

Kriptografi mempunyai banyak algoritma, masing-masing algoritma mempunyai tujuan penggunaan dan tingkat keamanan yang berbeda. Algoritma kriptografi modern tidak lagi mengandalkan keamanannya pada kerahasiaan algoritma tetapi pada kerahasiaan kunci. Sebuah algoritma kriptografi tidak menjadi masalah dipublikasikan dan dianalisis oleh semua orang, tetapi jika seseorang tidak mengetahui kunci rahasianya maka ia tidak dapat membuka pesan atau informasi yang dikirim tersebut. Contoh sistem semacam ini adalah kartu debit dan kartu kredit. Semua kartu debit dan kartu kredit yang beredar diseluruh dunia menggunakan algoritma kriptografi yang sama yaitu *Data Encryption Standard* (DES) dan *Rivest-Shamir-Adleman* (RSA) [3]. Semua orang bisa mengetahui secara rinci algoritma DES dan RSA, tetapi tanpa mengetahui kunci

dari algoritma tersebut mereka tidak dapat melakukan dekripsi. Dengan memberikan satu kunci kepada setiap kartu, keamanan kartu kredit atau kartu debet dapat diandalkan.

Algoritma yang akan digunakan dalam skripsi ini adalah algoritma DES yang merupakan algoritma kriptografi simetri yang paling umum digunakan di dunia. DES merupakan nama dari sebuah algoritma untuk mengenskripsi data yang dikeluarkan oleh *Federal Information Processing Standard* 46 (FIPS PUB 46) Amerika Serikat. Algoritma dasarnya dikembangkan oleh *International Business Machine* (IBM), *National Security Agency* (NSA), dan *National Bureau of Standard* (NBS) yang berperan penting dalam pengembangan bagian akhir algoritmanya. DES secara resmi digunakan sebagai algoritma standar untuk enkripsi pada tahun 1977 oleh *Nasional Institute of Standard and Technology* (NIST) [11].

Algoritma DES sangat banyak digunakan untuk melindungi data dalam dunia elektronik khususnya di bidang perbankan, *financial* dan *e-commerce.* Salah satu contoh aplikasi algoritma DES di bidang perbankan adalah untuk membuat penyandian pada kartu Anjungan Tunai Mandiri atau *Automated Teller Machines* (ATM) yaitu untuk menentukan *Personal Identification Number* (PIN) dalam kartu ATM. Pada penulisan skripsi ini akan jelaskan aplikasi algoritma DES yang digunakan untuk menentukan empat digit PIN dalam kartu ATM Eurocheque (EC). ATM Eurocheque merupakan salah satu ATM yang digunakan di negara-negara Eropa. Selanjutnya akan ditunjukkan bagaimana cara memecahkan rahasia penyandian algoritma DES pada kartu ATM Eurocheque menggunakan teori probabilitas.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, rumusan masalah yang di kemukakan dalam penulisan skripsi ini adalah :

1. Bagaimana proses penentuan PIN sebuah kartu ATM menggunakan algoritma DES ?

2. Bagaimana memecahkan rahasia penyandian algoritma DES pada kartu ATM menggunakan teori probabilitas ?

1.3 Batasan Masalah

Untuk membatasi agar penulisan skripsi ini tidak terlalu meluas, diberikan beberapa batasan masalah sebagai berikut :

1. ATM yang dibahas dalam skripsi ini adalah ATM Eurocheque
2. PIN hanya terdiri dari empat digit desimal,
3. Data pada *magnetic stripe* dan ATM diketahui.

1.4 Tujuan Penulisan

Tujuan dari penulisan skripsi ini adalah :

1. Dapat menentukan PIN sebuah kartu ATM menggunakan algoritma DES,
2. Dapat memecahkan rahasia penyandian algoritma DES pada kartu ATM menggunakan teori probabilitas.

1.5 Manfaat Penulisan

1. Secara teoritis manfaat yang dapat diperoleh dari penulisan skripsi ini adalah untuk mengembangkan ilmu pengetahuan khususnya bidang matematika dan kriptografi,
2. Manfaat praktis dari penulisan ini yaitu dapat memahami aplikasi algoritma DES yang digunakan dalam proses penyandian suatu kartu ATM dan memahami penguraian penyandian pada algoritma DES. Selanjutnya aplikasinya dapat dikembangkan dalam berbagai bidang keuangan maupun *e-comercee.*

BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

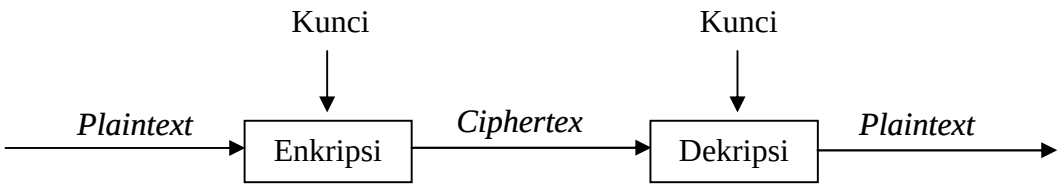
Untuk memberikan landasan penulisan skripsi ini, perlu diberikan beberapa istilah, definisi, teorema, dan pengertian yang relevan dengan pembahasan. Beberapa landasan teori yang akan diberikan pada subbab ini meliputi kriptografi, DES dan konsep dasar dalam teori probabilitas

2.1.1 Kriptografi

1. Terminologi

Menurut Kurniawan [3] dan Schneier [9], kriptografi adalah suatu seni atau ilmu pengetahuan untuk menjaga keamanan pesan atau informasi. Kriptografi berasal dari bahasa Yunani yaitu *"cryptos"* berarti *"hidden"* (tersembunyi) atau *"crypto"* yang berarti *"secret"* (rahasia) dan *"graphy"* atau *"graphein"* yang berarti *"writing"* (tulisan). Jadi kriptografi artinya *"secret writing"* (tulisan rahasia) atau *"hidden writing"* (tulisan yang tersembunyi) [8]. Para pelaku atau praktisi kriptografi disebut sebagai kriptografer (*cryptographer).*

Dalam kriptografi, suatu pesan yang tidak disandikan disebut sebagai *plaintext* atau *cleartext.* Teknik untuk membuat *plaintext* menjadi kode-kode tertentu disebut sebagai enkripsi dan hasilnya disebut *ciphertext.* Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah *"encipher"*. Sedangkan teknik untuk membuat *chipertext* menjadi *plaintext* disebut dekripsi. Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah *"decipher"* [8,9]. Secara sederhana istilah-istilah di atas dapat digambarkan seperti Gambar 2.1.



Gambar 2.1. Proses Enkripsi dan Dekripsi Sederhana

Kebalikan dari kriptografi adalah *cryptanalysis*, yaitu seni atau ilmu untuk memecahkan rahasia suatu penyandian tanpa melalui cara yang seharusnya.

Pelaku atau praktisi yang menjalankan *cryptanalysis* disebut *cryptanalyst*. Cabang Matematika yang mencakup kriptografi dan *criptanalysis* disebut *criptology* dan pelakunya disebut *cryptologists*. Sedangkan *cryptographic system* atau kriptosistem *(cryptosystem*) adalah sebuah algoritma kriptografi ditambah seluruh kemungkinan *plaintext*, *ciphertex*t, dan kunci-kuncinya [3].

Menurut Wibowo [15], kriptografi yang baik tidak ditentukan oleh kerumitan dalam mengolah data atau pesan yang akan disampaikan, tetapi dapat memberikan kerahasiaan dalam komunikasi serta memberikan aspek-aspek keamanan sebagai berikut.

a. **Kerahasiaan.** Aspek ini menjamin bahwa pesan *(plaintext)* hanya dapat dibaca oleh pihak yang memiliki wewenang.
b. **Integritas.** Dalam aspek ini penerima pesan harus dapat memastikan bahwa pesan yang diterima tidak dimodifikasi ketika sedang dalam proses tranmisi data.
c. **Autentikasi.** Dalam aspek ini pengirim harus dapat diidentifikasi dengan pasti, penyusup harus dipastikan tidak bisa berpura-pura menjadi orang lain.
d. **Non-Repudiation** Aspek ini menjaga agar seseorang tidak dapat menyangkal telah mengirimkan sebuah informasi.

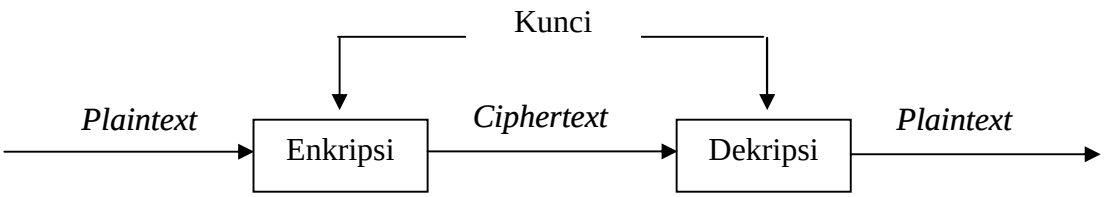
2. Algoritma Kriptografi

Algoritma kriptografi atau sering disebut *chiper* merupakan persamaan matematik yang digunakan untuk proses enkripsi dan dekripsi. Biasanya kedua persamaan matematik (untuk enkripsi dan dekripsi) tersebut memiliki hubungan matematis yang cukup erat. Algoritma kriptografi ini bekerja dengan menggunakan kunci yaitu sederetan bit yang diperlukan untuk mengenkripsi dan mendekripsi data (pesan) seperti kata, nomor maupun frase tertentu. Algoritma kriptografi modern tidak lagi mengandalkan keamanannya pada kerahasiaan algoritma tetapi kerahasiaan kunci. Jika dilakukan enkripsi pada *plaintext* yang sama dengan menggunakan kunci yang berbeda maka akan menjadi *chipertext* yang berbeda pula [7].

Berdasarkan jenis kuncinya, algoritma kriptografi dapat dibagi menjadi dua kelompok, yaitu :

a. Algoritma Simetri

Menurut Schneier [9] algoritma simetri adalah algoritma yang menggunakan kunci enkripsi sama dengan kunci dekripsinya. Algoritma ini mengharuskan pengirim dan penerima menyetujui suatu kunci tertentu sebelum mereka dapat berkomunikasi dengan aman. Algoritma kunci simetris sering juga disebut sebagai algoritma konvensional *(conventional algorithms),* algoritma kunci rahasia *(secret-key algorithms),* algoritma kunci tunggal *(single-key algorithms)* dan algoritma satu kunci *(one-key algorithms).* Proses enkripsi dan dekripsi dengan algoritma simetri dapat dilihat pada Gambar 2.2.



Gambar 2.2. Proses enkripsi dan deskripsi pada algoritma simetri

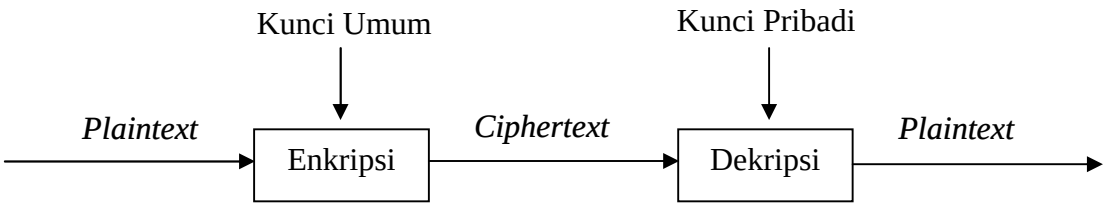
Menurut Kurniawan [3], algoritma kriptografi simetris dibagi menjadi dua kategori, yaitu : algoritma aliran (s*tream ciphers*) dan algoritma blok (*block ciphers*). Algoritma aliran beroperasi pada *plaintext* yang berupa satu bit tunggal pada satu waktu. Sedangkan pada algoritma blok beroperasi pada *plaintext* dalam grup-grup bit yang disebut blok. Keamanan algoritma simetri tergantung pada kunci, sehingga jika kunci bocor maka orang lain dapat mengenkripsi dan mendekripsi pesan. Contoh dari algoritma kunci simetris adalah DES, *Blowfish*, *Rijndael* (AES), *Gosudarstevennyi Standard* (GOST), *Rivest Code 2* (RC2), *Rivest Code 4* (RC4) dan lain-lain.

b. Algoritma Asimetri

Algoritma asimetri didesain sedemikian sehingga kunci yang digunakan untuk enkripsi berbeda dengan kunci yang digunakan untuk dekripsi. Kunci yang digunakan untuk enkripsi disebut kunci publik *(public key)* yang bisa diketahui oleh semua orang. Sedangkan kunci untuk dekripsi dinamakan kunci rahasia atau sering disebut sebagai *private key* dan hanya diketahui oleh pemiliknya.

Algoritma asimetri sering juga disebut algoritma kunci publik *(public–key algorithms).*

Semua orang dapat mengenkripsi pesan menggunakan kunci publik yang diketahui umum. Tetapi pesan yang telah terenkripsi hanya dapat didekripsi menggunakan *privat key* yang hanya diketahui oleh penerima pesan. Contoh dari algoritma asimetri adalah *ElGamal,* RSA*, Elliptic Curve Cryptography* (ECC)*, Digital Signature Algorithm* (DSA) dan lain-lain. Proses enkripsi-dekripsi dengan algoritma asimetris dapat dilihat pada Gambar 2.3.



Gambar 2.3. Proses enkripsi dan dekripsi algoritma asimetri

2.1.2 DES

1. Sejarah Singkat DES

Sejarah DES dimulai dari permintaan pemerintah Amerika Serikat untuk memasukkan proposal enskripsi. Menurut Stalling [11], DES memiliki sejarah dari algoritma Lucifer yaitu algoritma yang dibuat oleh Horst Feistel pada tahun 1971 ketika menjadi peneliti di IBM. Lucifer merupakan *cipher* blok yang beroperasi pada blok ukuran 64 bit dengan menggunakan kunci ukuran 128 bit. Kemudian pada tahun 1972 IBM mengembangkan algoritma DES dibawah kepemimpinan Walter Tuchman. DES baru secara resmi digunakan oleh pemerintah Amerika Serikat (diadopsi oleh NBS) pada tahun 1977. DES dikeluarkan oleh FIPS PUB46 dan disertifikasi setiap 5 tahun oleh NIST.

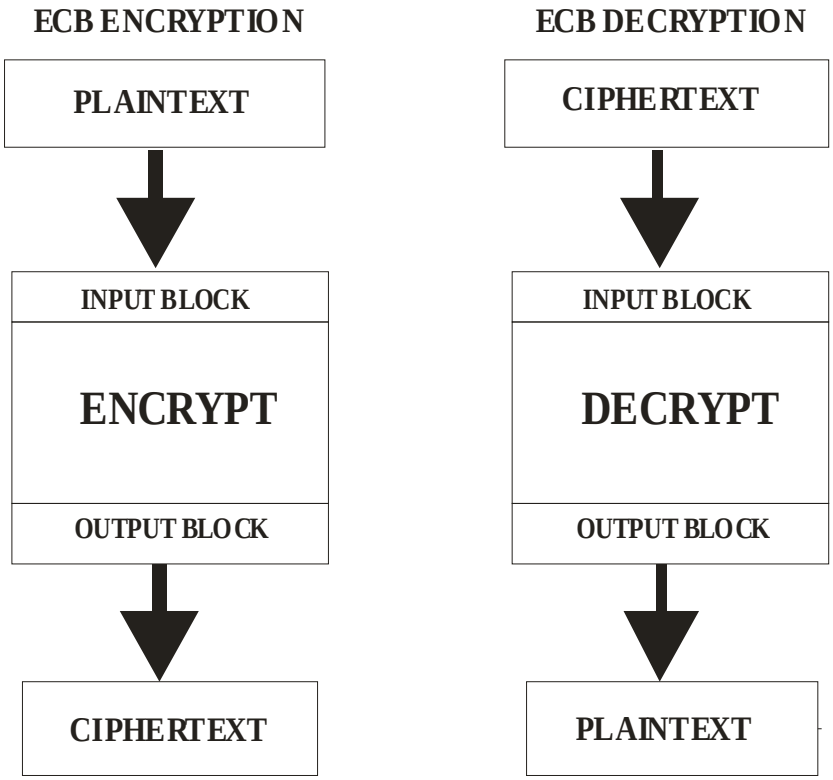
2. Mode Operasi DES

DES termasuk algoritma simetri jenis *cipher* blok dan beroperasi pada ukuran blok 64 bit. Menggunakan DES dalam beberapa aplikasi, diperlukan mode

untuk mengoperasikannya. Mode operasi ini bertujuan mengatasi keamanan cara penyandian dan mempermudah penyandian. Menurut Kurniawan [3] dan Wibowo [15], ada empat mode yang dapat digunakan dalam mengoperasikan DES yang didefinisikan oleh *Federal Information Processing Standard* 81 (FIPS 81) yaitu :

a. *Elektronic Codebook (*ECB)

ECB adalah mode operasi yang paling sederhana dan paling mudah untuk diimplementasikan. Cara yang digunakan adalah dengan membagi data ke dalam blok-blok data terlebih dahulu yang besarnya sudah ditentukan. Blok-blok data inilah yang disebut *plaintext* karena blok data ini belum disandikan. Proses enkripsi akan langsung mengolah *plaintext* menjadi *ciphertext* tanpa melakukan operasi tambahan. Suatu blok *plaintext* yang dienkripsi dengan menggunakan kunci yang sama akan menghasilkan *ciphertext* yang sama.



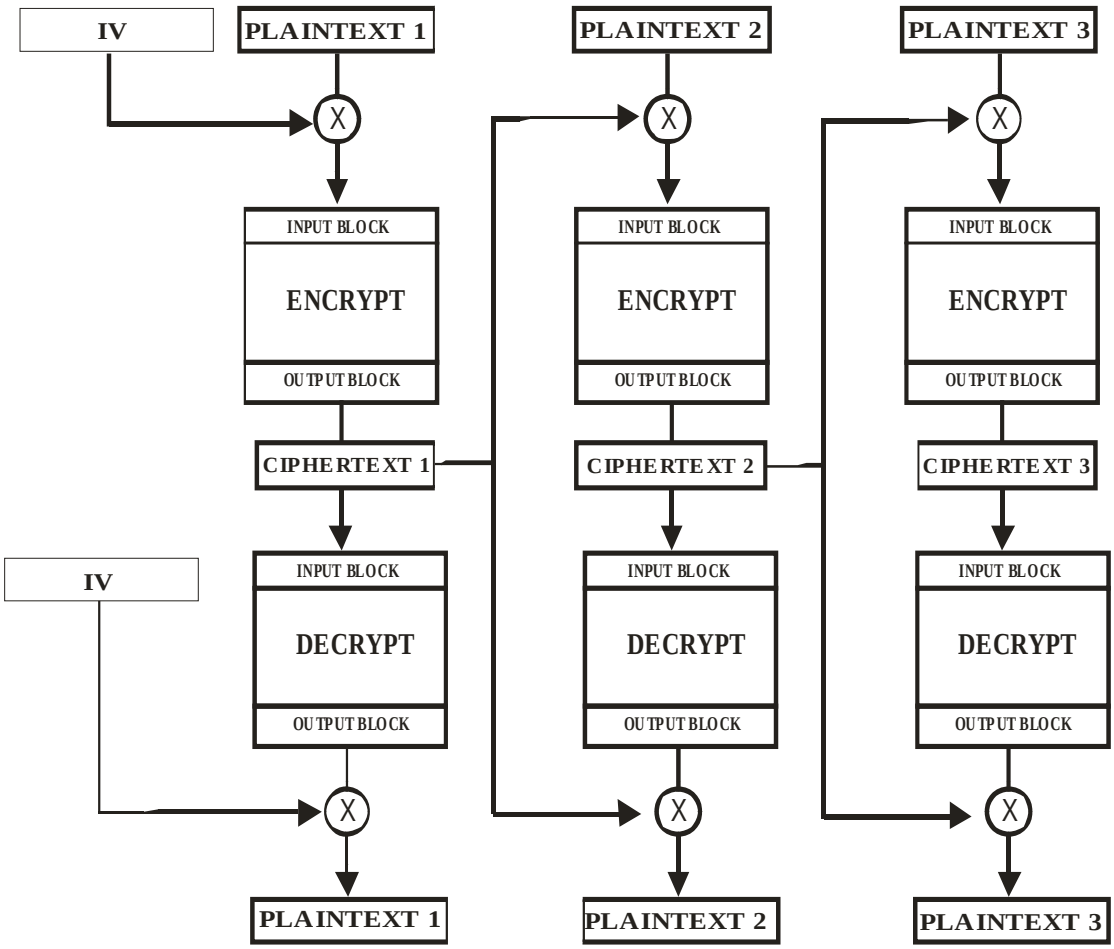
Gambar 2.4. Mode Operasi ECB

Keuntungan dari mode ECB ini adalah kemudahan dalam implementasi dan pengurangan resiko salahnya semua *plaintext* akibat kesalahan pada satu *plaintext*. Namun mode ini memiliki kelemahan pada aspek keamanannya.

