LAPORAN TUGAS AKHIR

ENKRIPSI DATA DENGAN JAVA PADA SISTEM KOMUNIKASI WEB SERVER

Pengaju :

Nama : Muh Syatir Kaffah
NIM : 41405110116

Peminatan :
Teknik Telekomunikasi

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS MERCU BUANA

TAHUN 2006/2007

A. LEMBAR PENGESAHAN

Proposal dengan Judul :

ENKRIPSI DATA DENGAN JAVA PADA SISTEM KOMUNIKASI WEB SERVER

Yang diajukan oleh :

Nama	: Muh Syatir Kaffah Y A
NIM	: 41405110116
Jurusan	: Teknik Elektro – Telekomunikasi
No Telp	: (021)68860401

Telah disetujui untuk diajukan pada tanggal 28 November 2007 oleh :

Pembimbing :

(**Ir Bambang Hutomo BcTT**)

Koordinator Tugas Akhir | **Ketua Jurusan Teknik Elektro**

(Yudhi Gunardhi ST.MT) | **(Ir.Budiyanto H MSc)**

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I. PENDAHULUAN

I.1. Abstrak

Dalam era global network dan *electronic commerce* saat ini, teknologi keamanan dengan perangkat kriptografi atau enkripsi sangat dibutuhkan, karena pada sistem jaringan terlebih pada *public network* sangatlah tidak aman, teknik enkripsi data akan memungkinkan suatu data yang akan dikirim untuk dicak sehingga tidak dapat diterjemahkan dengan mudah oleh penerima data yang ditak diinginkan dan kemudian pada sisi penerima data / tujuan akan didekripsi-kan kembali untuk mendapatkan terjemahan data yang benar. Tugas akhir ini dilaksanakan antara bulan Oktober 2006 sampai dengan April 2007 ini bertujuan untuk melakukan analisa teknologi sekuriti digital yang lebih mendalam terutama dengan menggunakan perangkat kriptografi atau enkripsi berupa software. Fokus dari penelitian ini adalah pemanfaatan teknologi kriptografi untuk sekuriti digital yang diaplikasikan pada **bahasa pemograman Java**.

I.2. Latar Belakang

Sekuriti Digital adalah teknologi yang digunakan untuk melindungi informasi dari pencurian dan pemalsuan lewat tehnik penyandian atau enkripsi data yang praktis tidak bisa ditembus. Pada saat ini, perkembangan teknologi Internet, Sekuriti Digital dan Electronic Commerce bergerak amat cepat dalam mendefinisikan teknologi informasi masal yang baru. Kombinasi ketiganya memungkinkan dikembangkannya komunikasi elektronis pada skala masal dan global, dengan faktor keamanan data (dari pencurian dan pemalsuan) yang tinggi. Ini menjadikan teknologi enkripsi data sangat strategis nilainya untuk Indonesia di era Internet dan jaringan global berbasis LAN, MAN dan WAN.

Indonesia memiliki banyak kepentingan untuk mengembangkan sarana komputasi / komunikasi modern demi menunjang pembangunan. Tak bisa dipungkiri bahwa bisnis masa depan mengarah pada sistem e-commerece bahkan satu bisnis tertentu dapat menggunakan jaringan publik/umum untuk digunakan untuk pertukaran informasi yang bersifat khusus atau rahasia. Penggunaan teknologi informasi publik tanpa sekuriti data sangatlah berbahaya. Sebaliknya, eksploitasi teknologi sekuriti akan memberi jalan ke berbagai kemungkinan bernilai strategis yang sebelumnya

dianggap tidak mungkin. Dalam konteks ini, penting bagi Indonesia untuk mengantisipasi makin pentingnya isu sekuriti dalam tatanan dunia informasi modern.

I.3. Tujuan Penulisan

Pada era komunikasi digital sekarang ini bidang telekomunikasi tidak harus berkutat pada perancangan ataupun instalasi hardware saja, namun juga harus berpadu dengan teknik software digital pengiriman data. Disini penulis berusaha menggabungkan dua bidang ilmu tersebut yaitu Telekomunikasi – Informatika menjadi telematika.

Secara umum penelitian ditujukan untuk melakukan analisa lebih dalam tentang ilmu dan pengembangan teknologi telekomunikasi berbasis sekuriti digital, pada bidang telekomunikasi teknik ini disebut "Enkripsi Data" yang secara aljabar-algoritma dipelajari dalam sistem telekomunikasi sedangkan pengaplikasian teorema enkripsi tersebut dituangkan dalam software dari sistem telekomunikasi.

I.4. Teori Dasar

Enkripsi Data

Enkripsi data adalah proses pengambilan data dan pengolahan data yang menghasilkan data tak berarti dimana orang ketiga tidak akan mengerti arti data tersebut. Yang kemudian akan di deskripsikan pada sisi penerima / tujuan data hingga data tersebut kembali mempunyai arti yang sama dari pengirim.

Pada dasarnya enkripsi hanya mengacak data dari alphabet berarti ke alphabet tidak berarti ataupun ke numerik yang takberarti.

Password-Based Encryption

Password-Base Encryption adalah pemberian password pengacakan data hingga akan lebih banyak menghabiskan banyak waktu bagi pencuri data sebelum mengakses pada pendekripsian data.

I.5. Rencana Hasil Tugas Akhir

Tugas akhir ini mempunyai tahap-tahap pelaksanaan sebagai berikut :

Pengumpulan teorema-teorema dasar algoritma encripsi kemudian memilih sistem enkripsi yang terbaik.

Perancangan algoritma perograman dengan bahasa pemrograman JAVA untuk membantu proses encripsi data dalam proses pengiriman dan penerimaan data. Setelah pembuatan program telah selesai maka akan dilakukan pengetesan pada jaringan peer to peer cennection untuk melakukan uji coba kehandalan proses enkripsi. Hasil akhir dari perancangan program akan diterapkan pada aplikasi Web Server yang diharapkan akan memberikan sekuritas terbaik dalam proses pengiriman dan penerimaan data pada jaringan publik

I.6. Sistemetika Penulisan

I. PENDAHULUAN

I.1. Abstrak

I.2. Latar Belakang

I.3. Tujuan Penulisan

I.4. Dasar Teori

I.5. Rencana Hasil Tugas Akhir

I.6 Sistematikan Penulisan

II. Dasar Teori

II.1. Jaringan Webserver

II.2. Algoritma Enkripsi

II.3. Protokol Pengiriman Data pada Jaringan

III. PERANCANGAN

III.1. Perancangan Flowchart pada Enkripsi Data

III.2. Perancangan Algoritma Pemrograman

III.3. Performansi Hasil Pemrograman

IV. IMPLEMENTASI

IV.1. Hasil uji Implementasi AES pada Jaringan

V. KESIMPULAN

V.1. Kesimpulan

V.2. Saran

DAFTAR PUSTAKA

LAMPIRAN

BAB II
DASAR TEORI

II.1. Jaringan Webserver

Dewasa ini di Indonesia komunikasi data merupakan hal yang sudah umum dilakukan baik oleh perorangan, kelompok, maupun perusahaan dengan berbagai media yang tersedia. Beragam jenis pengguna komunikasi data tersebut menggunakan media yang telah tersedia untuk mengirimkan berbagai jenis informasi, tanpa dibatasi oleh ruang dan waktu, secara murah dan efisien. Salah satu jenis komunikasi data yang paling banyak digunakan adalah Internet. Internet dipilih karena saat ini merupakan alternatif komunikasi paling murah, mudah dipelajari, dan titik akses yang paling luas.

Semenjak isu privasi jaringan dan kemanan jaringan mulai dimunculkan pada jaringan publik / innternet, penelitian mengenai perlindungan informasi yang dikirim pada jaringan publik mulai meningkat. Cabang ilmu "kriptografi" yang juga dikenal dengan istilah "chiper" membentuk aspek mendasar dalam usaha penelitian kemanan jaringan. Sistem algoritma kriptografi yang paling sering digunakan dalam 20 tahun terakhir adalah *DES (Data Encryption Standard)* yang mulai diperkenalkan awal tahun 70-an. DES menjadi standar enkripsi pada tahun 1977.

Pada tahun 1983 mulai terlihat kelemahan DES pada pengiriman data dengan kata kunci yang pendek (byte kecil), kemudian muncul jenis enkripsi terbaru sebagai pengembangan dari DES yaitu Triple-DES yang memperformasikan enkripsi DES tiga kali perulangan. Namun demikian terjadi kendala kecepatan pada aplikasi jaringan yang padat. Di tahun 1997 *National Institute of Standard and Technology* mulai mengembangkan standar kriptografi selain DES dan Triple DES yang diharapkan mempunyai tingkat keamanan yang sama namun memiliki performa kecepatan yang lebih baik. Kemudian sistem enkripsi baru tersebut di kenal sebagai Rijndael yang lebih dikenal dengan *AES (Advance Encryptioan Standard).* Perkembangan kecepatan jaringan yang semakin tinggi mengarah pada kebutuhan teknolgi sekuriti informasi yang baik, saat ini pelayanan keamanan jaringan mulai diimplementasikan seperti pada Public Key Infrastructure (PKI), aplikasi WEB Server, Route berkecepatan tinggi dan Firewalls.

Sistem kemanan komunikasi saat ini berdasar pada berbagai variasi protokol seperti Secure Socket Layer (SSL), IP Security, Transport Layer Security (TLS). Protokol-protokol tersebut tidak hanya memakai satu jenis sistem kriptografi saja tapi berbagai macam jenis kriptografi berlapis.

Ada beberapa perangkat penting yang harus dikenal dalam sistem transmisi data pada jaringan internet, yaitu repeater, bridges, router, gateaway dan proxy.

Repeater adalah jenis koneksi paling mendasar dalam berkomunikasi dimana repeater ini hanya berfungsi sebagai penguatan signal data tanap terjadi pengolahan data pada repeater.

Bridges fungsinya hampir sama dengan repeater yaitu sebagai penguat namun dalam hal ini bridges berfungsi untuk menghubungkan sekaligus beberapa jaringan.

Router adalah jembatan utama dalam suatu jaringan dimana terjadi pengolahan data dalam sebuah router. Router menyamakan header paket ke segmen LAN dan memilih jalur komunikasi terbaik untuk pengiriman paket dan sekaligus melakukan optimalisasi performansi komunikasi jaringan.

Gateaway berfungasi sebagai penterjemah atau penghubung jaringan yang mempunyai protokol lapisan OSI yang berbeda.

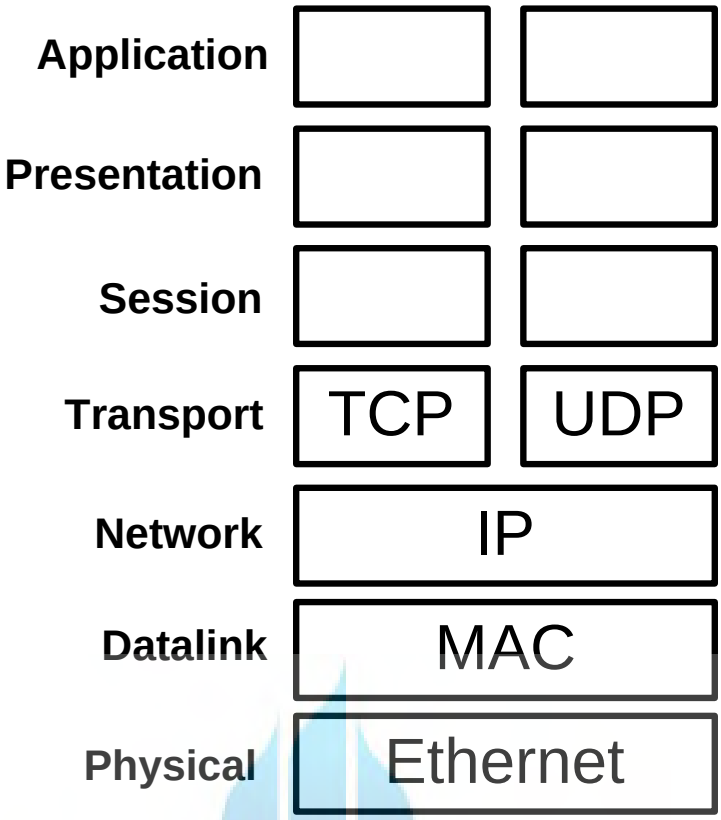
Proxy adalah metode penyimpanan transmisi data dalam bentuk lain dan proxy membutuhkan gateaway untuk aplikasi protokol jaringan.

Macam macam protokol pada jaringan antara lain adalah *IPX/ SPX (Internetwork Packet Exchange / Sequenced Packet Exchange), UDP (User Datagram Protocol) dan TCP/IP (Transfer Control Protocol / Internet Protocol).*

II.1.a. Memahami cara kerja TCP/IP.

TCP adalah suatu protokol pengiriman data yang berbasis *Internet Protocol (IP)* dan bersifat connection oriented. Pada *OSI (Open System Interconnection)* layer TCP berada pada layer transport yang fungsinya mengatur pengiriman suatu data dari client ke server.