Dengan mengetahui pasangan *plaintext* dan *ciphertext*, seorang *cryptanalyst* dapat menyusun suatu *codebook* tanpa perlu mengetahui kuncinya. ECB sangat ideal digunakan untuk data dengan jumlah data yang pendek seperti DES.

b. Cipher Blok Chaining(CBC)

Pada CBC digunakan operasi umpan balik atau dikenal dengan operasi berantai (*chaining*). Pada CBC, hasil enkripsi dari blok sebelumnya adalah *feedback* untuk enkripsi dan dekripsi pada blok berikutnya. Dengan kata lain, setiap blok *ciphertext* dipakai untuk memodifikasi proses enkripsi dan dekripsi pada blok berikutnya.

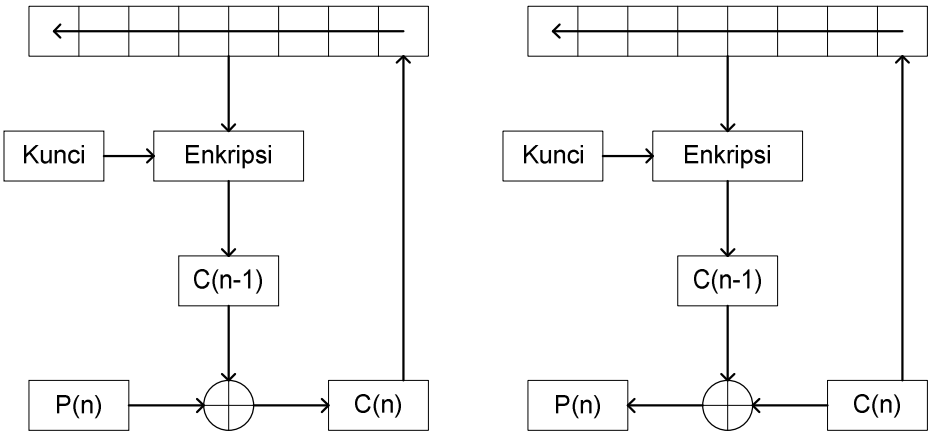


Gambar 2.5. Mode Operasi CBC

Pada CBC diperlukan data acak sebagai blok pertama. Blok data acak ini sering disebut *initialization vector* atau IV. IV digunakan hanya untuk membuat suatu pesan menjadi unik sehingga tidak perlu dirahasiakan.

c. *Cipher Feedback (CFB*)

Pada mode CBC, proses enkripsi atau dekripsi tidak dapat dilakukan sebelum blok data yang diterima lengkap terlebih dahulu. Masalah ini diatasi pada mode *Cipher Feedback* (CFB). Pada mode ini, data dapat dienkripsi pada unit-unit yang lebih kecil atau sama dengan ukuran satu blok. Misalkan pada CFB 8 bit, maka data akan diproses tiap 8 bit.



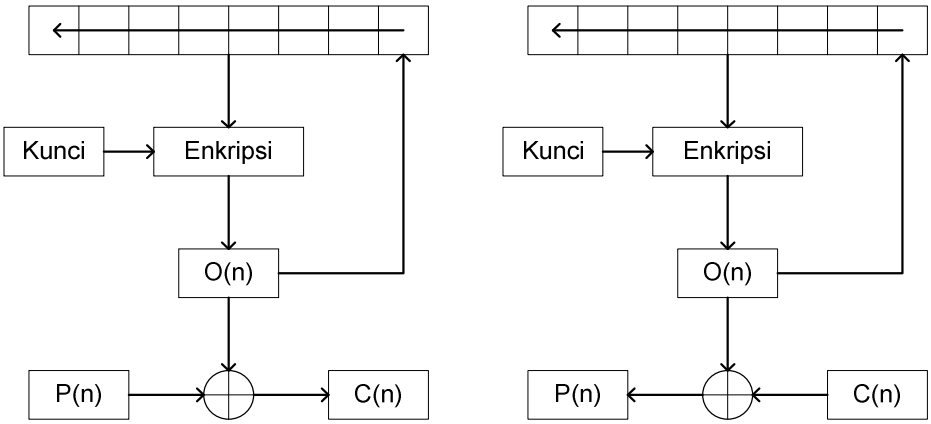
Gambar 2.6. Mode Operasi CFB

Pada permulaan proses enkripsi, IV dimasukkan dalam suatu *register* geser. IV ini dienkripsi dengan menggunakan kunci yang sudah ada. Dari hasil enkripsi tersebut, diambil 8 bit paling kiri atau *Most Significant Bit* untuk di-XOR dengan 8 bit dari *plaintext.* XOR adalah operasi *exlusive-or* dengan simbol ⊕ (Tabel 2.1). Hasil operasi XOR inilah yang menjadi *ciphertext* dimana *ciphertext* ini tidak hanya dikirim untuk ditransmisikan tetapi juga dikirim sebagai *feedback* ke dalam *register* geser untuk dilakukan proses enkripsi untuk 8 bit berikutnya.

d. Output Feedback (OFB)

Sama pada mode CFB, mode OFB juga memerlukan sebuah *register* geser dalam pengoperasiannya. Pertama kali, IV masuk ke dalam *register* geser dan dilakukan enkripsi. Hasil proses enkripsi diambil 8 bit paling kiri untuk dilakukan

XOR dengan *plaintext* yang menghasilkan *ciphertext*. *Ciphertext* tidak diumpan balik ke dalam *register* geser, tetapi yang diumpan balik adalah hasil dari enkripsi IV.



Gambar 2.7. Mode Operasi OFB

3. Elemen Pembangun dalam Algoritma DES

a. Aritmatika Modular

Aritmatika modular merupakan operasi matematika yang banyak diimplementasikan pada metode kriptografi. Pada metode kriptografi simetris, operasi aritmatika modular yang sering dipakai adalah operasi penjumlahan modulo dua dan operasi XOR *(exclusive-or*) dengan simbol ⊕. Operasi modulo dua ini melibatkan bilangan bilangan 0 dan 1 saja sehingga identik dengan bit pada komputer. Seluruh kemungkinan nilai operasi XOR dapat dilihat dalam Tabel 2.1.

Tabel 2.1 Operasi XOR pada modulo dua

a	b	a ⊕ b
0	0	0
0	1	1
1	0	1
1	1	0

Dari tabel di atas dapat dilihat sifat-sifat unik dari operasi XOR pada modulo dua yaitu : A ⊕ A = 0, A ⊕ 0 = A, A ⊕ 1 = A’, dengan A’ adalah komplemen dari A.

b. Jaringan Feistel (*Feistel Network*)

Jaringan Feistel adalah metode yang umum digunakan pada algoritma kriptografi *block cipher*. Bagian utama dari jaringan Feistel adalah fungsi *f*, yaitu fungsi pemetaan *string input* menjadi *string output*. Pada setiap putaran, blok sumber merupakan input bagi fungsi *f* , kemudian output dari fungsi *f* tersebut di XOR-kan dengan blok tujuan. Setelah itu kedua blok tersebut ditukar. Alasan digunakan jaringan Feistel yaitu *cipher* yang dibuat dengan fungsi ini dijamin dapat dikembalikan untuk proses dekripsi. Jaringan Feistel ini banyak digunakan oleh algoritma enkripsi seperti DES, Lucifer, FEAL, Khufu, LOKI, GOST, CAST, Blowfish dan lain-lain.

c. *Substitution Box* (S-Box)

S-Box merupakan suatu tabel subtitusi yang banyak digunakan pada kebanyakan algoritma blok cipher. S-Box pertama kali digunakan oleh Lucifer, kemudian DES dan setelah itu banyak digunakan dalam algoritma kriptografi yang lain, misalnya LOKI97. Dalam algoritma DES digunakan 8 buah S-Box yang dalam masing–masing tabelnya terdiri dari 4 baris dan 16 kolom. Setiap S-Box menerima input 6 bit dan menghasilkan output 4 bit. Kelompok 6 bit pertama menggunakan S_1, kelompok 6 bit kedua menggunakan S_2, dan seterusnya sampai 6 bit kedelapan menggunakan S_8. Detail S-Box yang digunakan dalam algoritma DES ini disajikan dalam Lampiran 2. Menurut penelitian para ahli kriptografi, DES didesain dengan sangat cermat sehingga bila ada yang mengubah-ubah S-Box ini secara acak, sangat mungkin DES yang dihasilkan justru menjadi lebih mudah untuk dibobol [3].

d. Jumlah Putaran DES

Penentuan banyak putaran pada algoritma kriptografi didasarkan atas azas keseimbangan. Semakin sedikit jumlah putaran akan menyebabkan algoritma kriptografi menjadi mudah untuk dipecahkan. Tetapi jika jumlah putaran semakin banyak akan menyebabkan kecepatan proses enkripsi atau dekripsi semakin berkurang. Putaran yang digunakan dalam Algoritma DES adalah 16 kali putaran. Hal ini didasarkan pada penelitian yang menunjukkan DES dengan jumlah

putaran kurang dari 16 kali dengan mudah dapat dipecahkan dengan *known-plaintext attack* [6].

e. Panjang Kunci DES

Panjang kunci eksternal DES adalah 64 bit atau 8 karakter. Kunci eksternal adalah kunci yang diberikan oleh pengguna sebelum proses enkripsi atau bersamaan dengan proses enkripsi. Setelah melalui proses permutasi, dari 64 bit kunci eksternal hanya 56 bit yang digunakan, sehingga dikatakan panjang kunci DES adalah 56 bit. Pada awal perancangan DES, panjang kunci yang diusulkan IBM 128 bit, tetapi atas permintaan NSA panjang kunci diperkecil menjadi 56 bit.

2.1.3 Teori Probabilitas

Dalam tulisan ini digunakan beberapa konsep dasar probabilitas, untuk itu perlu istilah-istilah probabilitas yang terkait. Setiap proses yang menghasilkan data mentah disebut eksperimen. Ruang sampel *(sample space)* yang dinotasikan dengan *S* adalah himpunan yang memuat semua peristiwa yang terjadi dalam eksperimen [1]. Sedangkan anggota dari suatu ruang sampel disebut titik sampel *(sample point)*. Suatu fungsi yang membawa setiap titik sampel dalam ruang sampel S ke suatu bilangan real disebut variabel random.

Kejadian adalah himpunan bagian suatu ruang sampel. Suatu proses disebut acak kalau proses itu tidak dapat ditentukan sebelumnya dengan pasti, sedangkan suatu kejadian disebut acak (*random event*) kalau terjadinya kejadian itu tidak dapat diketahui dengan pasti sebelumnya. Beberapa kejadian dikatakan saling meniadakan *(mutually exclusive)* jika kejadian-kejadian tersebut tidak dapat terjadi bersama-sama. Menurut Supranto [13], probabilitas adalah suatu nilai yang digunakan untuk mengukur tingkat terjadinya suatu kejadian yang acak.

Definisi 2.1 [Bain, [1]] *Jika suatu kejadian A dalam suatu eksperimen pada ruang sampel berhingga S yang setiap titik sampelnya berkemungkinan sama untuk muncul, maka probabilitas dari A yang dinotasikan dengan P(A), didefinisikan sebagai berikut.*

$$P(A) = \frac{n(A)}{n(S)}$$

Definisi di atas disebut sebagai definisi klasik dari probabilitas.

Definisi 2.2 [Bain, [1]] *Diberikan sebuah percoban, S suatu ruang sampel dan A, A_1, A_2,…menunjukkan kemungkinan kejadian. P(A) disebut probabilitas dari A, jika memenuhi syarat :*

$0 \le P(A)$ *untuk setiap A*

$P(S) = 1$

$$P(\tilde{\bigcup_{i=1}} A_i) = \tilde{\sum_{i=1}} P(A_i)$$

dengan A_1,A_2,…adalah kejadian yang saling meniadakan (mutually exclusive).

Probabilitas munculnya kejadian B jika kejadian A telah terjadi disebut probabilitas bersyarat *(conditional probability)* dan disajikan dengan P(B|A).

Definisi 2.3 [Menezes et al, [5]] *Jika A dan B adalah dua kejadian di dalam ruang sampel S dan $P(A) \neq 0$, maka peluang bersyarat kejadian B jika diketahui kejadian A didefinisikan sebagai berikut.*

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)}$$

$P(B \mid A)$ *adalah probabilitas kejadian B jika diberikan kejadian A.*

Dari definisi di atas dapat diturunkan aturan multiplikasi yang dituliskan dalam teorema berikut.

Teorema 2.4 [Bain, [1]] *Jika A dan B adalah dua kejadian di dalam ruang sampel S maka* $P(A \cap B) = P(B)P(A \mid B) = P(A)P(B \mid A)$

Teorema 2.5 [Walpole et.al, [14]] ***(Probabilitas Total)*** *Jika B_1, B_2,…, B_k merupakan partisi dari ruang sampel S dengan $P(B) \neq 0$ untuk i=1,2,…k maka untuk setiap kejadian A dari ruang sampel S*

$$P(A) = \sum_{i=1}^{k} P(B_i \cap A) = \sum_{i=1}^{k} P(B_i)P(A \mid B_i)$$

Teorema 2.6 [Walpole et.al, [14]] **(*Teorema Bayes)*** *Jika kejadian B_1, B_2,…, B_k membentuk partisi di dalam ruang sampel S, dimana $P(B_i) \neq 0$ untuk i=1,2,…k maka untuk sebarang kejadian A di dalam S dengan $P(A) \neq 0$ berlaku,*

$$P(B_i \mid A) = \frac{P(B_i)P(A \mid B_i)}{\sum_{i=1}^{k} P(B_i)P(A \mid B_i)}$$

Dua kejadian A dan B dalam ruang sampel S disebut saling bebas atau saling independen apabila muncul atau tidak munculnya kejadian A tidak mempengaruhi muncul atau tidak munculnya kejadian B, atau sebaliknya.

Definisi 2.7 [Supranto, [13]] *Jika A dan B adalah kejadian saling bebas, maka berlaku* $P(A \cap B) = P(A)P(B) = P(B)P(A)$

Distribusi seragam deskrit *(discrete uniform distribution)* merupakan distribusi variabel random diskrit yang mengasumsikan semua nilai mempunyai kemungkinan yang sama untuk muncul.

Definisi 2.8 [Bain, [1]] *Jika pada variabel random X nilai-nilai* $x_1, x_2, x_3, \ldots, x_n$ *mempunyai peluang yang sama, maka variabel random X disebut berdistribusi seragam jika fungsi peluangnya berbentuk*

$$f(x;k) = \frac{1}{k}$$

untuk $X = x_1, x_2, x_3, \ldots, x_n$.

Teori probabilitas di atas, dalam skripsi ini digunakan untuk menentukan probabilitas munculnya setiap digit PIN dalam sebuah kartu ATM.

2.2 Kerangka Pemikiran

Berdasarkan tinjauan pustaka yang telah dipaparkan dapat disusun suatu kerangka pemikiran untuk melakukan pembahasan dalam tulisan ini. Dengan melihat latar belakang masalah, algoritma DES merupakan salah satu algoritma simetris yang paling umum digunakan saat ini karena mempunyai kelebihan kecepatan dalam enskripsi. Salah satu aplikasi algoritma DES dalam perbankan yaitu untuk membuat persandian pada kartu ATM.

Pada tulisan ini secara garis besar disajikan algoritma DES yang digunakan untuk menentukan empat digit PIN sebuah kartu ATM. Algoritma DES akan mengenskripsi suatu *plaintext* berupa 16 digit hexadesimal (64 bit) dari data *magnetic stripe* pada sebuah kartu ATM dengan kunci rahasia 56 bit, menjadi

sebuah *chipertext* yang ditulis sebagai 16 digit hexadesimal. Kemudian diambil empat digit, yaitu digit ke tiga sampai dengan digit ke enam dari 16 digit hexadesimal tersebut dan mengganti semua huruf hexadesimal A sampai F berturut-turut dengan angka 0 sampai 5. Setelah proses penentuan PIN, selanjutnya ditunjukkan cara untuk memecahkan rahasia penyandian algoritma DES pada kartu ATM Eurocheque menggunakan teori probabilitas.

BAB III
METODE PENELITIAN

Metode penelitian yang digunakan dalam penulisan skripsi ini adalah studi literatur dan simulasi, yaitu memberikan ilustrasi tentang alur dari algoritma DES dan aplikasinya dalam kartu ATM serta *cryptanalysis* aplikasi tersebut menggunakan teori –teori dalam probabilitas.

Untuk mencapai tujuan penelitian ini, dapat dilakukan langkah-langkah sebagai berikut :

1. Studi literatur, tentang
   a. Kriptografi,
   b. DES,
   c. Konsep dasar teori probabilitas.
2. Enkripsi dan Dekripsi data dengan DES,

   yaitu mendeskripsikan algoritma DES yang digunakan untuk mengenkripsi suatu ***plaintext*** yang berupa 16 digit hexadesimal menjadi ***chipertext*** yang juga berupa 16 digit hexadesimal, dan sebaliknya. Selanjutnya diaplikasikan dalam sebuah contoh kasus.
3. Aplikasi algoritma DES dalam sistem keamanan ATM,

   yaitu mendeskripsikan aplikasi algoritma DES yang digunakan untuk menentukan empat digit PIN dalam suatu kartu ATM.
4. *Cryptanalysis* terhadap sistem keamanan ATM,

   yaitu menebak empat digit PIN dalam suatu kartu ATM yang digunakan oleh nasabah menggunakan teori-teori dalam probabilitas. Selanjutnya diaplikasikan dalam sebuah contoh kasus.

BAB IV
PEMBAHASAN

Dalam pembahasan ini dijelaskan tentang aplikasi algoritma DES yang digunakan untuk menentukan empat digit PIN dalam sebuah kartu ATM. Selanjutnya ditunjukkan cara memecahkan rahasia penyandian pada sistem keamanan ATM menggunakan teori probabilitas. Sebelum itu, terlebih dahulu dibahas tentang algoritma DES.

4.1 Algoritma DES

DES merupakan nama dari sebuah algoritma untuk mengenskripsi data yang dikembangkan di IBM dibawah kepemimpinan W.L. Tuchman pada tahun 1972. Algoritma ini didasarkan pada algoritma Lucifer yang dibuat oleh Horst Feistel. Algoritma DES telah disetujui oleh NBS setelah dinilai kekuatanya NSA Amerika Serikat. NIST melakukan sertifikasi terhadap algorima DES setiap lima tahun [11]. Dalam pembahasan ini algoritma DES yang dipaparkan diambil dari buku *Data Encryption Standard* yang ditulis oleh Rinaldi Munir [6].

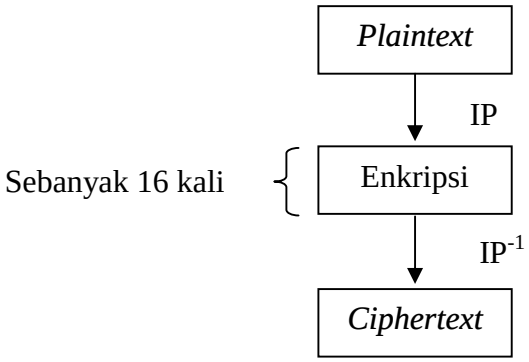
Algoritma DES merupakan algoritma simetris yang paling banyak digunakan di dunia untuk proses enkripsi. DES bekerja pada bit atau bilangan biner yaitu 0 dan 1. Sebelum dienkripsi menggunakan algoritma DES , sebuah *plaintext* yang berupa bilangan hexadesimal (bilangan berbasis 16) akan di konversi terlebih dahulu kedalam 64 bit bilangan biner dengan masing-masing grup tersusun dari 4 bit. Sebagai contoh, bilangan hexadesimal “0” akan dikonversi kedalam bilangan biner “0000”, bilangan hexadesimal “6” akan dikonversi kedalam bilangan biner “0110” dan bilangan hexadesimal “B” akan dikonversi kedalam bilangan biner “1011”. Konversi bilangan hexadesimal ke bilangan biner selengkapnya disajikan dalam Tabel 4.1. Begitu juga setelah proses enkripsi DES, hasil enkripsi yang berupa bilangan biner dikonversi kedalam bilangan hexadesimal. Jadi *plaintext* dan *ciphertext* yang merupakan input dan output dalam algoritma DES ini berupa bilangan hexadesimal.