OSI tersiri dari tujuh layer yang secara umum dibagi menjadi dua kelompok yaitu *Upper Layer* dan *Lower layer* atau biasa juga disebut aplication layer dan data transport layer. Yang termasuk dalam layer aplikasi adalah aplikasi, presentation, dan session.



Gambar 2.1. Posisi TCP pada Layer OSI

Layer Aplikasi berfungsi sebagai interface antara komputer dengan user, layer ini bertangung jawab untuk mengidentifikasi ketersediaan hubungan komunikasi, menetukan ketersediaan "resource" dan melakukan sinkronisasi komunikasi.

Layer presentasi berfungsi untuk menyediakan sistem penyajian data ke layer aplikasi berupa kode format dan proses konversi dimana bentuk yang dihasilkan akan berupa bentuk data yang umum dan bisa dibaca oleh semua jenis sistem.

Layer Session bertanggung jawab dalam proses pembentukan , pengolahan dan pemutusan sesi antar sistem yang sedang berkomunikasi.

Layer transport berfungsi dalam tiga tugas utama yaitu : pengemasan data upper ke dalam segmen dan menyediakan mekanisme multiplexing dari upper layer, pengiriman segmen antar host, penetapan bentuk hubungan antara host pengirim dan penerima..

Layer Network bertanggung jawab dalam pengalamatan data yang akan dikirim yang lebih sering dikenal dengan IP Address.

Layer Data link akan menjamin bahwa pesan dikirim ke media yang tepat dan juga menterjemahkan kode dari network layer ke dalam bentuk binner untuk kemudian diberikan pada physical layer.

Layer Phisycal berupa bentuk fisik alat yang akan mengirim data, physical layer dapat berupa Ethernet Card, Modem, Router dan alat lainnya.

berupa Ethernet Card, Modem, Router dan alat lainnya.

Umumnya pada sebuah jaringan yang menggunakan server akan terlebih dahulu berkomunikasi dengan server jika ingin mengirimkan data keluar dari jaringan. Pada server inilah dapat ditambahkan program enkripsi, namun jika proses enkripsi ingin dibentuk pada komputer user, hal inipun dapat dilakukan.

Untuk pengiriman datanya, pada masing-masing komputer (client server) akan menggunakan port dengan pendefinisian terlebih dahulu. Kemudian dari client akan mengirimkan data dari port pada PCnya ke arah port pada PC servernya

Gambar 2.2 Komunikasi Client dan Server melaui port tertetu

II.1.b. Secure Socket Layer (SSL)

Umumnya enkripsi diterapkan pada sistem komunikasi data yang membutuhkan tingkat kemanan yang tinggi. Sebuah sistem enkripsi diterapkan pada software yang digunakan dalam berkomunikasi, software ini bisa berada pada hardware maupun pada software yang dapat di program ulang. Enkripsi yang diterapkan pada hardware biasanya di tempatkan pada prosesor router ataupun perangkat lain yang berhubungan dengan komunikasi jaringan.

Meski sebuah sistem informasi sudah dirancang memiliki perangkat pengamanan, dalam operasi masalah keamanan harus selalu dimonitor. Hal ini disebabkan oleh beberapa hal, antara lain:

- Ditemukannya lubang keamanan (security hole) yang baru. Perangkat lunak dan perangkat keras biasanya sangat kompleks sehingga tidak mungkin untuk diuji seratus persen. Kadang-kadang ada lubang keamanan yang ditimbulkan oleh kecerobohan implementasi.
- Kesalahan konfigurasi. Kadang-kadang karena lalai atau lupa, konfigurasi sebuah sistem kurang benar sehingga menimbulkan lubang keamanan. Misalnya mode (permission atau kepemilikan) dari berkas yang menyimpan password (/etc/passwd di sistem UNIX) secara tidak sengaja diubah sehingga dapat diubah atau ditulis oleh orang-orang yang tidak berhak.
- Penambahan perangkat baru (hardware dan/atau software) yang menyebabkan menurunnya tingkat security atau berubahnya metoda untuk mengoperasikan sistem. Operator dan administrator harus belajar lagi. Dalam masa belajar ini banyak hal yang jauh dari sempurna, misalnya server atau software masih menggunakan konfigurasi awal dari vendor (dengan password yang sama).

Pada dasarnya model komunikasi yang terjadi pada client dapat digambarkan sebgai berikut :

1. Membuka koneksi (establish) ke host yang dituju.
2. Mendapatkan stream (Input / Output) dari koneksi yang dibuka.
3. Mengetahui aapa yang akan dilakukan terhadap I/O stream.
4. Menutup Stream
5. Release koneksi dengan host.

Pengertian Secure Socket Layer adalah Protokol berlapis. Dalam tiap lapisannya, sebuah data terdiri dari panjang, deskripsi dan isi. SSL mengambil data untuk dikirimkan, dipecahkan kedalam blok-blok yang teratur, kemudian dikompres jika perlu, menerapkan MAC, dienkripsi, dan hasilnya dikirimkan. Di tempat tujuan, data didekripsi, verifikasi, dekompres, dan disusun kembali. Hasilnya dikirimkan ke klien di atasnya.

Protokol SSL merupakan salah satu metode enkripsi dalam berkomunikasi dalam jaringan public, pada awalnya dkembangkan oleh Netscape untuk menjamin pengiriman data melalui HTTP, LDAP ataupun POP3 pada “Aplication Layer”. SSL di design untuk membuat TCP sebagai layer komunikasi dapat menjamin reabilitas

antara dua poin koneksi. Pada perkembangannya SSL digunakan untuk melindungi data pada setiap service yang dikerjakan suatu jaringan, saat ini hampir setiap alamat HTTP pada Web Server menyeiakan support untuk sesi SSL. Bagaimana SSL berjalan dapat digambarkan sebagai berikut :

- Pada saat koneksi mulai berjalan, klien dan server membuat dan mempertukarkan kunci rahasia, yang dipergunakan untuk mengenkripsi data yang akan dikomunikasikan. Meskipun sesi antara klien dan server diintip pihak lain, namun data yang terlihat sulit untuk dibaca karena sudah dienkripsi.
- SSL mendukung kriptografi public key, sehingga server dapat melakukan autentikasi dengan metode yang sudah dikenal umum seperti RSA dan Digital Signature Standard (DSS).
- SSL dapat melakukan verifikasi integritas sesi yang sedang berjalan dengan menggunakan algoritma digest seperti MD5 dan SHA. Hal ini menghindarkan pembajakan suatu sesi.

II.1.c. Serangan pada Jaringan

Berbagai macam cara dapat digunakan dalam mencuri data yang sedang terkirim pada sebuah jaringan diantaranya yaitu:

- Packet Sniffing
- Password Guessing
- Spoofing
- Viruses
- Denial – of – service (DoS)

1. **Packet Sniffing**

   Packet Sniffing berarti mencuri dengar packet yang dikirim dan berusaha menterjemahkannya. Cara ini tidak akan mengubah protokol yang sedang terkirim namun semua traffic yang terjadi pada jaringan akan direkam untuk kemudian berusaha dipecahkan methode pengiriman datanya ataupun mendapatkan file yang hendak di ambil atau dicuri.

2. **Password Guessing**

   Pasword Guessing adalah mencoba mendapatkan fasilitas layaknya seorang administrator dengan mencoba berbagai password secara automatis. Cara ini

akan membutuhkan waktu yang cukup lama karena terlalu banyak kemungkinan alphabet dan numerik yangm ungkin dipakai oleh seorang administrator dalam menentukan password. Ada kalanya seorang administrator membuat password yang mudah diingat dan sangat dekat dengan lingkungannya sehingga jika seorang hacker mengenal betul seorang administrator akan sangat mudah untuk menebak password yang mungkin dipakai oleh seorang administrator.

3. **Spoofing**

   Spoofing merupakan sistem penyamaran data yaitu dengan mencoba mengirimkan data palsu ke sebuah pusat jaringan hingga pusat jaringan tersebut menganggap data yang diterima adalah data yang dapat dipercaya kemudian saat datam ulai diproses maka saat itu pula kemanan sistem jaringan mulai dipelajari dan dipecahkan. Caranya cukup dengan mencari nama host atau IP Address yang termasuk dalam list jaringan yang dipercaya administrator kemudian mengirimkan data menggunakan alamat tersebut.

4. **Virus**

   Virus digunakan tidak untuk mencuri data namun untuk menghancurkan jaringan yang telah ada banyak sekali jeni-jenis virus yang dikenal saat ini diantaranya adalah:

   - Virus e-mail
   - Worm
   - Trojan horse

   Virus e-mail adalah jenis virus yang umum dikenal semenjak virus komputer pertama kali dikenal, hanya penyebarannya dilakukan dengan mengirimkan e-mail ke salah satu alamat jaringan sehingga jika penerima e-mail membuka e-mail yang tak dikenal tersebut dengan segera virus akan menggerogoti sistem.

   Worm merupakan jenis virus yang masuk pada jaringan melalui jaringan internet yang bisa menduplikat dirinnya secara terus menerus hingga space yang rtersisa untuk menyimpan data semakin sedikit dan berakibat semakin lambatnya kinerja sebuah jaringan.

   Trojan horse merupakan jenis virus yang akan masuk kejaringan dan merusak akses setiap user yang diperbolehkan pada jaringan tersebut sehingga pada

jaringan yang terinfeksi virus trojan tidak akan mengenali user-user yang seharusnya memiliki akses ke jaringan tersebut.

5. **Denial of Service**

   Denial of Service dilakukan oleh para hacker jika segala macam cara yang dilakukan tidak dapat menmbus kemanan jaringan tersebut maka dia akan meng-eksploit jaringan tersebut sehingga tidak dapat berfungsi sama sekali. Ini merupakan kelemahan jaringan berbasis windows dengan adanya fasilitas flooding, winkill, ping o death, NT crash.

Pemrograman jaringan dikembangkan untuk menyambungkan proses-proses komputasi yang lumrahnya berjalan dalam mesin-mesin yang berbeda yang terhubung dalam suatu jaringan komputer. Contohnya adalah :

- Web browser mengakses Web server melalui jaringan.
- Aplikasi basis data, di mana terdapat client dan server yang berada di mesin komputer yang terpisah, dan berinteraksi dalam dalam jaringan.
- Aplikasi e-mail di mana terdapat mail client berkomunikasi dengan mail server.

Dalam pemrograman jaringan, inti tantangannya adalah bagaimana caranya dapat mengirimkan data dari sebuah proses ke proses lainnya melalui jaringan yang menghubungkan keduanya. Jaringan antara dua proses ini dapat dianalogikan dengan saluran telepon antara dua pembicara. Dalam analogi sambungan telepon, seorang pembicara dapat berbicara di satu sisi , kemudian suaranya akan mengalir melalui kabel telepon ke lawan bicaranya, sehingga lawan bicara dapat mendengarnya, dan sebaliknya.

Dalam pemrograman jaringan, setelah kita membuat sebuah saluran data antara dua proses, maka masing-masing proses akan memegang satu ujung. Setiap ujung darisaluran data ini disebut socket. Aliran data antara kedua socket ini disebut data stream. Masing-masing socket menggenggam dua stream, yaitu output stream dan input stream.

Sebuah proses dapat mengirim data ke proses lain dengan menuliskannya ke output stream yang dimiliki sebuah socket. Sebuah proses dapat menerima data dengan membacanya dari input stream yang dimiliki sebuah socket. Dengan cara ini maka dapat terjadi pertukaran data, komunikasi dan interaksi antara dua proses yang

berbeda meskipun mereka terpisah dalam jaringan. Proses-proses ini dapat terpisah tetapi terjaring, baik dalam intranet, extranet atapun internet, tidak peduli apakah satu proses berada di Afrika Tengah dan proses lainnya berada di tengah Samudra Pasifik.

Layanan jaringan publik pada jaringan internet diantaranya adalah :

1. **Web Server**

   Web server lebih dikenal dengan layanan WWW atau World Wide Web adalah aplikasi multiplatform dan berbasis grafis atau sering di sebut GUI atau Graphic User Interface. Bahasa pemrograman yang digunakan adalah markup language yaitu bahasa yang sangat populer dan di support oleh semua web server serng juga disebut HTML atau Hypertext Markup Language.

   Web Server menggunakan port 80 sebagai jalur komunikasi, web server mempunyai kemampuan untuk mentransmisi data berupa text, database, suara, gambar dan video secara realtime, sedang kecepatan transmisinya tergantung kepada kecepatan layanan koneksi yang dimiliki.

2. **FTP server**

   File Transfer Protocol, atau sering disebut dengan FTP adalah suatu fungsi management antar mesin dalam jaringan tanpa melalui sesi remote dalam telnet. FTP memiliki kemampuan transfer baik untuk mengirim maupun menerima data, melakukan pengaturan direktori, dan mengakses e-mail. FTP tidak dirancang untuk mengeksekusi langsung program pada mesin tersebut melainkan harus terlebih dahulu di download ke mesin penghubung Kelebihan FTP dibandingkan dengan protokol dan layanan lainnya adalah kecepatan transfer data yang sangat baik.