Tabel 4.1 Konversi bilangan hexadesimal, biner dan desimal

Bilangan Hexadesimal	Bilangan Biner	Bilangan Desimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Menurut Munir [6], algoritma DES termasuk algoritma simetri jenis *cipher* blok dan beroperasi pada ukuran blok 64 bit. DES mengenkripsikan 64 bit *plaintext* menjadi 64 bit *ciphertext* menggunakan 56 bit kunci internal (*internal key)* sebanyak 16 putaran. Kunci internal ini dibangkitkan dari kunci eksternal *(external key)* yang panjangnya 64 bit. Skema global dari algoritma DES adalah sebagai berikut (Gambar 4.1).

1. Blok *plaintext* dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP).
2. Hasil permutasi awal kemudian dienkripsi sebanyak 16 putaran. Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil enkripsi kemudian dipermutasi dengan matriks permutasi balikan *(invers initial permutation* atau IP^{-1}) menjadi blok *ciphertext.*



Gambar 4.1. Skema Global Algoritma DES

Di dalam proses enkripsi, blok *plaintext* dibagi menjadi dua bagian yaitu 32 bit kiri (blok *L*) dan 32 bit kanan (blok *R*). Kemudian kedua bagian ini masuk ke dalam 16 putaran DES.

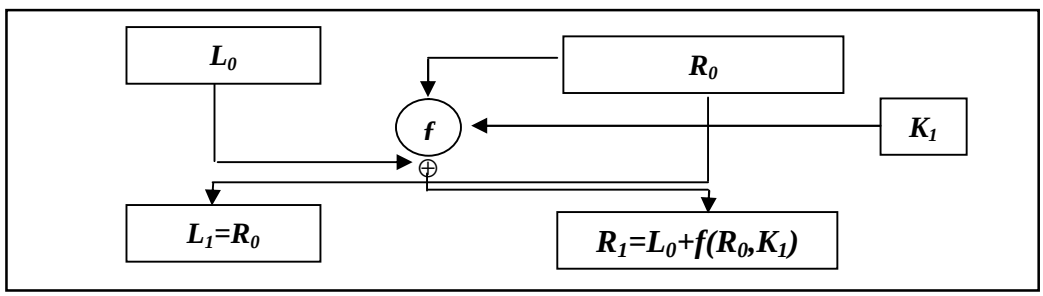
Pada setiap putaran *i*, blok *R* merupakan input untuk fungsi transformasi yang disebut *f*. Pada fungsi *f*, blok *R* dikombinasikan dengan kunci internal K_i. Output dari fungsi *f* di-XOR-kan dengan blok *L* untuk mendapatkan blok R yang baru. Sedangkan untuk blok *L* yang baru, langsung diambil dari blok R sebelumnya. Ini adalah satu putaran DES yang secara matematis dinyatakan sebagai berikut:

$$L_i = R_{i-1}$$

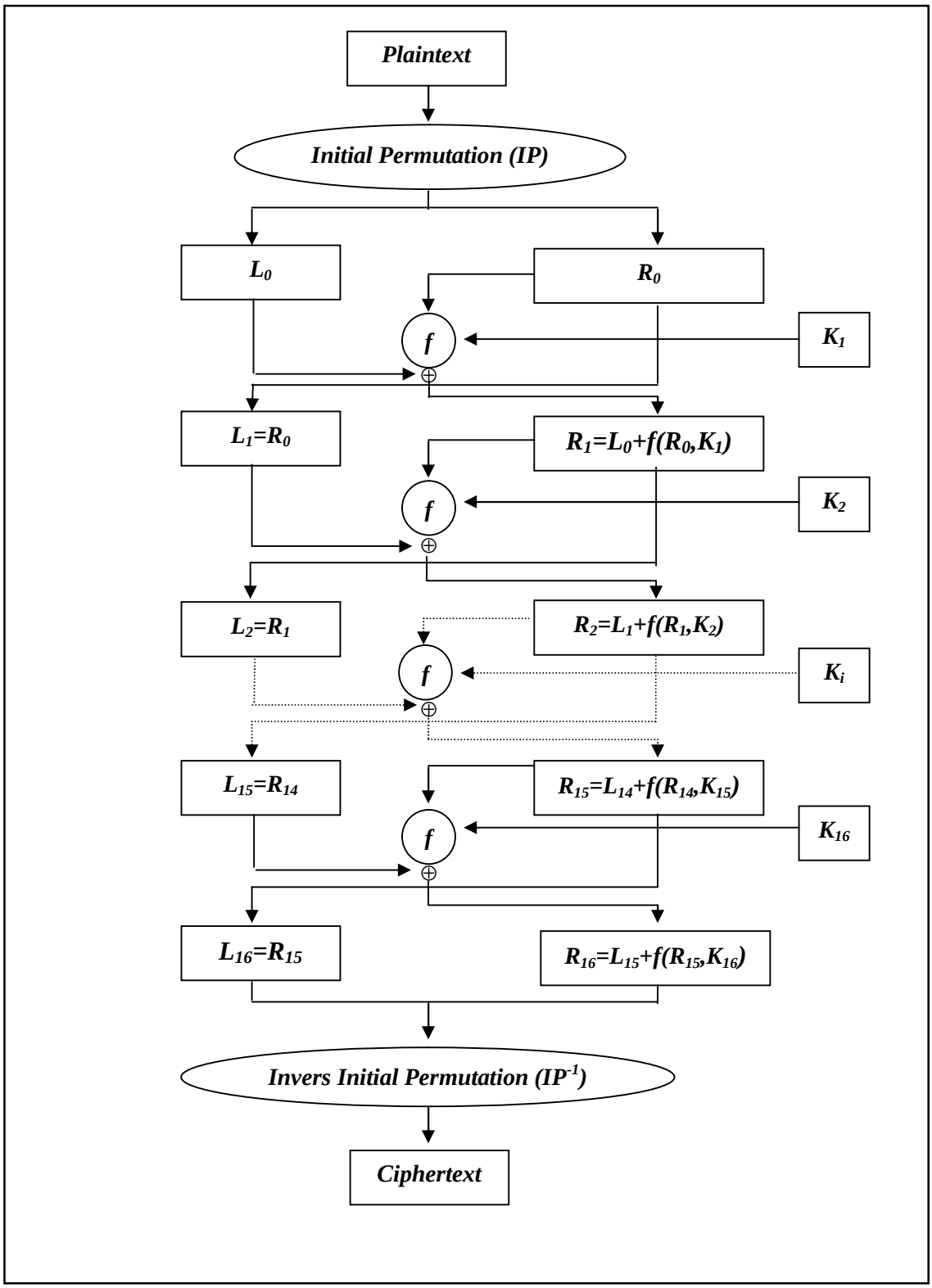
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \qquad (4.1)$$

Satu putaran DES ini akan diulang sebanyak 16 kali putaran. Skema dari diagram alir enkripsi menggunakan algoritma DES secara lebih rinci disajikan dalam Gambar 4.3.

Sebagai catatan, satu putaran DES merupakan model jaringan Feistel seperti dalam Gambar 4.2 berikut.



Gambar 4.2 Jaringan Feistel untuk satu putaran DES



Gambar 4.3. Diagram Alir Enkripsi menggunakan Algoritma DES

Perlu diketahui dari Gambar 4.3 setelah algoritma DES melakukan 16 kali putaran, jika (L_{16} , R_{16}) merupakan hasil dari putaran ke-16, (R_{16},L_{16}) merupakan pra-*ciphertext* dari proses enkripsi algoritma DES . *Chipertext* yang sebenarnya dapat diperoleh dengan melakukan *invers* permutasi awal (IP^{-1}) terhadap blok pra-*ciphertext* tersebut.

4.1.1. Permutasi Awal

Sebelum melakukan putaran pertama dalam proses enkripsi DES, terlebih dahulu dilakukan permutasi awal (IP) terhadap input yang berupa blok *plaintext* yang panjangnya 64 bit. Tujuan dari permutasi awal ini adalah untuk mengacak *plaintext* tersebut sehingga urutan bit-bit yang berada di dalamnya berubah. Pengacakan ini dilakukan menggunakan matriks permutasi awal sebagaimana dalam Tabel 4.2 sebagai berikut.

Tabel 4.2 *Initial Permutation (IP)*

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

Penerapan IP pada sebuah *plaintext* P adalah dengan memindahkan bit ke-58 dari P menjadi bit ke-1 pada IP, bit ke-50 dari P menjadi bit ke-2 pada IP, dan seterusnya sampai dengan bit ke-7 dari P menjadi bit ke-64 pada IP. Selanjutnya hasil permutasi dari *plaintext* P di enkripsi sebanyak 16 kali putaran menggunakan kunci internal yang berbeda.

4.1.2. Pembangkit Kunci Internal

Setiap blok *plaintext* mengalami 16 kali putaran pada proses enkripsi, untuk itu dibutuhkan kunci internal sebanyak 16 buah, yaitu $K_1, K_2,\ldots, K_{16}$. Kunci-kunci internal ini dapat dibangkitkan dari kunci eksternal yang diberikan oleh pengguna sebelum proses enkripsi atau bersamaan dengan proses enkripsi. Kunci eksternal yang panjangnya 64 bit atau 8 karakter menjadi input untuk permutasi menggunakan permutasi kompresi 1 (PC-1) sebagaimana dalam Tabel 4.3. Permutasi kompresi menggunakan Tabel 4.3 ini menghasilkan *K*+.

Tabel 4.3 Permutasi Kompresi 1

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

Dalam permutasi pada Tabel 4.3, setiap bit kelipatan delapan *(parity bit)* dari delapan byte kunci diabaikan, yaitu bit 8, 16, 24, 32, 40, 48, 56, dan 64. Hasil permutasi kompresi sepanjang 56 bit, sehingga dapat dikatakan kunci DES adalah 56 bit.

Input pertama dalam Tabel 4.3 adalah 57 yang artinya bit ke-57 pada kunci *K* menjadi bit pertama dalam permutasi *K*+. Bit ke 49 pada kunci K menjadi bit kedua dalam permutasi K+, dan seterusnya bit ke-4 pada kunci *K* menjadi bit terakhir dalam permutasi K+.

Selanjutnya 56 bit *K*+ ini akan dibagi menjadi 2 blok yaitu blok kiri (C_0) dan blok kanan (D_0), dimana masing-masing panjangnya 28 bit yang tersimpan dalam C_0 dan D_0.

C_0 berisi bit-bit dari $K+$ pada posisi :

57, 49, 41, 33, 25, 17, 9, 1, 58, 50, 42, 34, 26, 18

10, 2, 59, 51, 43, 35, 27, 19, 11, 3, 60, 52, 44, 36,

D_0 berisi bit-bit dari $K+$ pada posisi :

63, 55, 47, 39, 31, 23, 15, 7, 62, 54, 46, 38, 30, 22

14, 6, 61, 53, 45, 37, 29, 21, 13, 5, 28, 20, 12, 4.

Setelah itu kedua blok tersebut digeser ke kiri (*left shift*) sepanjang satu bit untuk i=1, 2, 9 atau 16 dan digeser sepanjang dua bit untuk i yang lain sehingga diperoleh C_i dan D_i untuk $1 \leq i \leq 16$. Jumlah pergeseran pada setiap putaran ditunjukkan pada Tabel 4.4.

Tabel 4.4 Jumlah pergeseran bit pada setiap putaran

Putaran (i)	Jumlah pergeseran bit
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

Misalkan (C_i,D_i) menyatakan penggabungan C_i dan D_i , (C_{i+1},D_{i+1}) dapat diperoleh dengan menggeser C_i dan D_i sepanjang satu atau dua bit.

Setelah pergeseran bit sebanyak 16 putaran di atas, (C_i,D_i) akan mengalami permutasi kompresi menggunakan matriks PC-2 sebagaimana dalam Tabel 4.5. Dengan permutasi kompresi ini pasangan (C_i,D_i) yang berukuran 56 bit diturunkan menjadi kunci internal K_i (untuk $1 \leq i \leq 16$) yang berukuran 48 bit.

Dalam hal ini K_i merupakan penggabungan dari bit-bit C_i pada posisi :

14, 17, 11, 24, 1, 5, 3, 28, 15, 6, 21, 10

23, 19, 12, 4, 26, 8, 16, 7, 27, 20, 13, 2,

dengan bit-bit D_i pada posisi :

41, 52, 31, 37, 47, 55, 30, 40, 51, 45, 33, 48

44, 49, 39, 56, 34, 53, 46, 42, 50, 36, 29, 32.

Jadi, setiap kunci internal K_i mempunyai panjang 48 bit.

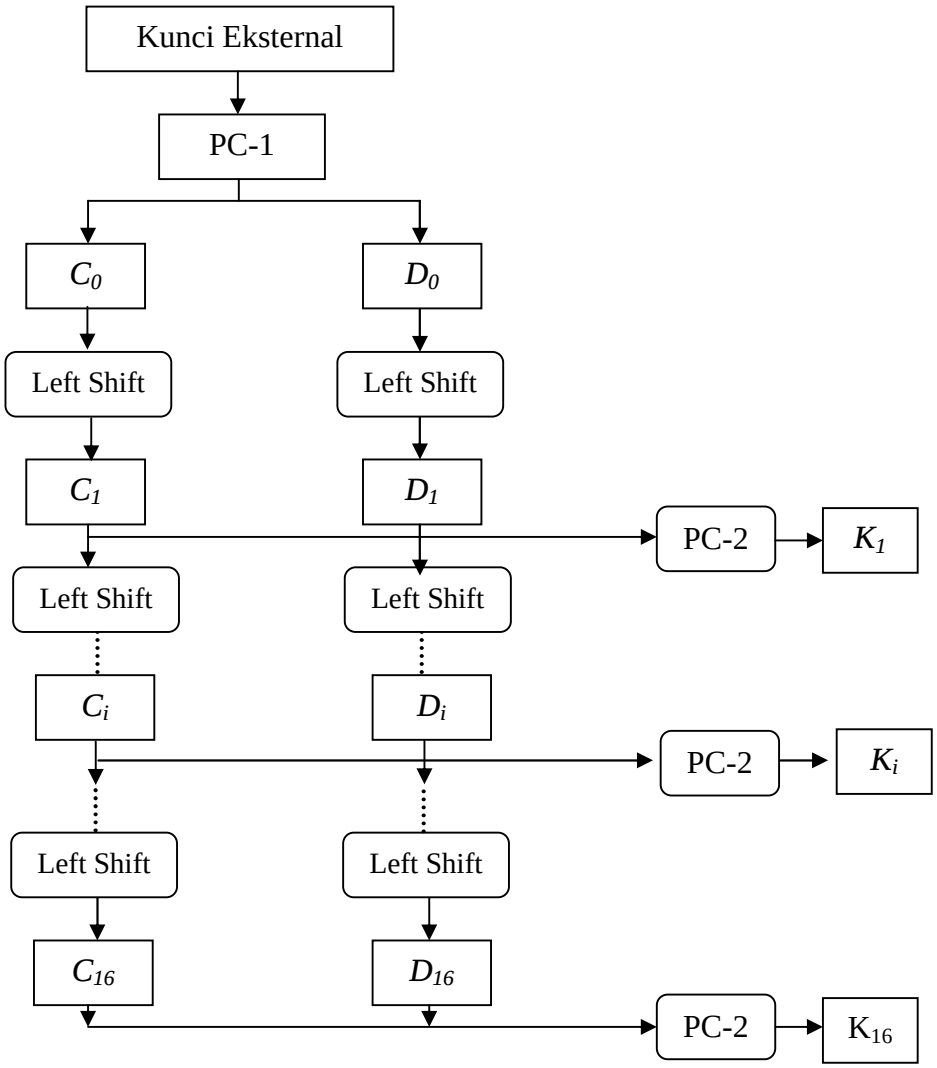
Tabel 4.5 Permutasi Kompresi 2

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

Input pertama dalam Tabel 4.5 adalah 14 yang artinya bahwa bit ke-14 dari (C_i,D_i) merupakan bit ke-1 dari K_i, bit ke-17 dari (C_i,D_i) merupakan bit ke-2 dari K_i, dan seterusnya sampai bit ke-32 dari (C_i,D_i) merupakan bit ke-48 dari K_i.

Bila jumlah pergeseran bit-bit pada Tabel 4.3 dijumlahkan semua, jumlah seluruhnya menjadi 28 yang sama dengan jumlah bit pada C_i dan D_i. Setelah putaran ke-16 didapatkan kembali $C_{16} = C_0$ dan $D_{16} = D_0$.

Pembangkitan kunci internal untuk proses dekripsi sama seperti pada proses enkripsi dengan pergeseran ke kiri diganti pergeseran ke kanan *(right shift)*. Detail proses pembangkitan 16 buah kunci internal K_i dari kunci eksternal ini ditunjukkan dalam Gambar 4.4.

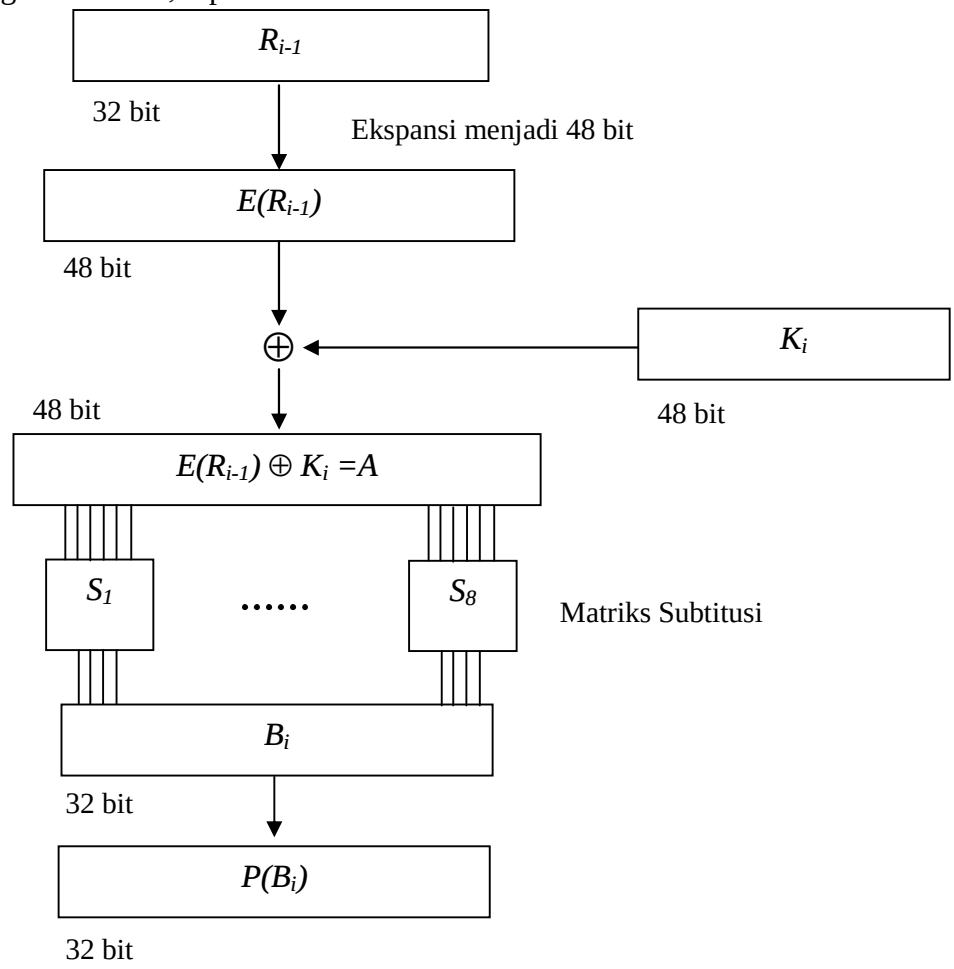


Gambar 4.4 Diagram alir pembangkitan kunci-kunci internal DES

4.1.3. Proses Enkripsi DES

Sebagaimana dijelaskan oleh Munir [6] proses enkripsi terhadap blok *plaintext* yang panjangnya 64 bit dilakukan setelah permutasi awal terhadap blok *plaintext* tersebut (Gambar 4.1). Setelah itu blok *plaintext* dibagi menjadi dua bagian yaitu kiri (*L*) dan kanan (*R*) yang masing-masing panjangnya 32 bit. Kemudian kedua bagian ini mengalami 16 kali putaran dengan setiap putaran menggunakan kunci internal yang berbeda. Setiap satu putaran merupakan jaringan Feistel yang secara matematis dinyatakan dalam Persamaan 4.1. Dalam

persamaan tersebut fungsi *f* merupakan fungsi tranformasi yang merupakan inti dari algoritma DES, diperlihatkan dalam Gambar 4.5.



Gambar 4.5 Diagram alir fungsi *f* (Algoritma DES)

Dalam gambar tersebut R_{i-1} diperoleh dari blok kanan sebuah ***plaintext*** kemudian menjadi input dari transformasi ***f.*** Fungsi ekspansi ***E*** berguna untuk memperluas blok R_{i-1} yang panjangnya 32 bit menjadi blok yang panjangnya 48 bit dengan matriks permutasi ekspansi ***(expansion permutation)*** sebagaimana dalam Tabel 4.6. Dalam matriks permutasi ekspansi ini blok input diperluas dengan melakukan penambahan sejumlah 16 bit (bilangan yang dicetak tebal) ke dalam Tabel 4.6.

Input pertama dalam Tabel 4.6 adalah 32 yang artinya bit ke-32 dari R_{i-1} merupakan bit ke-1 dari $E(R_{i-1})$, bit ke-1 dari R_{i-1} merupakan bit ke-2 dari $E(R_{i-1})$ dan seterusnya sampai bit ke-1 dari R_{i-1} merupakan bit ke-48 dari $E(R_{i-1})$.

Tabel 4.6 Permutasi Ekspansi

E					
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

Selanjutnya hasil ekspansi, $E(R_{i-1})$, yang panjangnya 48 bit di-XOR-kan dengan kunci internal K_i yang panjangnya 48 bit menghasilkan vektor A yang panjangnya 48 bit.

$$E(R_{i-1}) \oplus K_i = A \qquad (4.2)$$

Vektor A tersebut dikelompokkan menjadi delapan kelompok yang masing-masing kelompok terdiri dari 6 bit. Misalkan $A=A_1A_2A_3A_4A_5A_6A_7A_8$ dengan A_1 sampai dengan A_8 merupakan kelompok dari A. Delapan kelompok tersebut akan menjadi input untuk proses subtitusi yang dilakukan menggunakan delapan buah S-Box yaitu $S_1, S_2, \ldots S_8$ (Lampiran 1). Setiap S-Box menerima input sebanyak 6 bit dan menghasilkan output sebanyak 4 bit. Kelompok 6 bit pertama menggunakan S_1, kelompok 6 bit kedua menggunakan S_2, dan seterusnya sampai kelompok 6 bit kedelapan menggunakan S_8.

Misalkan delapan kelompok A dinyatakan dengan A_j $(1 \le j \le 8)$, $A_j=a_1a_2a_3a_4a_5a_6$ dengan a_1 sampai dengan a_6 merupakan 6 bit dalam A_j. Bit pertama dan terakhir dalam A_j (a_1 dan a_6) digunakan untuk menunjukkan indek baris ke-r dari S_j (S-Box ke-j) dengan $0 \le r \le 3$. Sedangkan empat bit yang tengah ($a_2a_3a_4a_5$) digunakan untuk menunjukkan indek kolom ke-c dari S_j dengan $0 \le c \le 15$.

Output dari subtitusi S-Box ini adalah vektor B yang panjangnya 32 bit. Vektor B terbagi menjadi delapan kelompok yang masing-masing kelompok terdiri dari empat bit. Misalkan $B=B_1B_2B_3B_4B_5B_6B_7B_8$ dengan B_1 sampai dengan

B_8 merupakan kelompok dari B. Secara umum kelompok dari B tersebut dapat dinyatakan dengan B_j. Untuk memperoleh B_j dilakukan dengan menggunakan A_j sebagai input dalam proses S-Box. Secara umum proses mendapatkan output dari S-Box dinyatakan dalam Persamaan 4.3.

$$B_j = S_j (A_j) = S_j (r,c) \text{ untuk } 1 \leq j \leq 8 \quad (4.3)$$

Selanjutnya vektor ***B*** ini menjadi input untuk proses permutasi menggunakan matriks permutasi P (P-Box) seperti Tabel 4.7 sehingga dihasilkan ***P(B)***. Tujuan permutasi P ini adalah untuk mengacak hasil proses subtitusi S-Box.

Tabel 4.7 P *Permutation*

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

Input pertama dalam Tabel 4.7 adalah 16 yang artinya bit ke-16 dari ***B*** merupakan bit ke-1 dari ***P(B),*** bit ke-7 dari ***B*** merupakan bit ke-2 dari ***P(B),*** dan seterusnya sampai bit ke-25 dari ***B*** merupakan bit ke-32 dari ***P(B).***

P(B) yang panjangnya 32 bit ini merupakan output dari fungsi ***f***.

$$P(B) = f(R_{i-1}, K_i) \quad (4.4)$$

Selanjutnya ***P(B)*** di-XOR-kan dengan L_{i-1} sehingga diperoleh R_i.

$$R_i = L_{i-1} \oplus P(B)$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Jadi output dari putaran ke-*i* untuk 1≤*i*≤16 adalah,

$$(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus f(R_{i-1}, K_i)) \quad (4.5)$$

Dari Gambar 4.3 dapat diketahui jika (L_{16},R_{16}) merupakan hasil dari putaran ke-16, maka (R_{16},L_{16}) merupakan pra-*ciphertext* dari proses enkripsi algoritma DES . *Chipertext* yang sebenarnya dapat diperoleh dengan melakukan *invers* permutasi awal (IP^{-1}) terhadap blok pra-*ciphertext* tersebut.

4.1.4. Permutasi Terakhir

Permutasi terakhir ini dilakukan setelah 16 kali putaran terhadap gabungan blok kiri dan blok kanan. Hasil setelah dilakukan 16 kali putaran tersebut adalah (L_{16},R_{16}). Kemudian hasil dari putaran ini di permutasi menggunakan matriks permutasi awal balikan seperti dalam Tabel 4.8.

Tabel 4.8 *Invers Initial Permutation*

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

Input pertama dalam Tabel 4.8 adalah 40 yang artinya bahwa bit ke-40 dari (L_{16},R_{16}) merupakan bit ke-1 dari IP^{-1}, bit ke-8 dari (L_{16},R_{16}) merupakan bit ke-2 dari IP^{-1} dan seterusnya sampai bit ke-25 dari (L_{16},R_{16}) merupakan bit ke-64 dari IP^{-1}.

Perlu diketahui, tabel permutasi *IP, Invers IP, S-Box, P, E, PC-1 dan PC-2* pada algoritma DES isinya merupakan ketetapan dari algoritma DES [5].

4.1.5. Proses Dekripsi DES

Seorang penerima pesan harus melakukan proses dekripsi terhadap *ciphertext* yang diperoleh untuk mengetahui *plaintext* yang dikirimkan. Proses dekripsi terhadap suatu *ciphertext* merupakan kebalikan dari proses enkripsi. Dalam hal ini DES menggunakan algoritma yang sama untuk melakukan proses enkripsi dan dekripsi. Jika pada proses enkripsi urutan kunci internal yang digunakan adalah $K_1, K_2,\ldots, K_{16}$, maka pada proses dekripsi urutan kunci yang digunakan adalah $K_{16}, K_{15},\ldots, K_1$.

Proses dekripsi DES dimulai dengan putaran ke-16, 15,…,2, 1. Untuk setiap putaran dalam proses dekripsi, dihasilkan output yang secara matematis dapat dinyatakan dalam Persamaan 4.6. Persamaan ini diturunkan dari Persamaan 4.1 yang merupakan output dari setiap putaran dalam proses enkripsi.

$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus f(L_i, K_i) \qquad (4.6)$$

Dalam hal ini (R_{16}, L_{16}) adalah input awal untuk proses dekripsi. Untuk memperoleh blok (R_{16}, L_{16}) maka dilakukan permutasi terhadap *ciphertext* menggunakan matriks permutasi IP^{-1} seperti dalam Tabel 4.8. Output terakhir dari proses dekripsi adalah (L_0,R_0). Selanjutnya dilakukan permutasi terhadap (L_0,R_0) menggunakan matriks permutasi awal pada Tabel 4.2 sehingga diperoleh kembali blok *plaintext* semula.

Proses pembangkitan kunci internal dalam proses dekripsi pada prinsipnya hampir sama dengan proses enkripsi sebagaimana dalam Gambar 4.4. Gambar tersebut menunjukkan K_{16} dihasilkan dari (C_{16}, D_{16}) setelah mengalami permutasi kompresi menggunakan matriks PC-2 dalam Tabel 4.5. Sebagaimana telah dijelaskan di depan bahwa setelah putaran ke-16 dari proses pembangkitan kunci internal, didapatkan kembali $C_{16} = C_0$ dan $D_{16} = D_0$, sehingga K_{16} dapat dihasilkan dari (C_0, D_0) tanpa harus melakukan pergeseran bit lagi. Perlu diingat (C_0, D_0) merupakan bit-bit dari kunci eksternal K yang diberikan oleh pengguna pada waktu proses dekripsi.

Selanjutnya K_{15} diperoleh dari (C_{15}, D_{15}). Untuk mendapatkan (C_{15}, D_{15}) diperoleh dengan cara menggeser C_{16} dan D_{16} satu bit ke kanan. Untuk K_{14} sampai

K_1 diperoleh dari (C_{14}, D_{14}) sampai (C_1, D_1). Secara umum dituliskan untuk mendapatkan (C_{i-1}, D_{i-1}) diperoleh dengan cara menggeser C_i dan D_i sepanjang satu atau dua bit menggunakan Tabel 4.4, tetapi pergeseran ke kiri diganti menjadi pergeseran ke kanan.

4.1.6. Contoh Kasus

Untuk memperjelas bagaimana proses enkripsi dan dekripsi menggunakan algoritma DES, berikut diberikan satu contoh kasus implementasi algoritma DES dalam proses enkripsi dan dekripsi. Dalam contoh kasus ini, untuk *plaintext* dan kunci eksternalnya diambilkan dari buku *Cryptography Theory and Practice* yang ditulis oleh Douglas R Stinson [12].

Ahmad bermaksud mengirimkan sebuah pesan melalui jaringan tertentu kepada Ibrahim. Karena Ahmad menginginkan hanya Ibrahim yang tahu pesannya, dia melakukan enkripsi pada pesan *plaintext*-nya menggunakan algoritma DES sehingga dihasilkan *ciphertext*. Ahmad berharap hanya Ibrahim yang bisa melakukan dekripsi terhadap *ciphertext* yang dikirimnya. Pesan Ahmad berupa *plaintext* P = 0123456789ABCDEF dan kunci eksternal yang digunakan untuk melakukan enkripsi adalah K = 133457799BBCDFF1. Proses enkripsi menggunakan algoritma DES terhadap *plaintext* Ahmad dilakukan dengan langkah-langkah sebagai berikut.

1. Permutasi Awal

Sebelum dilakukan permutasi awal terlebih dahulu *plaintext* P yang masih berupa bilangan hexadesimal dikonversi ke dalam bilangan biner menggunakan Tabel 4.1. Hasil konversi berupa bilangan biner dengan panjang 64 bit sebagai berikut :

P = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111.

Selanjutnya dilakukan permutasi awal terhadap blok *plaintext* tersebut menggunakan matriks permutasi awal dalam Tabel 4.2 sehingga diperoleh,

IP = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010.

2. Menentukan Kunci Internal

Untuk melakukan sebuah proses enkripsi menggunakan algoritma DES dibutuhkan kunci internal sebanyak 16 buah yang dibangkitkan dari kunci eksternal. Kunci eksternal K = 133457799BBCDFF1 berupa bilangan hexadesimal terlebih dahulu dikonversi ke dalam bilangan biner menggunakan Tabel 4.1. Kunci eksternal yang tersusun dari 64 bit, yaitu

K = 00010011 00110100 01010111 01111001 10011011 10111100 1101111 11110001.

Kunci eksternal K ini menjadi input untuk permutasi menggunakan permutasi kompresi Tabel 4.3 sehingga dihasilkan $K+$ sepanjang 56 bit, yaitu

$K+$ = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111.

Selanjutnya 56 bit dari $K+$ ini dibagi menjadi dua blok yaitu blok kiri (C_0) sepanjang 28 bit dan blok kanan (D_0) sepanjang 28 bit :

C_0 = 1111000 0110011 0010101 0101111

D_0 = 0101010 1011001 1001111 0001111.

Kedua bagian ini digeser ke kiri sepanjang satu atau dua putaran dengan jumlah pergeseran pada setiap putaran menggunakan Tabel 4.4, diperoleh C_i dan D_i untuk $1 \leq i \leq 16$ sebagai berikut :

C_0 = 1111000 0110011 0010101 0101111

D_0 = 0101010 1011001 1001111 0001111

C_1 = 1110000 1100110 0101010 1011111

D_1 = 1010101 0110011 0011110 0011110

C_2 = 1100001 1001100 1010101 0111111

D_2 = 0101010 1100110 0111100 0111101

C_3 = 0000110 0110010 1010101 1111111

D_3 = 0101011 0011001 1110001 1110101

C_4 = 0011001 1001010 1010111 1111100

D_4 = 0101100 1100111 1000111 1010101

C_5 = 1100110 0101010 1011111 1110000

D_5 = 0110011 0011110 0011110 1010101

C_6 = 0011001 0101010 1111111 1000011

D_6 = 1001100 1111000 1111010 1010101

C_7 = 1100101 0101011 1111110 0001100

D_7 = 0110011 1100011 1101010 1010110

C_8 = 0010101 0101111 1111000 0110011

D_8 = 1001111 0001111 0101010 1011001

C_9 = 0101010 1011111 1110000 1100110

D_9 = 0011110 0011110 1010101 0110011

C_{10} = 0101010 1111111 1000011 0011001

D_{10} = 1111000 1111010 1010101 1001100

C_{11} = 0101011 1111110 0001100 1100101

D_{11} = 1100011 1101010 1010110 0110011

C_{12} = 0101111 1111000 0110011 0010101

D_{12} = 0001111 0101010 1011001 1001111

C_{13} = 0111111 1100001 1001100 1010101

D_{13} = 0111101 0101010 1100110 0111100

C_{14} = 1111111 0000110 0110010 1010101

D_{14} = 1110101 0101011 0011001 1110001

C_{15} = 1111100 0011001 1001010 1010111

D_{15} = 1010101 0101100 1100111 1000111

C_{16} = 1111000 0110011 0010101 0101111

D_{16} = 0101010 1011001 1001111 0001111.

Selanjutnya (C_i, D_i) hasil pergeseran tersebut mengalami permutasi kompresi menggunakan matriks PC-2 dalam Tabel 4.5. Dengan permutasi kompresi ini pasangan (C_i,D_i) yang berukuran 56 bit diturunkan menjadi kunci internal K_i (untuk $1 \leq i \leq 16$) yang berukuran 48 bit.

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

K_2 = 011110 011010 111011 011001 110110 111100 100111 100101

K_3 = 010101 011111 110010 001010 010000 101100 111110 011001

K_4 = 011100 101010 110111 010110 110110 110011 010100 011101

K_5 = 011111 001110 110000 000111 111010 110101 001110 101000

K_6 = 011000 111010 010100 111110 010100 000111 101100 101111

K_7 = 111011 001000 010010 110111 111101 100001 100010 111100

K_8 = 111101 111000 101000 111010 110000 010011 101111 111011

K_9 = 111000 001101 101111 101011 111011 011110 011110 000001

K_{10} = 101100 011111 001101 000111 101110 100100 011001 001111

K_{11} = 001000 010101 111111 010011 110111 101101 001110 000110

K_{12} = 011101 010111 000111 110101 100101 000110 011111 101001

K_{13} = 100101 111100 010111 010001 111110 101011 101001 000001

K_{14} = 010111 110100 001110 110111 111100 101110 011100 111010

K_{15} = 101111 111001 000110 001101 001111 010011 111100 001010

K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101.

3. Proses Enkripsi DES

Di dalam proses enkripsi, setelah blok *plaintext* dikenakan permutasi awal diperoleh IP. Selanjutnya IP dibagi menjadi dua bagian yaitu blok kiri (L_0) dan blok kanan (R_0) yang masing-masing panjangnya 32 bit.

L_0 = 1100 1100 0000 0000 1100 1100 1111 1111

R_0 = 1111 0000 1010 1010 1111 0000 1010 1010.

Selanjutnya kedua blok ini diproses kedalam 16 putaran DES dengan satu putaran DES merupakan model jaringan Feistel pada Gambar 4.2 dan secara matematis dinyatakan dalam Persamaan 4.1.

Menggunakan Persamaan 4.1 untuk i = 1 diperoleh,

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

$L_1 = R_0$ = 1111 0000 1010 1010 1111 0000 1010 1010

$R_1 = L_0 \oplus f(R_0,K_1)$.

Untuk mendapatkan nilai dari R_1, terlebih dahulu dicari nilai dari $f(R_0,K_1)$ menggunakan alur dalam Gambar 4.5. Dalam hal ini R_0 merupakan input dari tranformasi f. Selanjutnya menggunakan fungsi ekspansi, R_0 yang panjangnya 32 bit diperluas menjadi blok yang panjangnya 48 bit dengan matriks permutasi ekspansi dalam Tabel 4.6, diperoleh $E(R_0)$ sebagai berikut,

R_0 = 1111 0000 1010 1010 1111 0000 1010 1010

E(R_0) = 011110 100001 010101 010101 011110 100001 010101 010101.

Selanjutnya hasil ekspansi, $E(R_0)$, di-XOR-kan dengan kunci internal K_1 menghasilkan vektor A yang panjangnya 48 bit. Dengan Persamaan 4.2 diperoleh vektor A sebagai berikut,

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

$E(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101

$E(R_0) \oplus K_1 = A$ = 011000 010001 011110 111010 100001 100110 010100 100111.

Vektor A yang telah dikelompokkan menjadi delapan kelompok dan masing-masing kelompok terdiri dari enam bit menjadi input untuk proses subtitusi S-Box.

A_1= 011000, A_2=010001,...., A_7= 010100, A_8=100111.

Dengan Persamaan 4.3 untuk $1 \leq j \leq 8$ diperoleh,

$B_1 = S_1(A_1) = S_1(r, c) = S_1(00, 1100) = S_1(0, 12) = 5 = 0101$

$B_2 = S_2(A_2) = S_2(r, c) = S_2(01, 1000) = S_2(1, 8) = 12 = 1100$

$B_3 = S_3(A_3) = S_3(r, c) = S_3(00, 1111) = S_3(0, 15) = 8 = 1000$

$B_4 = S_4(A_4) = S_4(r, c) = S_4(10, 1101) = S_4(2, 13) = 2 = 0010$

$B_5 = S_5(A_5) = S_5(r, c) = S_5(11, 0000) = S_5(3, 0) = 11 = 1011$

$B_6 = S_6(A_6) = S_6(r, c) = S_6(10, 0011) = S_6(2, 3) = 5 = 0101$

$B_7 = S_7(A_7) = S_7(r, c) = S_7(00, 1010) = S_7(0, 10) = 9 = 1001$

$B_8 = S_8(A_8) = S_8(r, c) = S_8(11, 0011) = S_8(3, 3) = 7 = 0111$.

Sehingga diperoleh output dari proses S-Box sebagai berikut,

$B = B_1B_2B_3B_4B_5B_6B_7B_8$ = 0101 1100 1000 0010 1011 0101 1001 0111.