3. **IRC server**

   Internet Relay Chat atau IRC sangat populer karena IRC menggunakan komunikasi bersifat realtime dan interaktif sehingga setiap individu bisa saling berkomunikasi secara langsung tanpa mengenal batasan wilayah. IRC server menggunakan protokol TCP/IP dengan port 6667 sampai 7001.

II.2. Algoritma Enkripsi

Dalam dunia criptografi dikenal istilah Adavance Encryption Standard (AES) yang juga dikenal sebagai “Rijndael”, yang diharapkan akan dijadikan standard dunia dan adanya analisa bersama yang lebih mendalam hingga mempercepat perkembangannya. Sistem ini mulai dikenal kalangan pengguna jaringan di tahun 2002 dan terus berkembang hingga saat ini. Sebelum AES diperkenalkan, DES (Data Encryption Standard) lebih dahulu dikenal di dunia, karena tingkat kompleksitasnya DES membutuhkan waktu yang lebih lama untuk di crack pada versi Triple-DES namun mempunyai kelemahan yaitu memperlambat sistem komunikasi, sedangkan AES tidak membebani proses pengiriman data dan waktu untuk meng-crack kode AES membutuhkan jutaan jam. Pada tabel dibawah dapat dilihat perbedaan AES dengan Triple-DES.

Tabel 2.1. Perbandingan performansi AES vs Triple-DES

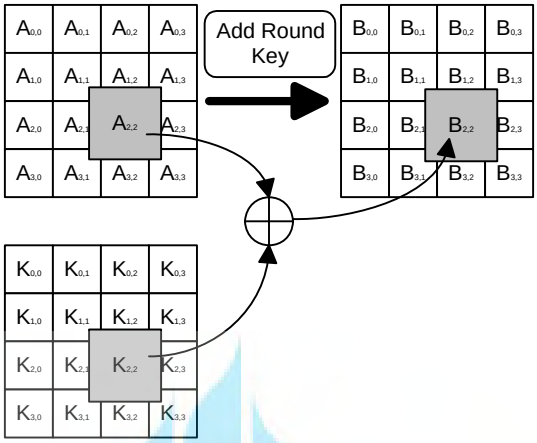
AES vs Triple-DES Item	AES	Triple-DES
Tipe Algoritma	Simetris, blok cipher	Simetris, feistel cipher
Key size [bits]	128, 192, 256	112, 168
Kecepatan	Tinggi	Rendah
Waktu crack [dengan asumsi 255 key/dt]	149 Milyarr Tahun	4,6 Trilyun Tahun
Konsumsi Resource	Rendah	Menengah

Pada dasarnya Rijndael tidak begitu mirip dengan AES karena pada Rijndael menggunakan size bit dan size key yang lebih besar sedangkan pada AES menggunakan bit yang fix yaitu 128 bit.

AES bekerja dengan standard Array 4x4 byte yang bekerja melalui empat tahap, yaitu:

1. **Add RoundKey**

   Add Round key adalah melakukan perhitungan logika XOR antara data yang akan di enkripsi dengan data kunci hingga di dapatkan data yang baru. Gambar 2.1



Gambar 2.3. Deskripsi langkah AddRoundKey

Selama ini para pemula selalu merasa "malas" ketika hendak mempelajari kriptografi. Berbagai alasan diungkapkan seperti terlalu kompleks, terlalu matematis dan puluhan alasan lain yang membuat malas untuk mempelajari kriptografi. Namun ada algoritma yang relatif gampang untuk dipelajari dan sudah dinyatakan oleh para ahli kriptografi sebagai *"perfect encryption algorithm"* yaitu *One Time Pad (OTP)* atau yang sering disebut sebagai *Vernam Cipher* karena ditemukan oleh Mayor J. Maugborne dan G. Vernam ditahun 1917.

Algoritma OTP merupakan algoritma berjenis symmetric key yang artinya bahwa kunci yang digunakan untuk melakukan enkripsi dan dekripsi merupakan kunci yang sama. Dalam proses enkripsi, algoritma ini menggunakan cara stream cipher dimana cipher berasal dari hasil XOR antara bit plaintext dan bit key.

Sebagai tambahan, algoritma *Vernam Cipher* sering digunakan dalam proses enkripsi pada pemrosesan transaksi online menggunakan kartu kredit karena prosesnya yang relatif mudah.

Beberapa hal yang menjadi persyaratan *Vernam Cipher* adalah:

1. Pemilihan kunci harus dilakukan secara acak agar tidak mudah diterka.

2. Jumlah karakter kunci harus sepanjang karakter plaintext.
3. Jika kunci tidak dapat diproduksi ulang maka algoritma dinyatakan aman

Konsep

Fungsi untuk mengenkripsi pesan hanyalah meng-XOR-kan plaintext dengan kunci yang telah disiapkan untuk menghasilkan ciphertext.

$$c = p \ XOR \ K$$

dimana :
c = Data yang telah dienkripsi
p = Data yang akan dienkripsi
K = Kunci Enkripsi

Sedangkan fungsi untuk mendekrip tinggal meng-XOR-kan ciphertext dengan kunciyang sudah disepakati

$$p = c \ XOR \ K$$

Contoh :

Data "RUSDI" disepakati memiliki kunci "CRASH", perlu diingat bahwa panjang karakter "p" harus sama dengan panjang kata kunci "K".

Tabel 2.2. Konversi karakter data

Karakter	ASCII	Biner
R	67	0100 0011
U	82	0101 0010
S	65	0100 0001
D	83	0101 0011
I	72	0100 1000

Tabel 2.3. Konversi karakter kunci

Karakter	ASCII	Biner
C	82	0101 0010
R	85	0101 0101
A	83	0101 0011
S	68	0100 0100
H	73	0100 1001

Kemudian "p" dan "K" akan di XOR:

Tabel 2.4. Hasil XOR Data dan Kunci

P	0100	0011	0101	0010	0100	0001	0101	0011	0100	1000
K	0101	0010	0101	0101	0101	0011	0100	0100	0100	1001
C	**0001**	**0001**	**0000**	**0111**	**0001**	**0010**	**0001**	**0111**	**0000**	**0001**

Maka pada saat sampai kepenerima, nilai “c” akan di XOR dengan “K” maka kembali akan menghasilkan kata”p” dan kembali di dapatkan kata “RUSDI”