Selanjutnya vektor B ini menjadi input untuk proses permutasi menggunakan matriks permutasi seperti Tabel 4.7 sehingga dihasilkan $P(B)$, yaitu

$P(B)$ = 0010 0011 0100 1010 1010 1001 1011 1011.

$P(B)$ merupakan output dari fungsi f. Dengan Persamaan 4.4 untuk i = 1, diperoleh $P(B) = f(R_0,K_1)$. Selanjutnya untuk mendapatkan R_1 dilakukan dengan meng-XOR-kan $f(R_0,K_1)$ dengan L_0 sehingga diperoleh R_1 sebagai berikut,

L_0 = 1100 1100 0000 0000 1100 1100 1111 1111

$f(R_0,K_1)$ = 0010 0011 0100 1010 1010 1001 1011 1011

$L_0 \oplus f(R_0,K_1)$ = 1110 1111 0100 1010 0110 0101 0100 0100.

Jadi, $R_1 = L_0 \oplus f(R_0,K_1)$ = 1110 1111 0100 1010 0110 0101 0100 0100.

Proses di atas (untuk i=1) merupakan putaran pertama proses enkripsi menggunakan algoritma DES. Selanjutnya dengan cara yang sama dilakukan putaran ke-2 (i=2) dan seterusnya sampai dengan putaran ke-16 (i=16) sebagaimana ditunjukkan pada Lampiran 3.

Untuk putaran ke-16 dari algoritma DES diperoleh data sebagai berikut,

K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101

$L_{16} = R_{15}$ = 0100 0011 0100 0010 0011 0010 0011 0100

$E(R_{15})$ = 001000 000110 101000 000100 000110 100100 000110 101000

$E(R_{15}) \oplus K_{16} = A$ = 111010 110101 011110 001111 000101 000101 011001 011101

B = 1010 0111 1000 0011 0010 0100 0010 1001

P(B) = $f(R_{15}, K_{16})$ = 1100 1000 1100 0000 0100 1111 1001 1000

R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101.

Menurut persamaan 4.5, output putaran ke-16 dari enkripsi DES data di atas adalah,

$(L_{16}, R_{16}) = (R_{15}, L_{15} \oplus f(R_{15}, K_{16}))$.

Selanjutnya dapat diperoleh pra-*ciphertext* dari proses enkripsi algoritma DES yaitu :

(R_{16}, L_{16}) = 0000 1010 0100 1100 1101 1001 1001 0101 0100 0011 0100 0010 0011 0010 0011 0100.

Proses berikutya melakukan permutasi akhir terhadap pra-*ciphertext* tersebut.

4. Permutasi Terakhir

Setelah dilakukan 16 kali putaran terhadap blok kiri *(L)* dan blok kanan *(R)* dihasilkan pra-*ciphertext* (R_{16}, L_{16})**.** Untuk mendapatkan c*hipertext* yang sebenarnya dilakukan permutasi akhir terhadap (R_{16}, L_{16}) menggunakan *invers* permutasi awal (IP^{-1}) seperti pada Tabel 4.7, diperoleh,

IP^{-1} = 1000 0101 1110 1000 0001 0011 0101 0100 0000 1111 0000 1010 1011 0100 0000 0101.

IP^{-1} merupakan c*hipertext* dari proses enkripsi DES. Jika IP^{-1} dikonversi kedalam bilangan hexadesimal menggunakan Tabel 4.1, diperoleh c*hipertext* C =85E813540F0AB405.

Jadi pesan Ahmad yang berupa *plaintext P* = 0123456789ABCDEF setelah dienkripsi dengan algoritma DES menggunakan kunci eksternal *K*=133457799BBCDFF1, diperoleh *ciphertext C* = 85E813540F0AB405.

5. Dekripsi DES

Proses dekripsi terhadap suatu *ciphertext* merupakan kebalikan dari proses enkripsi. Dalam hal ini DES menggunakan algoritma yang sama untuk melakukan proses enkripsi dan dekripsi. Agar Ibrahim dapat membaca pesan berupa *ciphertext* yang dikirimkan oleh Ahmad, Ibrahim harus mempunyai kunci eksternal yang sama seperti kunci yang digunakan Ahmad untuk melakukan enkripsi. Selanjutnya langkah-langkah yang harus dilakukan Ibrahim sebagai berikut.

Ciphertext yang diterima oleh Ibrahim dari Ahmad terlebih dahulu dikonversi kedalam bilangan biner menggunakan Tabel 4.1, diperoleh,

C = 85E813540F0AB405

C = 1000 0101 1110 1000 0001 0011 0101 0100 0000 1111 0000 1010 1011 0100 0000 0101.

Untuk mendapatkan (R_{16},L_{16}), dilakukan permutasi terhadap C menggunakan matriks *invers* permutasi awal seperti pada Tabel 4.7 sehingga diperoleh,

(R_{16},L_{16}) = 0000 1010 0100 1100 1101 1001 1001 0101 0100 0011 0100 0010 0011 0010 0011 0100.

(R_{16},L_{16}) merupakan blok input awal untuk proses dekripsi. Selanjutnya (R_{16},L_{16}) dibagi menjadi dua bagian yaitu blok kanan (R_{16}) dan blok kiri (L_{16}) yang masing-masing panjangnya 32 bit.

R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101.

L_{16} = 0100 0011 0100 0010 0011 0010 0011 0100.

Selanjutnya kedua blok ini diproses kedalam 16 putaran DES sebagaimana dalam proses enkripsi. Algoritma yang digunakan dalam proses dekripsi sama

seperti algoritma yang digunakan dalam proses enkripsi, hanya saja urutan kunci internal yang digunakan dibalik. Jika pada proses enkripsi urutan kunci internal yang digunakan adalah K_1, K_2,…, K_{16}, maka pada proses dekripsi urutan kunci yang digunakan adalah K_{16}, K_{15},…, K_1.

Untuk setiap putaran dalam proses dekripsi, dihasilkan output yang secara matematis dapat dinyatakan dalam Persamaan 4.6. Menggunakan persamaan tersebut untuk i = 16 diperoleh,

K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101

$R_{15} = L_{16}$ = 0100 0011 0100 0010 0011 0010 0011 0100

$L_{15} = R_{16} \oplus f(L_{16}, K_{16})$.

Untuk mendapatkan nilai dari L_{15}, terlebih dahulu dicari nilai $f(L_{16},K_{16})$ menggunakan alur dalam Gambar 4.5 dengan input R_{i-1} diganti L_i. Dalam hal ini L_{16} merupakan input tranformasi f. Selanjutnya menggunakan fungsi ekspansi, L_{16} yang panjangnya 32 bit akan diperluas menjadi blok yang panjangnya 48 bit dengan matriks permutasi ekspansi dalam Tabel 4.6, diperoleh $E(L_{16})$.

L_{16} = 0100 0011 0100 0010 0011 0010 0011 0100

$E(L_{16})$ = 001000 000110 101000 000100 000110 100100 000110 101000.

Selanjutnya hasil ekspansi, yaitu $E(L_{16})$, di-XOR-kan dengan kunci internal K_{16} menghasilkan vektor A yang panjangnya 48 bit. Dengan Persamaan 4.2 diperoleh vektor A sebagai berikut,

K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101

$E(L_{16})$ = 001000 000110 101000 000100 000110 100100 000110 101000

$E(L_{16}) \oplus K_{16} = A$= 111010 110101 011110 001111 000101 000101 011001 011101.

Vektor A yang telah dikelompokkan menjadi delapan kelompok dan masing-masing kelompok terdiri dari enam bit menjadi input untuk proses subtitusi S-Box.

A_1= 111010, A_2=110101,...., A_7= 011001, A_8=011101

Dengan Persamaan 4.3 untuk $1 \leq j \leq 8$ diperoleh,

$B_1 = S_1(A_1) = S_1(r, c) = S_1(10, 1101) = S_1(2, 13) = 10 = 1010$

$B_2 = S_2(A_2) = S_2(r, c) = S_2(11, 1010) = S_2(3, 10) = 7 = 0111$

$B_3 = S_3(A_3) = S_3(r, c) = S_3(00, 1111) = S_3(0, 15) = 8 = 1000$

$B_4 = S_4 (A_4) = S_4 (r, c) = S_4 (01, 0111) = S_4 (1, 7) = 3 = 0011$

$B_5 = S_5 (A_5) = S_5 (r, c) = S_5 (01, 0010) = S_5 (1, 2) = 2 = 0010$

$B_6 = S_6 (A_6) = S_6 (r, c) = S_6 (01, 0010) = S_6 (1, 2) = 4 = 0100$

$B_7 = S_7 (A_7) = S_7 (r, c) = S_7 (01, 1100) = S_7 (1, 12) = 2 = 0010$

$B_8 = S_8 (A_8) = S_8 (r, c) = S_8 (01, 1110) = S_8 (1, 14) = 9 = 1001.$

Sehingga diperoleh output dari proses S-Box yaitu

$B = B_1B_2B_3B_4B_5B_6B_7B_8$ = 1010 0111 1000 0011 0010 0100 0010 1001.

Selanjutnya vektor *B* ini menjadi input untuk proses permutasi menggunakan matriks permutasi seperti Tabel 4.7 sehingga dihasilkan

P(B) = 1100 1000 1100 0000 0100 1111 1001 1000.

P(B) merupakan output dari fungsi *f*. Dalam hal ini $P(B) = f(L_i, K_i)$ sehingga untuk i=16, $P(B) = f(L_{16}, K_{16})$. Selanjutnya untuk mendapatkan L_{15} dilakukan dengan meng-XOR-kan $f(L_{16}, K_{16})$ dengan R_{16} sehingga diperoleh L_{15} sebagai berikut,

R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101

$f(L_{16}, K_{16})$ = 1100 1000 1100 0000 0100 1111 1001 1000

$R_{16} \oplus f(L_{16}, K_{16})$ = 1100 0010 1000 1100 1001 0110 0000 1101.

Jadi, $L_{15} = R_{16} \oplus f(L_{16}, K_{16})$ = 1100 0010 1000 1100 1001 0110 0000 1101.

Proses di atas (untuk i=16) merupakan putaran pertama proses dekripsi menggunakan algoritma DES. Selanjutnya dengan cara yang sama dilakukan putaran ke-15 (i=15) dan seterusnya sampai dengan putaran ke-1 (i=1).

Untuk putaran ke-1 dari algoritma DES diperoleh data sebagai berikut,

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

$R_0 = L_1$ = 1111 0000 1010 1010 1111 0000 1010 1010

$E(L_1)$ = 011110 100001 010101 010101 011110 100001 010101 010101

$E(L_1) \oplus K_1 = A$ = 011000 010001 011110 111010 100001 100110 010100 100111

B = 0101 1100 1000 0010 1011 0101 1001 0111.

$P(B) = f(L_1, K_1)$ = 0010 0011 0100 1010 1010 1001 1011 1011

L_0 = 1100 1100 0000 0000 1100 1100 1111 1111.

Output putaran ke-i dari proses dekripsi DES adalah,

$(R_{i-1}, L_{i-1}) = (L_i, R_i \oplus f(L_i, K_i))$

sehingga untuk putaran terakhir (i=1), output dari proses dekripsi adalah,

$(R_0,L_0) = (L_1, R_1 \oplus f(L_1, K_1))$.

Dari data di atas dapat diperoleh pra-*plaintext* dari proses dekripsi algoritma DES yaitu :

(L_0, R_0) = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010.

Proses selanjutnya adalah dengan melakukan permutasi akhir terhadap pra-*ciphertext* tersebut menggunakan matriks permutasi awal (*IP)* seperti pada Tabel 4.2 sehingga diperoleh,

IP = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111.

IP merupakan *plaintext* dari proses dekripsi DES. Jika IP dikonversi kedalam bilangan hexadesimal menggunakan Tabel 4.1, akan diperoleh *plaintext* P =0123456789ABCDEF.

Jadi, pesan yang diterima Ibrahim dari Ahmad setelah dilakukan dekripsi dengan algoritma DES menggunakan kunci eksternal K yang sama diperoleh *plaintext* P =0123456789ABCDEF.

4.2. Aplikasi Algoritma DES dalam Sistem Keamanan ATM

Dalam skripsi ini setelah mengetahui bagaimana algoritma DES dapat merahasiakan data, selanjutnya ditunjukkan aplikasi algoritma DES dalam sistem keamanan ATM.

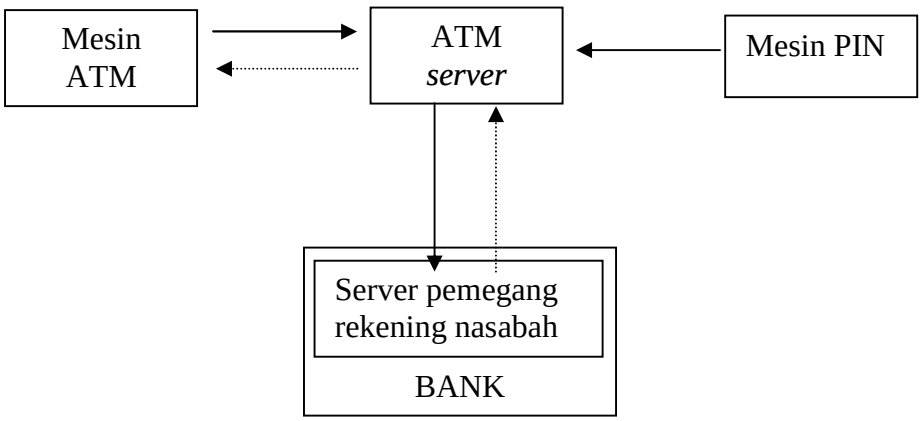
ATM adalah sebuah mesin yang digunakan nasabah bank untuk melakukan transaksi perbankan. ATM umumnya digunakan untuk menarik uang secara tunai *(cash withdrawal)* dari bank. Namun saat ini ATM juga digunakan untuk transfer uang (pemindahbukuan), mengecek saldo, membayar tagihan kartu ponsel, membeli tiket kereta api dan sebagainya. Transaksi melalui ATM memerlukan kartu ATM (disebut juga kartu *magnetic)* yang terdiri dari dua unsur penting, yaitu nomor kartu dan PIN. Masing-masing bank menyusun nomor kartu dari setiap nasabahnya secara unik.

PIN adalah nomor sandi rahasia yang dimiliki oleh setiap pemegang kartu ATM sebagai akses untuk dapat melakukan transaksi menggunakan ATM. Umumnya PIN terdiri dari empat digit yang harus dijaga kerahasiaannya oleh pemilik kartu ATM, sebab orang lain yang mengetahui PIN dapat menggunakan kartu ATM yang dicuri atau hilang untuk melakukan penarikan uang. PIN digunakan untuk memverifikasi kartu yang dimasukkan oleh nasabah di ATM.

Umumnya sistem keamanan pada kartu ATM menggunakan algoritma DES dan RSA. Dalam skripsi ini dipaparkan sistem keamanan pada kartu ATM menggunakan algoritma DES, yaitu untuk menentukan PIN sebuah kartu ATM. Sebelum itu, terlebih dahulu akan jelaskan secara umum tentang prinsip kerja ATM.

4.2.1 Prinsip Kerja ATM

Menurut Mandal [4], sistem keamanan ATM mempunyai tiga komponen utama yaitu mesin ATM *(cash dispenser),* ATM *server* dan mesin PIN sebagaimana dalam Gambar 4.6. Mesin ATM berfungsi untuk membaca nomor kartu dan PIN yang dimasukkan oleh nasabah dan mengirimkan data tersebut ke pusat ATM *server*. ATM server mempunyai basis data yang menyimpan nomor kartu ATM dan PIN secara detail. Mesin PIN digunakan untuk membuktikan keaslian PIN ATM dari nasabah.



Gambar 4.6. Skema prinsip kerja ATM

Penggunaan ATM oleh nasabah dimungkinkan dengan adanya kartu ATM. Setelah kartu ATM dimasukkan kedalam mesin ATM, *magnetic card reader*

(pembaca kartu magnetis) yang berada di dalam mesin akan membaca nomor kartu dari *magnetic stripe*. Fungsi dari *magnetic card reader* ini hanya sebagai pembaca dan penerima data. Karena fungsinya hanya sebagai pembaca dan penerima data, *magnetic card reader* tidak memiliki memori yang bisa menyimpan data nasabah.

Saat mesin berhasil membaca data dalam kartu ATM tersebut, mesin akan meminta data PIN. PIN ini tidak terdapat di dalam kartu ATM melainkan harus dimasukkan oleh nasabah. Kemudian setelah PIN dimasukkan, data PIN yang dimasukkan nasabah beserta data dalam kartu ATM tersebut dienkripsi dan dikirim ke ATM *server*. Enkripsi ini bertujuan agar data yang dikirim tidak bisa dibaca oleh pihak yang tidak berhak. Algoritma enkripsi yang digunakan dalam proses enkripsi PIN ini adalah algoritma DES dengan mode ECB [6].

PIN berikut data dari kartu ATM akan dikirim langsung ke sistem ATM *server* untuk diverifikasi. Selanjutnya dibantu oleh mesin PIN, ATM *server* melakukan verifikasi dengan cara membandingkan PIN yang dimasukkan oleh nasabah dengan PIN yang disimpan dalam basis data pada ATM *server*. Setelah data selesai diverifikasi, data dikirim kembali ke mesin ATM untuk memberi pesan tanggapan apakah transaksi dapat dilanjutkan atau ditolak.

Jika nomor kartu beserta data PIN yang dimasukkan oleh nasabah sama dengan nomor kartu dan PIN yang tersimpan dalam basis data pada ATM *server*, proses dalam mesin ATM dapat dilanjutkan. Jika tidak, mesin ATM akan memberikan kesempatan kepada nasabah untuk memasukkan PIN maksimum tiga kali. Apabila dalam tiga kali memasukkan PIN kedalam ATM tetap salah maka kartu akan diblokir oleh mesin ATM.

Perlu diketahui, mesin ATM tidak menyimpan data maupun PIN yang dimasukkan oleh nasabah. Ini karena prinsip kerja dari mesin ATM hanya menyampaikan pesan *(pass through request)* nasabah ke ATM *server*.

4.2.2. Sistem Keamanan PIN ATM

Keamanan PIN dari sebuah kartu ATM merupakan unsur terpenting dalam seluruh proses sistem keamanan ATM. Hal ini karena PIN merupakan data yang sangat rahasia dari nasabah. Untuk itu dalam semua jaringan ATM, sistem keamanan PIN dirancang sangat kuat terhadap upaya pencurian data dari pihak yang tidak berhak.

Menurut Mandal [4], ada dua kemungkinan yang dilakukan oleh *cryptanalyst* untuk menerka PIN ATM yang digunakan oleh seorang nasabah. Pertama, *cryptanalyst* mendeteksi jaringan ketika mesin ATM sedang mengirimkan data PIN ke ATM *server*. Kedua, *cryptanalyst* mengompromikan ATM *server* dan mesin PIN untuk menguraikan PIN dari nasabah.

Untuk mengantisipasi kemungkinan pertama, PIN yang dikirim oleh mesin ATM dienkripsi terlebih dahulu menggunakan algoritma DES selanjutnya baru dikirim ke ATM *server*. Kunci rahasia yang digunakan dalam proses enkripsi PIN ini sama dengan kunci rahasia yang tersimpan didalam ATM *server*. Untuk mengantisipasi kemungkinan kedua, PIN nasabah disusun menjadi dua bagian dan disimpan dalam dua mesin yang berbeda yaitu dalam ATM *server* dan mesin PIN. Perlu diketahui sistem ini didesain agar nasabah diizinkan untuk mengubah PIN-nya setiap waktu.