Tabel 2.5. Hasil XOR Hasil dan Kunci

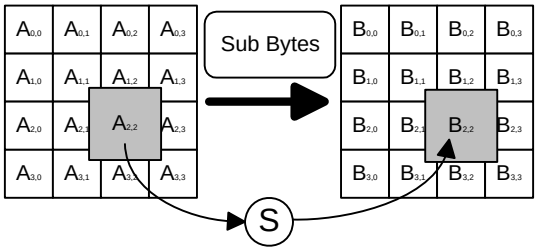
K	0101	0010	0101	0101	0101	0011	0100	0100	0100	1001
C	0001	0001	0000	0111	0001	0010	0001	0111	0000	0001
P	**0100**	**0011**	**0101**	**0010**	**0100**	**0001**	**0101**	**0011**	**0100**	**1000**

2. SubBytes

Pasa sistem ini, setiap bit di update dengan data baru yang menggunakan formula S-Box. S-Box akan mempermudah dalam proses transformasi non-linier dimana hasil akhir hexadecimal hanya mengacu pada S-Box.

Tabel 2.6. S-Box untuk mempermudah proses enkripsi AES

		Y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
X	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	63	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	A	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	B	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	C	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	D	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	E	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	F	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16



Gambar 2.4.Deskripsi langkah SubBytes

Sub byte merupakan transformasi substitusi non-linier byte yang beroperasi sendiri untuk tiap byte-nya dimana seperti dijelaskan diatas bahwan untuk

mempermudah proses perkaliat bit matrik maka dapat dilakukan dengan hanya melihat S-Box.

Secara matematis dapat dilihat pada formula berikut:

$$b'_i = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

Dalam bentuk matrik dapat dilihat sebagai berikut:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Untuk mempermudah perkalian dan penjumlahan matrik diatas maka dipakailah "S-Box" sebagai acuan matematis untuk bilangan hexadesimal. Sedangkan untuk proses deskripsi bantuan dari S-Box invers juga dapat dilakukan untuk mempermudah proses InverseSubByte. S-Box untuk proses Invers dapat dilihat pada tabel berikut:

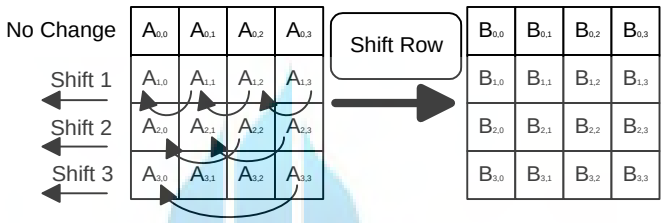
Tabel 2.7. Inverse S-Box

		Y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
X	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	A	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	B	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	C	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	D	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	E	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	F	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Sebagai contoh perhatikan jika kita mempunyai data desimal {53} maka dalam proses SubByte data akan berubah menjadi {ed} dan jika kita membalik bilangan hexadesimal tersebut dengan S-Box dari InverseSubByte didapatkan {53}

3. ShiftRows

Pada hasil enkripsi Sub Byte dilakukan proses shifting atau pergeseran 1 kali, 2 kali ataupun 3 kali dan seterusnya tergantung pada kesepakatan proses enkripsi sebelunya dimana setiap baris mendapatkan langkah pergeseran yang berbeda.

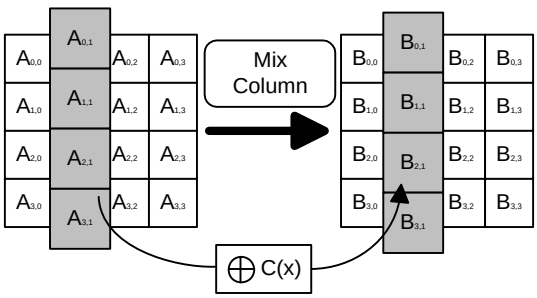


Gambar 2.5. Deskripsi langkah ShiftRows

4. MixColumn

Mixcolumn adalah proses perkalian matrix dimana setiap data pada colom akan mempengaruhi hasil data dari tiap data colom hasil perkalian matrik.

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$



Gambar 2.6. Deskripsi langkah MixColumns

Seperti dijelaskan sebelumnya bahwa mix kolom adalah transformasi colom demi colom dimana langsung melakukan proses perkalian matrik hexsadesimal satu kolom sekaligus untuk mendapatkan colom hasil enkripsi, dengan cara ini setiap data yang dihasilkan dalam satu kolom dipengaruhi oleh data dalam satu baris kolom sebelum di enkripsi. Perhatikan sistem matematis dibawah ini:

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$

Terlhat jelas bahwa setiap data akan saling mempengaruhi, secara matematis formula diatas dapat digambarkan sebagai berikut :

$$S'_{0,c} = (\{02\} \bullet S_{0.c}) \oplus (\{03\} \bullet S_{1,c}) \oplus S_{2,c} \oplus S_{3,c}$$
$$S'_{1,c} = S_{0,c} \oplus (\{02\} \bullet S_{1,c}) \oplus (\{03\} \bullet S_{2,c}) \oplus S_{3,c}$$
$$S'_{2,c} = S_{0,c} \oplus S_{1,c} \oplus (\{02\} \bullet S_{2,c}) \oplus (\{03\} \bullet S_{3,c})$$
$$S'_{3,c} = (\{03\} \bullet S_{0,c}) \oplus S_{1,c} \oplus S_{2,c} \oplus (\{02\} \bullet S_{3,0})$$

Perhatikan bahwa tanda • “dot” adalah perkalian polinomial bukanlah perkalian biasa dengan batasan 2^8. Ingat bahwa :

$2^8 = 256$

$2^7 = 128$

$2^6 = 64$

$2^5 = 32$

$2^4 = 16$

$2^3 = 8$

$2^2 = 2$

$2^0 = 1$

Contoh:

Jika kita mempunya bilangan hexadesimal {57} dan {83} maka langkah yang harus dilakukan untuk peralian {57}• {83} adalah:

Ubah 57 ke bilangan desimal = 87

Bilangan pemangkatan 2 yang terdekat adalah 2^6 yaitu 64

$87 – 64 = 23$

Untuk bilangan 23, bilangan pemangkatan terdekat adalah 2^4 yaitu 16

$23 – 16 = 7$

Untuk bilangan 7, bilangan pemangkatan terdekat adalah 2^2 yaitu 4

$7 – 4 = 3$

Untuk bilangan 3, bilangan pemangkatan terdekat adalah 2^1 yaitu 2

$3 – 2 = 1$

Untuk bilangan 1, bilangan pemangkatannya adalah 2^0 yaitu 1

Sehingga didapatkan hasil

$$87 = 2^6 + 2^4 + 2^2 + 2^1 + 2^0$$

Atau untuk keperluan perkalian AES :

$$87 = x^6 + x^4 + x^2 + x + 1$$

Hal yang sama dapat kita lakukan pada angka hexadesimal 83h yaitu 131, yang menghasilkan :

$$131 = x^7 + x + 1$$

Kemudian hasil 87 dan 131 dikalikan menjadi:

$(x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1) = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$

Kemudian di "Mod" dengan $(x^8 + x^4 + x^3 + x +1)$

Menghasilkan = $x^7 + x^6 + 1$, dimana hasil diatas jika dijabarkan menghasilkan angka desimal 193 atau dalam bentuk hexadesimal adalah {c1}:

Maka :

$$\{57h\} \bullet \{83h\} = \{c1\}$$

Satu hal lagi yang perlu diperhatikan adalah tanda $\oplus$ adalah "XOR" atau exclesive OR bukan penjumlahan biasa.

Sedangkan dalam proses invernya (InverseMixColumn) dapat dilakukan dengan rumus perkalian matrik berikut :

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} = \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$

Dengan adanya bahasa pemorgraman maka setiap administrator ataupun pengguna jaringan telekomunikasi dapat membuat sendiri sistem keamanan data khususnya proses enkripsi tanpa harus memakai sistem enkripsi default pada program / sistem jaringan.

Sebagai contoh enkripsi data dengan menggunakan AES 10 kali adalah sebagai berikut:

Input = 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
Cipher Key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

Round	Awal Ronund				Setelah SubByte				Setelah Shift Row				Setelah MixColumn				Hasil Round			
input	32	88	31	e0													2b	28	ab	9
	43	5a	31	37													7e	ae	f7	cf
	f6	30	98	7													15	d2	15	4f
	a8	8d	a2	34													16	a6	88	3c
1	19	a0	9a	e9	d4	e0	b8	1e	d4	e0	b8	1e	4	e0	48	28	a0	88	23	2a
	3d	f4	c6	f8	27	bf	b4	41	bf	b4	41	27	66	cb	f8	6	fa	54	a3	6c
	e3	e2	8d	48	11	98	5d	52	5d	52	11	98	81	19	d3	26	fe	2c	39	76
	be	2b	2a	8	ae	f1	e5	30	30	ae	f1	e5	e5	9a	7a	4c	17	b1	39	5
2	a4	68	6b	2	49	45	7f	77	49	45	7f	77	f2	7a	59	73	58	1b	db	1b
	9c	9f	5b	6a	de	db	39	2	db	39	2	de	c2	96	35	59	4d	4b	e7	6b
	7f	35	ea	50	d2	96	87	53	87	53	d2	96	95	b9	80	f6	ca	5a	ca	b0
	f2	2b	43	49	89	f1	1a	3b	3b	89	f1	1a	f2	43	7a	7f	f1	ac	a8	e5
3	aa	61	82	68	ac	ef	13	45	ac	ef	13	45	3d	47	1e	6d	75	20	53	bb
	8f	dd	d2	32	73	c1	b5	23	c1	b5	23	73	80	16	23	7a	ec	0b	c0	25
	5f	e3	4a	46	cf	11	d6	5a	d6	5a	cf	11	47	fe	7e	88	9	63	cf	d0
	3	ef	d2	9a	7b	df	b5	b8	b8	7b	df	b5	7d	3e	44	3b	93	33	7c	dc
4	48	67	4d	d6	52	85	e3	f6	52	85	e3	f6	ef	a8	b6	db	0f	60	6f	5e
	6c	1d	e3	5f	50	a4	11	cf	a4	11	cf	50	44	52	71	0b	d6	31	c0	b3
	4e	9d	b1	58	2f	5e	c8	6a	c8	6a	2f	5e	a5	5b	25	ad	da	38	10	13
	ee	0d	38	e7	28	d7	7	94	94	28	d7	7	41	7f	3b	0	a9	bf	6b	1
5	0f	60	6f	5e	e1	e8	35	97	e1	e8	35	97	25	bd	b6	4c	d4	7c	ca	11
	d6	31	c0	b3	4f	fb	c8	6c	fb	c8	6c	4f	d1	11	3a	4c	d1	83	f2	f9
	da	38	10	13	d2	fb	96	ae	96	ae	d2	fb	a9	d1	33	c0	c6	9d	b8	15
	a9	bf	6b	1	9b	ba	53	7c	7c	9b	ba	53	ad	68	8e	b0	f8	87	bc	bc

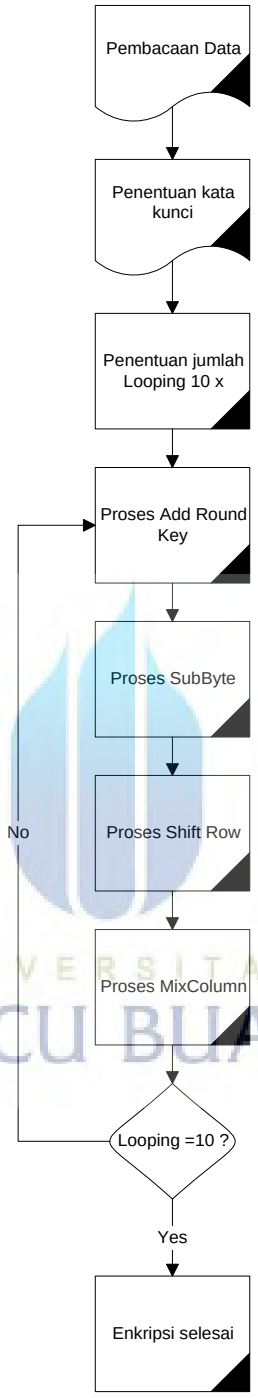
6	f1	c1	7c	5d	a1	78	10	4c	a1	78	10	4c	4b	2c	33	37	6d	11	db	ca
	0	92	c8	b5	63	4f	e8	d5	4f	e8	d5	63	86	4a	9d	d2	88	0b	f9	0
	6f	4c	8b	d5	a8	29	3d	3	3d	3	a8	29	8d	89	f4	18	a3	3e	86	93
	55	ef	32	0c	fc	df	23	fe	fe	fc	df	23	6d	80	e8	d8	7a	fd	41	fd
7	26	3d	e8	fd	f7	27	9b	54	f7	27	9b	54	14	46	27	34	4e	5f	84	4e
	0e	41	64	d2	ab	83	43	b5	ab	83	43	b5	15	16	46	2a	54	5f	a6	a6
	2e	b7	72	8b	31	a9	40	3d	31	a9	40	3d	b5	15	56	d8	f7	c9	4f	dc
	17	7d	a9	25	f0	ff	d3	3f	f0	ff	d3	3f	bf	ec	d7	43	0e	f3	b2	4f
8	5a	19	a3	7a	be	d4	0a	da	be	d4	0a	da	ea	b5	31	7f	0	b1	54	fa
	41	49	e0	8c	83	3b	e1	64	3b	e1	64	83	d2	8d	2b	8d	51	c8	76	1b
	42	dc	19	4	2c	86	d4	f2	d4	f2	2c	86	73	ba	f5	29	2f	89	6d	99
	b1	1f	65	0c	c8	c0	4d	fe	fe	c8	c0	4d	21	d2	60	2f	d1	ff	cd	ea
9	ea	4	65	85	87	f2	4d	97	87	f2	4d	97	47	40	a3	4c	ac	19	28	57
	83	45	5d	96	ec	6e	4c	90	6e	4c	90	ec	37	d4	70	9f	77	fa	d1	5c
	5c	33	98	b0	4a	c3	46	e7	46	e7	4a	c3	94	e4	3a	42	66	dc	29	0
	f0	2d	ad	c5	8c	d8	95	a6	a6	8c	d8	95	ed	a5	a6	bc	f3	21	41	6e
10	eb	59	8b	1b	eb	59	8b	1b	e9	cb	3d	af					d0	c9	e1	b6
	40	2e	a1	c3	40	2e	a1	c3	31	32	2e	9					14	ee	3f	63
	f2	38	13	42	f2	38	13	42	7d	2c	89	7					f9	25	0c	0c
	1e	84	e7	d2	1e	84	e7	d2	b5	72	5f	94					a8	89	c8	a6
Hasil	**39**	**2**	**dc**	**19**																
	25	**dc**	**11**	**6a**																
	84	**9**	**85**	**0b**																
	1d	**fb**	**97**	**32**																