Untuk menyusun PIN nasabah kedalam dua bagian akan ditunjukkan sebagai berikut. Misalkan a adalah PIN nasabah. Kemudian a disusun kedalam dua bagian yaitu b dan c sebagaimana Persamaan 4.7.

$$a = b + c \tag{4.7}$$

Pada persamaan 4.7, b adalah bagian variabel dari PIN dan disebut sebagai PIN *offset* yang tersimpan didalam ATM *server*. Jika suatu saat nasabah mengubah data PIN-nya, hanya data PIN *offset* ini yang berubah. Sedangkan c adalah bagian konstanta dari PIN dan disebut sebagai *natural* PIN yang dihasilkan dari mesin PIN setiap waktu. Untuk menghasilkan *natural* PIN dari masing-masing nasabah, konstanta c dapat dibentuk dari nomer kartu menjadi fungsi kriptografi

$c = f$ *(kartu #).*

Ada beberapa algoritma yang digunakan untuk menghasilkan *natural* PIN dari nomor kartu nasabah. Algoritma yang sering digunakan dalam proses ini adalah algoritma DES. Mesin PIN menyimpan kunci DES dalam *Electrically Erasable Programmable Read Only Memory* (EEPROM) [4]. Kunci ini digunakan untuk mengenkripsi nomor kartu dan menghasilkan nilai enkripsi DES. Hasil enkripsi DES ini berupa bilangan hexadesimal. Selanjutnya diambil empat digit dari hasil enkripsi DES ini dan mengganti semua huruf hexadesimal A sampai F berturut-turut dengan angka 0 sampai 5. Empat digit inilah yang disebut dengan *natural* PIN. Untuk mendapatkan PIN yang digunakan oleh nasabah (*a*), tambahkan PIN *offset (b)* yang tersimpan didalam ATM *server* dengan *natural* PIN *(c)* yang dihasilkan oleh mesin PIN.

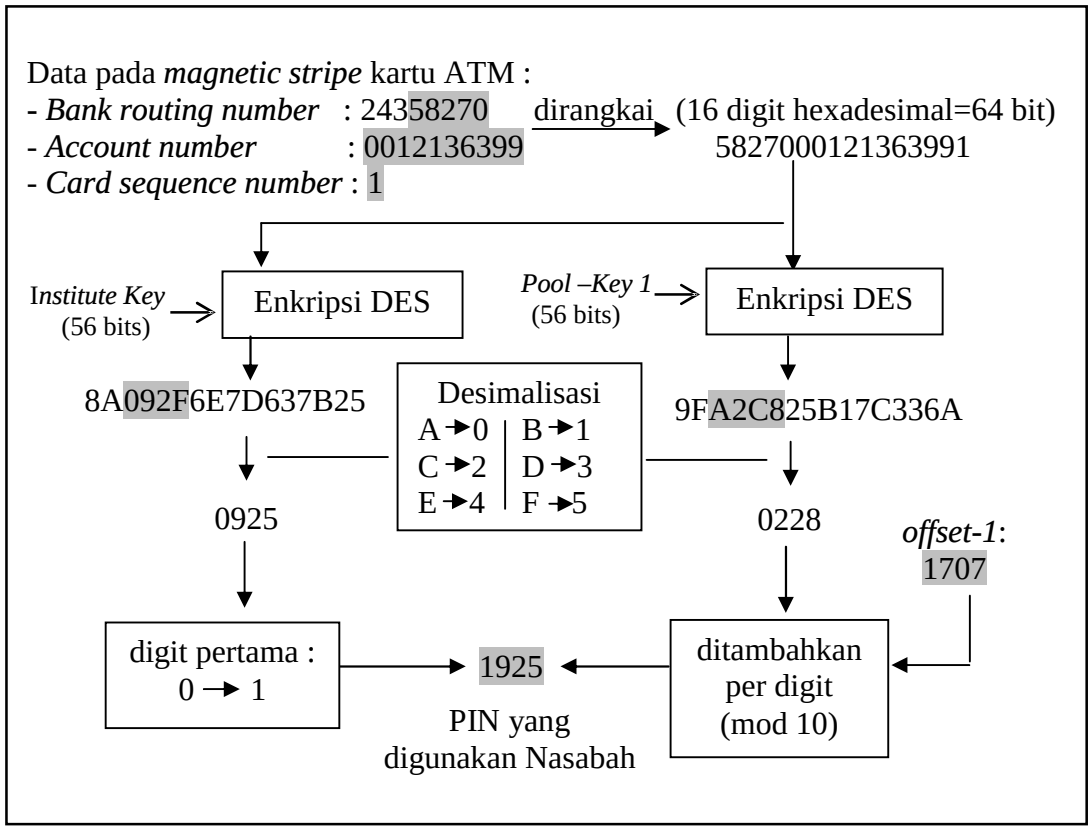
4.2.3. Sistem Keamanan PIN ATM pada ATM Eurocheque

Setelah mengetahui sistem keamanan PIN ATM secara umum, selanjutnya ditunjukkan sistem keamanan PIN ATM pada ATM Eurocheque yaitu metode untuk menentukan PIN yang digunakan oleh nasabah dalam ATM Eurocheque. Dalam pembahasan ini penulis mengambil referensi dari jurnal *Probability Theory for Pickpockets- ec-PIN Guessing* yang ditulis oleh Markus G. Kuhn [3].

Menurut Kuhn [3], PIN ATM yang digunakan oleh nasabah pengguna kartu ATM Eurocheque ditentukan oleh bank bersangkutan. Bank menghitung dan menentukan PIN untuk setiap nasabahnya sebagaimana ditunjukkan dalam Gambar 4.7.

Gambar tersebut menunjukkan 16 digit nomor kartu yang berupa bilangan hexadesimal dirangkai dari lima digit *bank routing number* (nomor urut bank), sepuluh digit *account number* (nomor rekening), dan satu digit *card sequence number* (nomor urutan kartu) yang terdapat dalam *magnetic stripe* pada kartu ATM Eurocheque. Kemudian 16 digit bilangan hexadesimal tersebut ditransformasi kedalam 64 bit bilangan biner dengan masing-masing grup tersusun dari 4 bit yang selanjutnya menjadi *plaintext* dari kartu ATM Eurocheque. Selanjutnya *plaintext* ini dienskripsi menggunakan algoritma DES dengan kunci rahasia sepanjang 56 bit yang disebut dengan *institute key* (K_I). Hasil dari enkripsi

ini adalah *ciphertext* berupa bilangan biner dengan panjang 64 bit yang selanjutnya dikonversi kedalam 16 digit bilangan hexadesimal.



Gambar 4.7 Diagram alir penentuan PIN untuk ATM Eurocheque

Tahap selanjutnya Kuhn [3] menjelaskan *chipertext* yang berupa 16 digit bilangan hexadesimal tersebut diambil empat digit, yaitu digit ke tiga sampai dengan digit ke enam dan mengganti semua huruf hexadesimal A sampai F berturut-turut dengan angka 0 sampai 5. Jika empat digit tersebut yang pertama adalah 0, maka harus di ganti dengan 1. Jaringan ATM yang dimiliki oleh bank penerbit kartu dikenal dengan K_I. Jaringan ini menentukan PIN dengan cara yang sama dan membandinganya dengan PIN yang dimasukkan oleh nasabah.

Jaringan ATM dari bank-bank lain menggunakan kunci enkripsi DES yang berbeda. Kunci yang digunakan untuk proses enkripsi DES ini adalah *pool key* (K_P) yang panjangnya 56 bit. Kunci ini menghasilkan enkripsi DES yang berbeda dengan hasil enkripsi DES menggunakan *institute key*. Selanjutnya, menggunakan *pool key* ini dihasilkan 16 digit hexadesimal yang kemudian diambil empat digit, yaitu digit ke tiga sampai dengan digit ke enam dan

mengganti semua huruf hexadesimal A sampai F berturut-turut dengan angka 0 sampai 5. Dalam hal ini, angka 0 dalam digit pertama dari keempat digit tersebut tidak diganti dengan 1. Keempat digit yang diambil itu disebut sebagai *natural* PIN.

Menurut Kuhn [3], *magnetic stripe* dari sebuah kartu ATM terdiri dari tiga buah PIN *offset* yang masing-masing terdiri dari empat digit desimal. Selanjutnya untuk mendapatkan PIN yang digunakan oleh nasabah, *natural* PIN yang telah diperoleh tersebut ditambahkan masing-masing digitnya dengan PIN *offset*. Jika kunci yang digunakan dalam enkripsi DES adalah *pool key 1 (K_{P1})*, PIN *offset* yang digunakan adalah PIN *offset 1 (O_1)*. Penjumlahan *natural* PIN dengan PIN *offset* dalam proses ini dilakukan dalam operasi modulo 10.

Pool key 1 telah dikenal oleh semua bank di Eropa dan dapat dikompromikan lebih mudah. Untuk itu terdapat dua cadangan *pool key* yaitu *pool key* 2 (K_{P2}) dan *pool key* 3 (K_{P3}) yang tersimpan didalam mesin PIN. Selain itu *magnetic stripe* dalam kartu ATM juga menyimpan dua PIN *offset* yang saling berhubungan yaitu PIN *offset* 2 *(O_2)* dan PIN *offset* 3 *(O_3)*. Ketika seorang nasabah bermaksud mengubah nomor PIN-nya haruslah mengonfirmasikan penggantian K_{P1} sehari sebelumnya. Hal ini bertujuan untuk mengaktifkan K_{P2} dan menulis ulang O_1 dalam kartu ATM pada kunjungan berikutnya.

Walaupun sistem keamanan PIN telah dirancang dengan keamanan berlapis, ternyata masih membuka peluang *cryptanalys* untuk mencoba melakukan *cryptanalysis* terhadap PIN sebuah kartu ATM.

4.3. *Cryptanalysis* Sistem Keamanan ATM Eurocheque

Dalam skripsi ini setelah mengetahui sistem keamanan yang digunakan oleh ATM Eurocheque dalam menentukan PIN yang digunakan oleh nasabahnya, selanjutnya ditunjukkan bagaimana melakukan proses *cryptanalysis* pada sistem keamanan ATM Eurocheque menggunakan teori probabilitas. Dalam pembahasan ini penulis mengambil referensi dari jurnal *Probability Theory for Pickpockets-ec-PIN Guessing* yang ditulis oleh Markus G. Kuhn [3].

Cryptnanalysis yang dilakukan dalam skripsi ini adalah untuk menentukan empat digit PIN ATM Eurocheque yang digunakan nasabah dengan mencari nilai probabilitas maksimal (terbesar) dari semua nilai probabilitas yang ada pada setiap digit PIN. Dalam pembahasan ini dibatasi untuk menentukan setiap digit PIN yang digunakan oleh nasabah, harus diketahui atau ditentukan terlebih dahulu PIN *offset*-nya.

Tujuan dari pembahasan ini adalah untuk menentukan empat digit PIN $\hat{P} = (\hat{P}_1, \hat{P}_2 \hat{P}_3, \hat{P}_4)$ yang digunakan oleh nasabah pada kartu ATM Eurocheque. Menurut Kuhn [3], *magnetic stripe* dari sebuah kartu ATM terdiri dari tiga buah PIN *offset* yang masing-masing terdiri dari empat digit desimal. Misalkan PIN *offset* dalam sebuah kartu ATM dituliskan sebagai O_i dengan $1 \le i \le 3$, yang dituliskan sebagai berikut.

$O_1 = (O_{1,1}, O_{1,2}, O_{1,3}, O_{1,4})$

$O_2 = (O_{2,1}, O_{2,2}, O_{2,3}, O_{2,4})$

$O_3 = (O_{3,1}, O_{3,2}, O_{3,3}, O_{3,4})$.

Dari Gambar 4.7 dapat diketahui empat digit PIN yang digunakan oleh nasabah dapat dihitung menggunakan algoritma DES dengan kunci rahasia *institute key* dan *pool key*. Untuk mendapatkan nilai $\hat{P}$, terlebih dahulu dicari probabilitas dari $\tilde{P}_j$ untuk $1 \le j \le 4$. $\tilde{P}_j$ merupakan variabel random yang menunjukkan digit ke- j dari PIN yang dicari dalam sebuah kartu ATM yang dihitung menggunakan *institute key*. Selanjutnya, dari proses penentuan PIN menggunakan *pool key* dicari probabilitas bersyarat dari $\tilde{O}_{i,j}$ untuk semua $1 \le i \le 3$ dan $1 \le j \le 4$ jika diberikan $\tilde{P}_j$. $\tilde{O}_{i,j}$ merupakan variabel random yang menunjukkan digit ke-j dalam PIN *offset i* jika diketahui nilai dari $\tilde{P}_j$.

Perlu diketahui hasil dari enkripsi DES berupa bilangan kexadesimal yang mempunyai anggota : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E dan F. Dalam hal ini dilakukan proses desimalisasi terhadap hasil enkripsi DES dengan mengganti

semua huruf hexadesimal A sampai F berturut-turut dengan angka 0 sampai 5. Diasumsikan empat digit PIN yang diambil dari hasil enkripsi DES saling independen dan 16 digit hexadesimal dari hasil enkripsi DES berdistribusi seragam [3], sehingga semua digit hexadesimal mempunyai kemungkinan yang sama untuk muncul.

Selanjutnya untuk mencari probabilitas $\tilde{P}_j$, terlebih dahulu dihitung nilai ($\tilde{P}_j = k$) untuk setiap $k \in (0,1,...,8,9)$, yang dapat dihitung dalam Tabel 4.9.

Tabel 4.9 Nilai dari $\tilde{P}_j$ untuk setiap k

($\tilde{P}_j = k$), untuk $1 \le j \le 4$ dan $k \in (0,1,...,8,9)$			
$\tilde{P}_1$	$\tilde{P}_2$	$\tilde{P}_3$	$\tilde{P}_4$
($\tilde{P}_1 = 0$) = { }	($\tilde{P}_2 = 0$) = {0,A}	($\tilde{P}_3 = 0$) = {0,A}	($\tilde{P}_4 = 0$) = {0,A}
($\tilde{P}_1 = 1$) = {0,1,A,B}	($\tilde{P}_2 = 1$) = {1,B}	($\tilde{P}_3 = 1$) = {1,B}	($\tilde{P}_4 = 1$) = {1,B}
($\tilde{P}_1 = 2$) = {2,C}	($\tilde{P}_2 = 2$) = {2,C}	($\tilde{P}_3 = 2$) = {2,C}	($\tilde{P}_4 = 2$) = {2,C}
($\tilde{P}_1 = 3$) = {3,D}	($\tilde{P}_2 = 3$) = {3,D}	($\tilde{P}_3 = 3$) = {3,D}	($\tilde{P}_4 = 3$) = {3,D}
($\tilde{P}_1 = 4$) = {4,E}	($\tilde{P}_2 = 4$) = {4,E}	($\tilde{P}_3 = 4$) = {4,E}	($\tilde{P}_4 = 4$) = {4,E}
($\tilde{P}_1 = 5$) = {5,F}	($\tilde{P}_2 = 5$) = {5,F}	($\tilde{P}_3 = 5$) = {5,F}	($\tilde{P}_4 = 5$) = {5,F}
($\tilde{P}_1 = 6$) = {6}	($\tilde{P}_2 = 6$) = {6}	($\tilde{P}_3 = 6$) = {6}	($\tilde{P}_4 = 6$) = {6}
($\tilde{P}_1 = 7$) = {7}	($\tilde{P}_2 = 7$) = {7}	($\tilde{P}_3 = 7$) = {7}	($\tilde{P}_4 = 7$) = {7}
($\tilde{P}_1 = 8$) = {8}	($\tilde{P}_2 = 8$) = {8}	($\tilde{P}_3 = 8$) = {8}	($\tilde{P}_4 = 8$) = {8}
($\tilde{P}_1 = 9$) = {9}	($\tilde{P}_2 = 9$) = {9}	($\tilde{P}_3 = 9$) = {9}	($\tilde{P}_4 = 9$) = {9}

Dari Tabel 4.9 dapat disederhanakan, probabilitas dari ($\tilde{P}_j = k$) untuk $1 \le j \le 4$ dan $k \in \{0,1,...,8,9\}$, dapat ditulis dalam Persamaan 4.8.

$$p(\tilde{P}_j = k) = \begin{cases} 0/16, & jika\ j = 1\ dan\ k = 0 \\ 4/16, & jika\ j = 1\ dan\ k = 1 \\ 2/16, & jika\ j > 1\ dan\ k \in \{0,1\} \\ 2/16, & \forall_j\ dan\ k \in \{2,...,5\} \\ 1/16, & \forall_j\ dan\ k \in \{6,...,9\} \end{cases} \quad (4.8)$$

Selanjutnya probabilitas bersyarat dari $\tilde{O}_{i,j}$ untuk semua $1 \le i \le 3$ dan $1 \le j \le 4$ jika diberikan nilai dari $\tilde{P}_j$ adalah

$$p(\tilde{O}_{i,j} = k \mid \tilde{P}_j = l) = \begin{cases} 2/16, & jika\ (l-k) \bmod 10 \in \{0,...,5\} \\ 1/16, & jika\ (l-k) \bmod 10 \in \{6,...,9\} \end{cases} \quad (4.9)$$

PIN $\hat{P}$ dalam pembahasan ini adalah P untuk probabilitas bersyarat $p(\tilde{P} = P \mid \forall_i : \tilde{O}_i = O_i)$ maksimal. Karena semua digit dari PIN diasumsikan saling independen, dapat ditentukan PIN $\hat{P}$ digit ke-j yang ditulis dengan $\hat{P}_j$ sebagai P_j yang memaksimalkan nilai $p(\tilde{P}_j = P_j \mid \forall i : \tilde{O}_{i,j} = O_{i,j})$. Untuk memperoleh $\hat{P}_j$ digunakan probabilitas bersyarat dalam Persamaan 4.10.

$$p(\tilde{P}_j = P_j \mid \forall_i : \tilde{O}_{i,j} = O_{i,j}) = \frac{p(\tilde{P}_j = P_j \wedge \forall_i : \tilde{O}_{i,j} = O_{i,j})}{p(\forall_i : \tilde{O}_{i,j} = O_{i,j})} \quad (4.10)$$

Selanjutnya dengan Teorema 2.4, Persamaan 4.10 dapat ditulis menjadi

$$p(\tilde{P}_j = P_j \mid \forall_i : \tilde{O}_{i,j} = O_{i,j}) = \frac{p(\forall_i : \tilde{O}_{i,j} = O_{i,j} \mid \tilde{P}_j = P_j).p(\tilde{P}_j = P_j)}{p(\forall_i : \tilde{O}_{i,j} = O_{i,j})}. \quad (4.11)$$

Menggunakan Teorema 2.5, Persamaan 4.11 dapat ditulis kembali menjadi Persamaan 4.12 yang dikenal dengan Teorema Bayes.

$$p(\tilde{P}_j = P_j \mid \forall_i : \tilde{O}_{i,j} = O_{i,j}) = \frac{p(\forall_i : \tilde{O}_{i,j} = O_{i,j} \mid \tilde{P}_j = P_j).p(\tilde{P}_j = P_j)}{\sum_{k=0}^{9} p(\forall_i : \tilde{O}_{i,j} = O_{i,j} \mid \tilde{P}_j = k).p(\tilde{P}_j = k)} \quad (4.12)$$

Teorema Bayes dalam Persamaan 4.12 di atas digunakan untuk menghitung masing-masing digit PIN $\hat{P}_j$ untuk $1 \le j \le 4$. Karena diawal telah diasumsikan empat digit PIN hasil enkripsi DES menggunakan ketiga *pool key* saling independen, untuk menghitung probabilitas bersyarat

$p(\tilde{P}_j = P_j \mid \forall i : \tilde{O}_{i,j} = O_{i,j})$ dari semua digit dalam ketiga PIN *offset* dapat dihitung menggunakan prinsip perkalian dalam teori probabilitas. Diperoleh,

$$p(\tilde{P}_j = P_j \mid \forall_i : \tilde{O}_{i,j} = O_{i,j}) = \frac{\prod_{i=1}^{3} p(\tilde{O}_{i,j} = O_{i,j} \mid \tilde{P}_j = P_j).p(\tilde{P}_j = P_j)}{\sum_{k=0}^{9}\prod_{i=1}^{3} p(\tilde{O}_{i,j} = O_{i,j} \mid \tilde{P}_j = k).p(\tilde{P}_j = k)} . \quad (4.13a)$$

Dengan Persamaan 4.13a dapat dihitung probabilitas bersyarat $p(\tilde{P}_j = P_j \mid \forall i : \tilde{O}_{i,j} = O_{i,j})$ untuk semua $P_j \in \{0,..,9\}$. Dalam pembahasan ini untuk mendapatkan nilai $\hat{P}_j$, harus diketahui terlebih dahulu nilai dari $O_{1,j}$,$O_{2,j}$ dan $O_{3,j}$. Selanjutnya dicari empat digit PIN $\hat{P}_j$ untuk $1 \le j \le 4$. Digit pertama dari PIN $\hat{P}_j$ adalah $\hat{P}_1$,diperoleh dengan mencari nilai $P_j \in \{0,..,9\}$ yang membuat nilai probabilitas bersyarat $p(\tilde{P}_1 = P_1 \mid \forall i : \tilde{O}_{i,1} = O_{i,1})$ maksimal untuk j=1. Digit kedua dari PIN $\hat{P}_j$ adalah $\hat{P}_2$,diperoleh dengan mencari nilai $P_j \in \{0,..,9\}$ yang membuat nilai probabilitas bersyarat $p(\tilde{P}_2 = P_2 \mid \forall i : \tilde{O}_{i,2} = O_{i,2})$ maksimal untuk j=2. Dengan cara yang sama dapat ditentukan nilai dari $\hat{P}_3$ dan $\hat{P}_4$.

4.4. Implementasi Kasus

Untuk memperjelas bagaimana proses *cryptanalysis* sistem keamanan ATM Eurocheque menggunakan teori probabilitas, berikut diberikan contoh aplikasi beserta perhitunganya.

Misalkan diketahui nilai dari PIN *offset* sebuah kartu ATM sebagai berikut :

O_1 = ($O_{1,1}$= 2, $O_{1,2}$ = 4, $O_{1,3}$ = 0, $O_{1,4}$ = 5)

O_2 = ($O_{2,1}$= 1, $O_{2,2}$ = 9, $O_{2,3}$ = 8, $O_{2,4}$= 0)

O_3 = ($O_{3,1}$= 5, $O_{3,2}$ = 4, $O_{3,3}$ = 3, $O_{3,4}$ = 2).

Untuk mencari empat digit PIN $\hat{P} = (\hat{P}_1, \hat{P}_2 \hat{P}_3, \hat{P}_4)$ yang digunakan oleh nasabah, dihitung menggunakan Persamaan 4.13a untuk $P_j \in \{0,..,9\}$.

$$p(\tilde{P}_j = P_j \,|\, \forall_i : \tilde{O}_{i,j} = O_{i,j}) = \frac{\prod_{i=1}^{3} p(\tilde{O}_{i,j} = O_{i,j} \,|\, \tilde{P}_j = P_j).p(\tilde{P}_j = P_j)}{\sum_{k=0}^{9}\prod_{i=1}^{3} p(\tilde{O}_{i,j} = O_{i,j} \,|\, \tilde{P}_j = k).p(\tilde{P}_j = k)}$$

$$= \frac{\begin{array}{l}\left[p(\tilde{O}_{1,j} = O_{1,j} \,|\, \tilde{P}_j = P_j).p(\tilde{P}_j = P_j) \right]\left[p(\tilde{O}_{2,j} = O_{2,j} \,|\, \tilde{P}_j = P_j).p(\tilde{P}_j = P_j) \right] \\ \left[p(\tilde{O}_{3,j} = O_{3,j} \,|\, \tilde{P}_j = P_j).p(\tilde{P}_j = P_j) \right]\end{array}}{\sum_{k=0}^{9}\left\{\begin{array}{l}\left[p(\tilde{O}_{1,j} = O_{1,j} \,|\, \tilde{P}_j = k).p(\tilde{P}_j = k) \right]\left[p(\tilde{O}_{2,j} = O_{2,j} \,|\, \tilde{P}_j = k).p(\tilde{P}_j = k) \right] \\ \left[p(\tilde{O}_{3,j} = O_{3,j} \,|\, \tilde{P}_j = k).p(\tilde{P}_j = k) \right]\end{array}\right\}} \quad (4.13b)$$

1. Untuk j=1

Untuk j=1, Persamaan 4.13b menjadi

$$p(\tilde{P}_1 = P_1 \,|\, \forall i : \tilde{O}_{i,1} = O_{i,1}) =$$

$$\frac{\begin{array}{l}\left[p(\tilde{O}_{1,1} = 2 \,|\, \tilde{P}_1 = P_1).p(\tilde{P}_1 = P_1) \right]\left[p(\tilde{O}_{2,1} = 1 \,|\, \tilde{P}_1 = P_1).p(\tilde{P}_1 = P_1) \right] \\ \left[p(\tilde{O}_{3,1} = 5 \,|\, \tilde{P}_1 = P_1).p(\tilde{P}_1 = P_1) \right]\end{array}}{\sum_{k=0}^{9}\left\{\begin{array}{l}\left[p(\tilde{O}_{1,1} = 2 \,|\, \tilde{P}_1 = k).p(\tilde{P}_1 = k) \right]\left[p(\tilde{O}_{2,1} = 1 \,|\, \tilde{P}_1 = k).p(\tilde{P}_1 = k) \right] \\ \left[p(\tilde{O}_{3,1} = 5 \,|\, \tilde{P}_1 = k).p(\tilde{P}_1 = k) \right]\end{array}\right\}}. \quad (4.14)$$

Selanjutnya dengan Persamaan 4.14 dihitung $p(\tilde{P}_1 = P_1 \,|\, \forall i : \tilde{O}_{i,1} = O_{i,1})$ untuk $P_1 \in \{0,..,9\}$.

a. Untuk P_1=0,

$$p(\tilde{P}_1 = 0 \mid \forall i : \tilde{O}_{i,1} = O_{i,1}) = \frac{\left(\frac{1}{16}\frac{0}{16}\right)\left(\frac{1}{16}\frac{0}{16}\right)\left(\frac{2}{16}\frac{0}{16}\right)}{\left[\left(\frac{1}{16}\frac{0}{16}\right)\left(\frac{1}{16}\frac{0}{16}\right)\left(\frac{2}{16}\frac{0}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{4}{16}\right)\left(\frac{2}{16}\frac{4}{16}\right)\left(\frac{1}{16}\frac{4}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]}$$

$$= \frac{\frac{0}{16^6}}{\frac{0}{16^6}+\frac{128}{16^6}+\frac{32}{16^6}+\frac{32}{16^6}+\frac{32}{16^6}+\frac{64}{16^6}+\frac{8}{16^6}+\frac{4}{16^6}+\frac{2}{16^6}+\frac{2}{16^6}} = \frac{\frac{0}{16^6}}{\frac{304}{16^6}} = \frac{0}{304} = 0$$

b. Untuk P_1=1,

$$p(\tilde{P}_1 = 1 \mid \forall i : \tilde{O}_{i,1} = O_{i,1}) = \frac{\left(\frac{1}{16}\frac{4}{16}\right)\left(\frac{2}{16}\frac{4}{16}\right)\left(\frac{1}{16}\frac{4}{16}\right)}{\frac{304}{16^6}} = \frac{\frac{128}{16^6}}{\frac{304}{16^6}} = \frac{128}{304} = 0.4211$$

c. Untuk P_1=2,

$$p(\tilde{P}_1 = 2 \mid \forall i : \tilde{O}_{i,1} = O_{i,1}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\frac{304}{16^6}} = \frac{\frac{32}{16^6}}{\frac{304}{16^6}} = \frac{32}{304} = 0.1053$$

d. Untuk P_1=3,

$$p(\tilde{P}_1 = 3 \mid \forall i : \tilde{O}_{i,1} = O_{i,1}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\frac{304}{16^6}} = \frac{\frac{32}{16^6}}{\frac{304}{16^6}} = \frac{32}{304} = 0.1053$$

e. Untuk P_1=4,

$$p(\tilde{P}_1 = 4 \mid \forall i : \tilde{O}_{i,1} = O_{i,1}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\frac{304}{16^6}} = \frac{\frac{32}{16^6}}{\frac{304}{16^6}} = \frac{32}{304} = 0.1053$$

f. Untuk P_1=5,

$$p(\tilde{P}_1 = 5 \mid \forall i : \tilde{O}_{i,1} = O_{i,1}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)}{\frac{304}{16^6}} = \frac{\frac{64}{16^6}}{\frac{304}{16^6}} = \frac{64}{304} = 0.2105$$

g. Untuk P_1=6,

$$p(\tilde{P}_1 = 6 \mid \forall i : \tilde{O}_{i,1} = O_{i,1}) = \frac{\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{304}{16^6}} = \frac{\frac{8}{16^6}}{\frac{304}{16^6}} = \frac{8}{304} = 0.0263$$

h. Untuk P_1=7,

$$p(\tilde{P}_1 = 7 \mid \forall i : \tilde{O}_{i,1} = O_{i,1}) = \frac{\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{304}{16^6}} = \frac{\frac{32}{16^6}}{\frac{304}{16^6}} = \frac{4}{304} = 0.0132$$

i. Untuk P_1=8,

$$p(\tilde{P}_1 = 8 \mid \forall i : \tilde{O}_{i,1} = O_{i,1}) = \frac{\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{304}{16^6}} = \frac{\frac{32}{16^6}}{\frac{304}{16^6}} = \frac{2}{304} = 0.0066$$

j. Untuk P_1=9,

$$p(\tilde{P}_1 = 9 \mid \forall i : \tilde{O}_{i,1} = O_{i,1}) = \frac{\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{304}{16^6}} = \frac{\frac{32}{16^6}}{\frac{304}{16^6}} = \frac{2}{304} = 0.0066$$

2. Untuk j=2

Untuk j=2, Persamaan 4.13b menjadi

$$p(\tilde{P}_2 = P_2 \mid \forall i : \tilde{O}_{i,2} = O_{i,2}) =$$

$$\frac{\left[p(\tilde{O}_{1,2}=4\left|\tilde{P}_2=P_2\right.).p(\tilde{P}_2=P_2)\right]\left[p(\tilde{O}_{2,2}=9\left|\tilde{P}_2=P_2\right.).p(\tilde{P}_2=P_2)\right]\left[p(\tilde{O}_{3,2}=4\left|\tilde{P}_2=P_2\right.).p(\tilde{P}_2=P_2)\right]}{\sum_{k=0}^{9}\left\{\left[p(\tilde{O}_{1,2}=4\left|\tilde{P}_2=k\right.).p(\tilde{P}_2=k)\right]\left[p(\tilde{O}_{2,2}=9\left|\tilde{P}_2=k\right.).p(\tilde{P}_2=k)\right]\left[p(\tilde{O}_{3,2}=4\left|\tilde{P}_2=k\right.).p(\tilde{P}_2=k)\right]\right\}}. \quad (4.15)$$

Selanjutnya dengan Persamaan 4.15 dihitung $p(\tilde{P}_2=P_2\left|\forall i:\tilde{O}_{i,2}=O_{i,2}\right.)$ untuk $P_2 \in \{0,..,9\}$.

a. Untuk P_2=0,

$$p(\tilde{P}_2=0\left|\forall i:\tilde{O}_{i,2}=O_{i,2}\right.)=\frac{\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\begin{array}{l}\left[\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\\ \left[\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\\ \left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\right]+\\ \left[\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\\ \left[\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]\end{array}}$$

$$=\frac{\frac{16}{16^6}}{\frac{16}{16^6}+\frac{16}{16^6}+\frac{16}{16^6}+\frac{16}{16^6}+\frac{64}{16^6}+\frac{32}{16^6}+\frac{4}{16^6}+\frac{4}{16^6}+\frac{4}{16^6}+\frac{8}{16^6}}=\frac{\frac{16}{16^6}}{\frac{180}{16^6}}=\frac{16}{180}=0.089$$

b. Untuk P_2=1,

$$p(\tilde{P}_2=1\left|\forall i:\tilde{O}_{i,2}=O_{i,2}\right.)=\frac{\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\frac{180}{16^6}}=\frac{\frac{16}{16^6}}{\frac{180}{16^6}}=\frac{16}{180}=0.089$$

c. Untuk P_2=2,

$$p(\tilde{P}_2 = 2 \mid \forall i : \tilde{O}_{i,2} = O_{i,2}) = \frac{\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\frac{180}{16^6}} = \frac{\frac{16}{16^6}}{\frac{180}{16^6}} = \frac{16}{180} = 0.089$$

d. Untuk P_2=3,

$$p(\tilde{P}_2 = 3 \mid \forall i : \tilde{O}_{i,2} = O_{i,2}) = \frac{\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\frac{180}{16^6}} = \frac{\frac{16}{16^6}}{\frac{180}{16^6}} = \frac{16}{180} = 0.089$$

e. Untuk P_2=4,

$$p(\tilde{P}_2 = 4 \mid \forall i : \tilde{O}_{i,2} = O_{i,2}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)}{\frac{180}{16^6}} = \frac{\frac{64}{16^6}}{\frac{180}{16^6}} = \frac{64}{180} = 0.3556$$

f. Untuk P_2=5,

$$p(\tilde{P}_2 = 5 \mid \forall i : \tilde{O}_{i,2} = O_{i,2}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)}{\frac{180}{16^6}} = \frac{\frac{32}{16^6}}{\frac{180}{16^6}} = \frac{32}{180} = 0.1778$$

g. Untuk P_2=6,

$$p(\tilde{P}_2 = 6 \mid \forall i : \tilde{O}_{i,2} = O_{i,2}) = \frac{\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{180}{16^6}} = \frac{\frac{4}{16^6}}{\frac{180}{16^6}} = \frac{4}{180} = 0.0222$$

h. Untuk P_2=7,

$$p(\tilde{P}_2 = 7 \mid \forall i : \tilde{O}_{i,2} = O_{i,2}) = \frac{\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{180}{16^6}} = \frac{\frac{4}{16^6}}{\frac{180}{16^6}} = \frac{4}{180} = 0.0222$$

i. Untuk P_2=8,

$$p(\tilde{P}_2 = 8 \mid \forall i : \tilde{O}_{i,2} = O_{i,2}) = \frac{\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{180}{16^6}} = \frac{\frac{4}{16^6}}{\frac{180}{16^6}} = \frac{4}{180} = 0.0222$$

j. Untuk P_2=9,

$$p(\tilde{P}_2 = 9 \mid \forall i : \tilde{O}_{i,2} = O_{i,2}) = \frac{\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{180}{16^6}} = \frac{\frac{8}{16^6}}{\frac{180}{16^6}} = \frac{8}{180} = 0.0444$$

3. Untuk j=3

Untuk j=3, Persamaan 4.13b menjadi

$$p(\tilde{P}_3 = P_3 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) =$$

$$\frac{\left[p(\tilde{O}_{1,3} = 0 \mid \tilde{P}_3 = P_3).p(\tilde{P}_3 = P_3)\right]\left[p(\tilde{O}_{2,3} = 8 \mid \tilde{P}_3 = P_3).p(\tilde{P}_3 = P_3)\right]\left[p(\tilde{O}_{3,3} = 3 \mid \tilde{P}_3 = P_3).p(\tilde{P}_3 = P_3)\right]}{\sum_{k=0}^{9}\left\{\left[p(\tilde{O}_{1,3} = 0 \mid \tilde{P}_3 = k).p(\tilde{P}_3 = k)\right]\left[p(\tilde{O}_{2,3} = 8 \mid \tilde{P}_3 = k).p(\tilde{P}_3 = k)\right]\left[p(\tilde{O}_{3,3} = 3 \mid \tilde{P}_3 = k).p(\tilde{P}_3 = k)\right]\right\}}. \quad (4.16)$$

Selanjutnya dengan persamaan 4.16 dihitung $p(\tilde{P}_3 = P_3 \mid \forall i : \tilde{O}_{i,3} = O_{i,3})$ untuk $P_3 \in \{0,..,9\}$.

a. Untuk P_3=0,

$$p(\tilde{P}_3 = 0 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\right]$$

$$= \frac{\frac{32}{16^6}}{\frac{32}{16^6}+\frac{32}{16^6}+\frac{32}{16^6}+\frac{64}{16^6}+\frac{32}{16^6}+\frac{32}{16^6}+\frac{2}{16^6}+\frac{2}{16^6}+\frac{4}{16^6}+\frac{2}{16^6}} = \frac{\frac{32}{16^6}}{\frac{234}{16^6}} = \frac{32}{234} = 0.1367$$

b. Untuk P_3=1,

$$p(\tilde{P}_3 = 1 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\frac{234}{16^6}} = \frac{\frac{32}{16^6}}{\frac{234}{16^6}} = \frac{32}{234} = 0.1367$$

c. Untuk P_3=2,

$$p(\tilde{P}_3 = 2 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\frac{234}{16^6}} = \frac{\frac{32}{16^6}}{\frac{234}{16^6}} = \frac{32}{234} = 0.1367$$

d. Untuk P_3=3,

$$p(\tilde{P}_3 = 3 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)}{\frac{234}{16^6}} = \frac{\frac{64}{16^6}}{\frac{234}{16^6}} = \frac{64}{234} = 0.2735$$

e. Untuk P_3=4,

$$p(\tilde{P}_3 = 4 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)}{\frac{234}{16^6}} = \frac{\frac{32}{16^6}}{\frac{234}{16^6}} = \frac{32}{234} = 0.1367$$

f. Untuk P_3=5,

$$p(\tilde{P}_3 = 5 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\frac{234}{16^6}} = \frac{\frac{32}{16^6}}{\frac{234}{16^6}} = \frac{32}{234} = 0.1367$$

g. Untuk P_3=6,

$$p(\tilde{P}_3 = 6 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) = \frac{\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{234}{16^6}} = \frac{\frac{2}{16^6}}{\frac{234}{16^6}} = \frac{2}{234} = 0.0085$$

h. Untuk P_3=7,

$$p(\tilde{P}_3 = 7 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) = \frac{\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{234}{16^6}} = \frac{\frac{2}{16^6}}{\frac{234}{16^6}} = \frac{2}{234} = 0.0085$$

i. Untuk P_3=8,

$$p(\tilde{P}_3 = 8 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) = \frac{\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{234}{16^6}} = \frac{\frac{4}{16^6}}{\frac{234}{16^6}} = \frac{4}{234} = 0.0171$$

j. Untuk P_3=9,

$$p(\tilde{P}_3 = 9 \mid \forall i : \tilde{O}_{i,3} = O_{i,3}) = \frac{\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)}{\frac{234}{16^6}} = \frac{\frac{2}{16^6}}{\frac{234}{16^6}} = \frac{2}{234} = 0.0085$$

4. Untuk j=4

Untuk j=4, Persamaan 4.13b menjadi

$$p(\tilde{P}_4 = P_4 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) =$$

$$\frac{\left[p(\tilde{O}_{1,4} = 5 \mid \tilde{P}_4 = P_4).p(\tilde{P}_4 = P_4)\right]\left[p(\tilde{O}_{2,4} = 0 \mid \tilde{P}_4 = P_4).p(\tilde{P}_4 = P_4)\right]\left[p(\tilde{O}_{3,4} = 2 \mid \tilde{P}_4 = P_4).p(\tilde{P}_4 = P_4)\right]}{\sum_{k=0}^{9}\left\{\left[p(\tilde{O}_{1,4} = 5 \mid \tilde{P}_4 = k).p(\tilde{P}_4 = k)\right]\left[p(\tilde{O}_{2,4} = 0 \mid \tilde{P}_4 = k).p(\tilde{P}_4 = k)\right]\left[p(\tilde{O}_{3,4} = 2 \mid \tilde{P}_4 = k).p(\tilde{P}_4 = k)\right]\right\}}. \quad (4.17)$$

Selanjutnya dengan persamaan 4.17 dihitung $p(\tilde{P}_4 = P_4 \mid \forall i : \tilde{O}_{i,4} = O_{i,4})$ untuk $P_4 \in \{0,..,9\}$.

a. Untuk P_4=0,

$$p(\tilde{P}_4 = 0 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\right]+\left[\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\right]}$$

$$= \frac{\frac{32}{16^6}}{\frac{32}{16^6}+\frac{16}{16^6}+\frac{32}{16^6}+\frac{32}{16^6}+\frac{32}{16^6}+\frac{64}{16^6}+\frac{4}{16^6}+\frac{4}{16^6}+\frac{2}{16^6}+\frac{2}{16^6}} = \frac{\frac{32}{16^6}}{\frac{220}{16^6}} = \frac{32}{220} = 0.1455$$

b. Untuk P_4=1,

$$p(\tilde{P}_4 = 1 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) = \frac{\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{1}{16}\frac{2}{16}\right)}{\frac{220}{16^6}} = \frac{\frac{16}{16^6}}{\frac{220}{16^6}} = \frac{16}{220} = 0.0727$$

c. Untuk P_4=2,

$$p(\tilde{P}_4 = 2 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) = \frac{\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)}{\frac{220}{16^6}} = \frac{\frac{32}{16^6}}{\frac{220}{16^6}} = \frac{32}{220} = 0.1454$$

d. Untuk P_4=3,

$$p(\tilde{P}_4 = 3 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) = \frac{\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)}{\frac{220}{16^6}} = \frac{\frac{32}{16^6}}{\frac{220}{16^6}} = \frac{32}{220} = 0.1454$$

e. Untuk P_4=4,

$$p(\tilde{P}_4 = 4 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) = \frac{\left(\frac{1}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)}{\frac{220}{16^6}} = \frac{\frac{32}{16^6}}{\frac{220}{16^6}} = \frac{32}{220} = 0.1454$$

f. Untuk P_4=5,

$$p(\tilde{P}_4 = 5 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) = \frac{\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)\left(\frac{2}{16}\frac{2}{16}\right)}{\frac{220}{16^6}} = \frac{\frac{64}{16^6}}{\frac{220}{16^6}} = \frac{64}{220} = 0.2909$$

g. Untuk P_4=6,

$$p(\tilde{P}_4 = 6 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) = \frac{\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{220}{16^6}} = \frac{\frac{4}{16^6}}{\frac{220}{16^6}} = \frac{4}{220} = 0.0182$$

h. Untuk P_4=7,

$$p(\tilde{P}_4 = 7 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) = \frac{\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{2}{16}\frac{1}{16}\right)}{\frac{220}{16^6}} = \frac{\frac{4}{16^6}}{\frac{220}{16^6}} = \frac{4}{220} = 0.0182$$

i. Untuk P_4=8,

$$p(\tilde{P}_4 = 8 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) = \frac{\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)}{\frac{220}{16^6}} = \frac{\frac{2}{16^6}}{\frac{220}{16^6}} = \frac{2}{220} = 0.0091$$

j. Untuk P_4=9,

$$p(\tilde{P}_4 = 9 \mid \forall i : \tilde{O}_{i,4} = O_{i,4}) = \frac{\left(\frac{2}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)\left(\frac{1}{16}\frac{1}{16}\right)}{\frac{220}{16^6}} = \frac{\frac{2}{16^6}}{\frac{220}{16^6}} = \frac{2}{220} = 0.0091$$

Dari perhitungan di atas, probabilitas bersyarat $p(\tilde{P}_j = P_j \mid \forall i : \tilde{O}_{i,j} = O_{i,j})$ untuk $1 \leq j \leq 4$ dan $P_j \in \{0,..,9\}$ dapat di sederhanakan sebagaimana dalam Tabel 4.10.

Tabel 4.10 Probabilitas dari $p(\tilde{P}_j = P_j \mid \forall i : \tilde{O}_{i,j} = O_{i,j})$ untuk $P_j \in \{0,..,9\}$

P_j	j = 1	j = 2	j =3	j =4
	$p(\tilde{P}_1 = P_1 \mid \forall i : \tilde{O}_{i,1} = O_{i,1})$	$p(\tilde{P}_2 = P_2 \mid \forall i : \tilde{O}_{i,2} = O_{i,2})$	$p(\tilde{P}_3 = P_3 \mid \forall i : \tilde{O}_{i,3} = O_{i,3})$	$p(\tilde{P}_4 = P_4 \mid \forall i : \tilde{O}_{i,4} = O_{i,4})$
0	0	8,89%	13,67%	14,54%
1	42,11%	8,89%	13,67%	7,27%
2	10,53%	8,89%	13,67%	14,54%
3	10,53%	8,89%	27,35%	14,54%
4	10,53%	35,55%	13,67%	14,54%
5	21,05%	17,78%	13,67%	29,09%
6	2,63%	2,22%	0,85%	1,82%
7	1,32%	2,22%	0,85%	1,82%
8	0,67%	2,22%	1,71%	0,91%
9	0,67%	4,44%	0,85%	0,91%

Dari Tabel 4.10 di atas dapat diketahui P_j yang membuat nilai probabilitas bersyarat $p(\tilde{P}_1 = P_1 \mid \forall i : \tilde{O}_{i,1} = O_{i,1})$ maksimal adalah P_j=1 untuk j=1 dengan nilai maksimal 42,11%. P_j yang membuat nilai probabilitas bersyarat $p(\tilde{P}_2 = P_2 \mid \forall i : \tilde{O}_{i,2} = O_{i,2})$ maksimal adalah P_j=4 untuk j=2 dengan nilai maksimal 35,55%. P_j yang membuat nilai probabilitas bersyarat $p(\tilde{P}_3 = P_3 \mid \forall i : \tilde{O}_{i,3} = O_{i,3})$ maksimal adalah P_j=3 untuk j=3 dengan nilai maksimal 27,35%. P_j yang membuat nilai probabilitas bersyarat $p(\tilde{P}_4 = P_4 \mid \forall i : \tilde{O}_{i,4} = O_{i,4})$ maksimal adalah P_j=5 untuk j=4 dengan nilai maksimal 29,09%. Sehingga dapat disimpulkan empat digit PIN $\hat{P} = (\hat{P}_1, \hat{P}_2 \hat{P}_3, \hat{P}_4)$ yang digunakan oleh nasabah adalah P_j yang membuat probabilitas bersyarat $p(\tilde{P}_j = P_j \mid \forall i : \tilde{O}_{i,j} = O_{i,j})$ maksimal. Jadi $\hat{P} = (1,4,3,5)$.

BAB V
PENUTUP

5.1. Kesimpulan

Berdasarkan uraian pada pembahasan, dapat diambil kesimpulan algoritma DES dapat digunakan untuk mengenkripsi data, yaitu untuk menghitung empat digit PIN yang digunakan nasabah dalam sistem keamanan ATM. Walaupun telah dirancang dengan keamanan bertingkat, namun aplikasi algoritma DES dalam sistem keamanan ATM ini masih membuka peluang *cryptanalys* untuk melakukan *cryptanalysis.*

Salah satu kemungkinan yang dilakukan oleh *cryptanalys* adalah dengan menebak empat digit PIN dalam kartu ATM menggunakan teori probabilitas dengan syarat nilai dari $O_{1,j}$,$O_{2,j}$ dan $O_{3,j}$ harus diketahui. Berdasarkan implementasi kasus dapat disimpulkan empat digit PIN $\hat{P} = (\hat{P}_{1,}\hat{P}_{2}\hat{P}_{3,}\hat{P}_{4})$ yang dicari adalah nilai dari $P_j \in \{0,..,9\}$ yang membuat probabilitas bersyarat $p(\tilde{P}_j = P_j \left| \forall_i : \tilde{O}_{i,j} = O_{i,j} \right.)$ maksimal.

5.2. Saran

Dalam skripsi ini digunakan teori probabilitas untuk melakukan *cryptanalysis* terhadap sistem keamanan ATM dengan nilai $O_{1,j}$,$O_{2,j}$ dan $O_{3,j}$ diketahui. Pembaca yang tertarik dapat menggunakan metode yang sama untuk nilai $O_{1,j}$,$O_{2,j}$ dan $O_{3,j}$ tidak diketahui. Pembaca juga bisa melakukan *cryptanalysis* terhadap sistem keamanan ATM menggunakan metode lain, misalnya menggunakan tabel desimalisasi *adaptive*.

DAFTAR PUSTAKA

[1] Bain, L.J., and M. Engelhardt, *Introduction to Probability and Mathematical Statistics*, 2 ed., Duxbury Press, Belmont, California, 1992.

[2] Kuhn, M.G., *Probability Theory for Pickpockets -- ec-PIN Guessing*, http://www.cl.cam.ac.uk/~mgk25/ec-pin-prob.pdf, 1997.

[3] Kurniawan, Y., *Kriptografi : Keamanan Internet dan Jaringan Komunikasi*, C.V. Informatika, Bandung, 2004.

[4] Mandal, M., *ATM PIN Security*, Paladion Network Pvt. Ltd, United States of Amerika, 2004.

[5] Menezes, P., V. Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press LLC, 1997.

[6] Munir, R, *Bahan Kuliah Kriptografi : Data Encryption Standard (DES)*, Departemen Teknik Informatika Institut Teknologi Bandung, Bandung, 2004

[7] Purbo, O.W. dan A.A. Wahyudi, *Mengenal e-Commerce*, P.T. Elex Media Komputindo, Jakarta, 2001.

[8] Rahardjo, B., *Keamanan Sistem Informasi Berbasis Internet*, 5 ed., P.T. Insan Infonesia, Bandung, 2002.

[9] Schneier, B., *Aplied Cryptography : Protocol, Algorithms, and Source Code in C*, 2 ed., John Wiley and Sons, 1996.

[10] Setiadi, B., Tugas Akhir Kuliah Security : *Analisis Keamanan Data dengan Menggunakan Metode DES dan Metode Gost*, Program Magister Teknik Elektro, ITB, Bandung, 2004.

[11] Stallings, W., *Cryptography and Network Security : Principle and Practice*, 3 ed., Prentice Hall, New Jersey, 2003.

[12] Stinson, D.R., *Cryptography Theory and Practice*, CRC Press LLC, United States of Amerika, 1995.

[13] Supranto, J., *Statistik Teori dan Aplikasi*, Penerbit Erlangga, 1989.

[14] Walpole, R.E., R.H. Myers, S.L. Myers, and Keying Yo, *Probability & Statistic for Engineers & Scientist*, 7ed, Prentice Hall, New Jersey, 2002.

[15] Wibowo, W.A., Tugas Kuliah Keamanan Sistem Informasi : *Advanced Encription Standard, Algoritma Rijndael*, Departemen Teknik Elektro, ITB, Bandung, 2004.

LAMPIRAN 1

Bukti Teorema

Bukti Teorema 2.4

Bukti :

Untuk membuktikan Teorema 2.4, dibuktikan dua persamaan yaitu untuk $P(A\cap B)=P(B)P(A\mid B)$ dan $P(A\cap B)=P(A)P(B\mid A)$

(i) Dari Definisi 2.3 dijelaskan peluang bersyarat kejadian B jika diketahui kejadian A dengan $P(A)\neq 0$ didefinisikan sebagai berikut.

$$P(B\mid A)=\frac{P(A\cap B)}{P(A)}$$

Sehingga dari Definisi 2.3 tersebut diperoleh $P(A\cap B)=P(A)P(B\mid A)$.

(ii) Sedangkan peluang bersyarat kejadian A jika diketahui kejadian B dengan $P(B)\neq 0$ didefinisikan sebagai berikut.

$$P(A\mid B)=\frac{P(A\cap B)}{P(B)}$$

Sehingga dari definisi di atas diperoleh $P(A\cap B)=P(B)P(A\mid B)$

Dari (i) dan (ii) terbukti jika A dan B dua kejadian di dalam ruang sampel S maka $P(A\cap B)=P(B)P(A\mid B)=P(A)P(B\mid A)$.

Bukti Teorema 2.5

Bukti :

Diketahui A suatu kejadian dalam ruang sampel S. Jika $B_1, B_2,\ldots, B_k$ merupakan partisi dari ruang sampel S dengan $P(B)\neq 0$ untuk i=1,2,…k maka

$B_1\cup B_2\cup\ldots\cup B_k=S$. Sehingga kejadian A dalam ruang sampel S dapat dituliskan $A=(A\cap B_1)\cup(A\cap B_2)\cup\ldots\cup(A\cap B_K)$ dan

$$P(A)=P\big((A\cap B_1)\cup(A\cap B_2)\cup\ldots\cup(A\cap B_K)\big)$$
$$P(A)=P(A\cap B_1)+P(A\cap B_2)+\ldots+P(A\cap B_K)$$
$$P(A)=\sum_{i=1}^{k}P(A\cap B_i)=\sum_{i=1}^{k}P(B_i\cap A)$$

Dengan Teorema 2.4 persamaan di atas menjadi,

$$P(A) = \sum_{i=1}^{k} P(B_i)P(A \mid B_i)$$

Jadi terbukti bahwa $P(A) = \sum_{i=1}^{k} P(B_i \cap A) = \sum_{i=1}^{k} P(B_i)P(A \mid B_i)$.

Bukti Teorema 2.6

Bukti :

Untuk membuktikan Teorema 2.6 digunakan beberapa definisi dan teorema yang bersesuaian. Dengan Definisi 2.3 diperoleh $P(B_i \mid A)$ yaitu

$$P(B_i \mid A) = \frac{P(A \cap B_i)}{P(A)} .$$

Selanjutnya dengan Teorema 2.4 persamaan di atas dapat ditulis kembali menjadi

$$P(B_i \mid A) = \frac{P(B_i)P(A \mid B_i)}{P(A)} .$$

Dengan Teorema 2.5, *P(A)* diganti dengan $\sum_{i=1}^{k} P(B_i)P(A \mid B_i)$ sehingga diperoleh persamaan $P(B_i \mid A) = \frac{P(B_i)P(A \mid B_i)}{\sum_{i=1}^{k} P(B_i)P(A \mid B_i)}$.

LAMPIRAN 2

Subtitution Box (S-Box)

S-Box 1: Substitution Box 1																
Baris/Kolom	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-Box 2: Substitution Box 2																
Baris/Kolom	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S-Box 3: Substitution Box 3																
Baris/Kolom	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S-Box 4: Substitution Box 4															
Baris/Kolom	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S-Box 5: Substitution Box 5																
Baris/Kolom	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S-Box 6: Substitution Box 6																
Baris/Kolom	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S-Box 7: Substitution Box 7																
Baris/Kolom	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S-Box 8: Substitution Box 8																
Baris/Kolom	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

LAMPIRAN 3

Output dari 16 putaran pada proses enkripsi menggunakan algoritma DES pada contoh kasus 4.1.6.

$E(R_0)$	=	011110100001010101010101011110100001010101010101
K_1	=	000110110000000101110111111111110001110000001110010
$E(R_0) \oplus K_1$	=	011000010001011110111010100001100110010100100111
S-box output	=	01011100100000101011010110010111
$f(R_0 , K_1)$	=	00100011010010101010100110111011
$L_2 = R_1$	=	11101111010010100110010101000100

$E(R_1)$	=	011101011110101001010100001100001010101000001001
K_2	=	011110011010111011011001110110111100100111100101
$E(R_1) \oplus K_2$	=	000011000100010010001101111010110110001111101100
S-box output	=	11111000110100000011101010101110
$f(R_1 , K_2)$	=	00111100101010111000011110100011
$L_3 = R_2$	=	11001100000000010111011100001001

$E(R_2)$	=	111001011000000000000010101110101110100001010011
K_3	=	010101011111110010001010010000101100111110011001
$E(R_2) \oplus K_3$	=	101100000111110010001000111110000010011111001010
S-box output	=	00100111000100001110000101101111
$f(R_2 , K_3)$	=	01001101000101100110111010100000
$L_4 = R_3$	=	10100010010111000000101111110100

$E(R_3)$	=	010100000100001011111000000001010111111110101001
K_4	=	011100101010110111010110110110110011010100011101
$E(R_3) \oplus K_4$	=	001000101110111100101110110111100100101010110100
S-box output	=	00100001111011011001111100111010
$f(R_3 , K_4)$	=	10111011001000110111011101001100
$L_5 = R_4$	=	01110111001000100000000001000101

$E(R_4)$	=	101110101110100100000100000000000000001000001010
K_5	=	011111001110110000000111111010110101001110101000
$E(R_4) \oplus K_5$	=	110001100000010100000011111010110101000110100010
S-box output	=	01010000110010000011000111101011
$f(R_4 , K_5)$	=	00101000000100111010110111000011
$L_6 = R_5$	=	10001010010011111010011000110111

$E(R_5)$	=	11000101010000100101111111010000110000011010111
K_6	=	01100011101001010011111001010000011110110010111
$E(R_5) \oplus K_6$	=	10100110111001110110000110000000101110101000000
S-box output	=	01000001111100110100110000111101
$f(R_5, K_6)$	=	10011110010001011100110100101100
$L_7 = R_6$	=	11101001011001111100110101101001

$E(R_6)$	=	11110101001010110000111111100101101010110101010011
K_7	=	11101100100001001011011111110110000110001011110 0
$E(R_6) \oplus K_7$	=	00011001101011111011100000010011101100111110111 1
S-box output	=	00010000011101010100000010101101
$f(R_6, K_7)$	=	10001100000001010001110000100111
$L_8 = R_7$	=	00000110010010101011101000010000

$E(R_7)$	=	00000000110000100101010101011111010000001010000 0
K_8	=	11110111100010100011101011000001001110111111101 1
$E(R_7) \oplus K_8$	=	11110111010010000110111110011110011111011010110 11
S-box output	=	01101100000110000111110010101110
$f(R_7, K_8)$	=	00111100000011101000011011111001
$L_9 = R_8$	=	11010101011010010100101110010000

$E(R_8)$	=	01101010101010110101001010100101011111001010000 1
K_9	=	11100000110110111110101111101101111001111000000 1
$E(R_8) \oplus K_9$	=	10001010011100001011100101001000100110110010000 0
S-box output	=	00010001000011000101011101110111
$f(R_8, K_9)$	=	00100010001101100111110001101010
$L_{10} = R_9$	=	00100100011111001100011001111010

$E(R_9)$	=	00010000100000111111100101100000110000111111010 0
K_{10}	=	10110001111100110100011110111010010001100100111 1
$E(R_9) \oplus K_{10}$	=	10100001011100001011111011011010100001011011101 1
S-box output	=	11011010000001000101001001110101
$f(R_9, K_{10})$	=	01100010101111001001110000100010
$L_{11} = R_{10}$	=	10110111110101011101011110110010

$E(R_{10})$	=	01011010111111101010101111101010111111011010010 1
K_{11}	=	00100001010111111101001111011110110100111000011 0
$E(R_{10}) \oplus K_{11}$	=	01111011101000010111100000110100001011100010001 1
S-box output	=	01110011000001011101000100000001
$f(R_{10}, K_{11})$	=	11100001000001001111101000000010
$L_{12} = R_{11}$	=	11000101011110000011110001111000

$E(R_{11})$	=	011000001010101111110000000111111000001111110001
K_{12}	=	011101010111000111110101100101000110011111101001
$E(R_{11}) \oplus K_{12}$	=	000101011101101000000101100010111110010000011000
S-box output	=	011110111000101100100110001101 01
$f(R_{11}, K_{12})$	=	110000100110100011001111111010 10
$L_{13} = R_{12}$	=	011101011011110100011000010110 00

$E(R_{12})$	=	001110101011110111111010100011110000001011110000
K_{13}	=	100101111100010111010001111110101011101001000001
$E(R_{12}) \oplus K_{13}$	=	101011010111100000101011011101011011100010110001
S-box output	=	100110101101000110001011010011 11
$f(R_{12}, K_{13})$	=	110111011011101100101001001000 10
$L_{14} = R_{13}$	=	000110001100001100010101011010 10

$E(R_{13})$	=	000011110001011000000110100010101010101011110100
K_{14}	=	010111110100001110110111111100101110011100111010
$E(R_{13}) \oplus K_{14}$	=	010100000101010110110001011110000100110111001110
S-box output	=	011001000111100110011010111 10001
$f(R_{13}, K_{14})$	=	101101110011000110001110010101 01
$L_{15} = R_{14}$	=	110000101000110010010110000011 01

$E(R_{14})$	=	111000000101010001011001100010101100000001011011
K_{15}	=	101111111001000110001101001111010011111100001010
$E(R_{14}) \oplus K_{15}$	=	010111111100010111010100011101111111111101010001
S-box output	=	101100101110100010001101001 11100
$f(R_{14}, K_{15})$	=	010110111000000100100111011011 10
$L_{16} = R_{15}$	=	010000110100001000110010001 10100

$E(R_{15})$	=	001000000110101000000100000110100100000110101000
K_{16}	=	110010110011110110001011000011100001011111110101
$E(R_{15}) \oplus K_{16}$	=	111010110101011110001111000101000101011001011101
S-box output	=	101001111000001100100100001 01001
$f(R_{15}, K_{16})$	=	110010001100000000100111110011 000
R_{16}	=	000010100100110011011001100 10101