Hasil Enkripsi adalah :

39 25 84 1d 02 dc 09 fb dc 11 85 97 19 6a 0b 32

Untuk mendapatkan setiap hasil enkripsi diatas maka dengan pemrobraman komputer akan dapat diselesaikan dengan mudah karena terlihat jelas perulangan 10 tahap maka program komputer tidak perlu membuat 10 kali formula yang sama akan tetapi cukup dengan metode pengulangan (Looping) program maka sistem pengulangan 10 kali dapat dilakukan dengan sangat mudah.

Flow chart Algoritma adalah sebagai berikut :



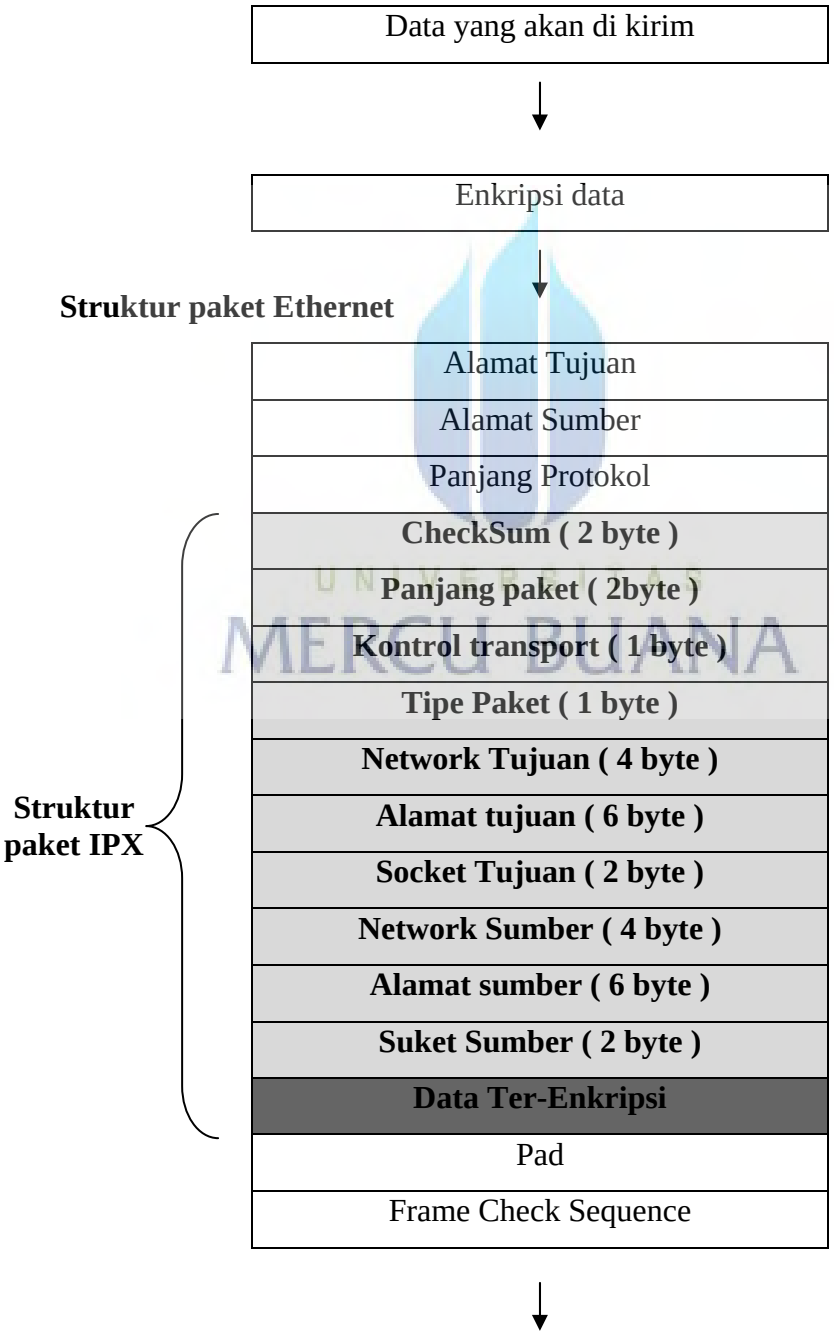
Gambar 2.7. Flowchart Enkripsi Umum

Protokol pengiriman data pada jaringan

Dalam proses enkripsi tidak diperlukan adanya protokol khusus dalam mengirim data. Hal ini disebabkan karena proses enkripsi dilakukan sebelum data dimasukkan dalam paket data IPX (Internet Protocol Exchange) begitu pula sebaliknya dalam proses deskripsi pada sisi penerima dilakukan setelah proses penerimaan paket data.

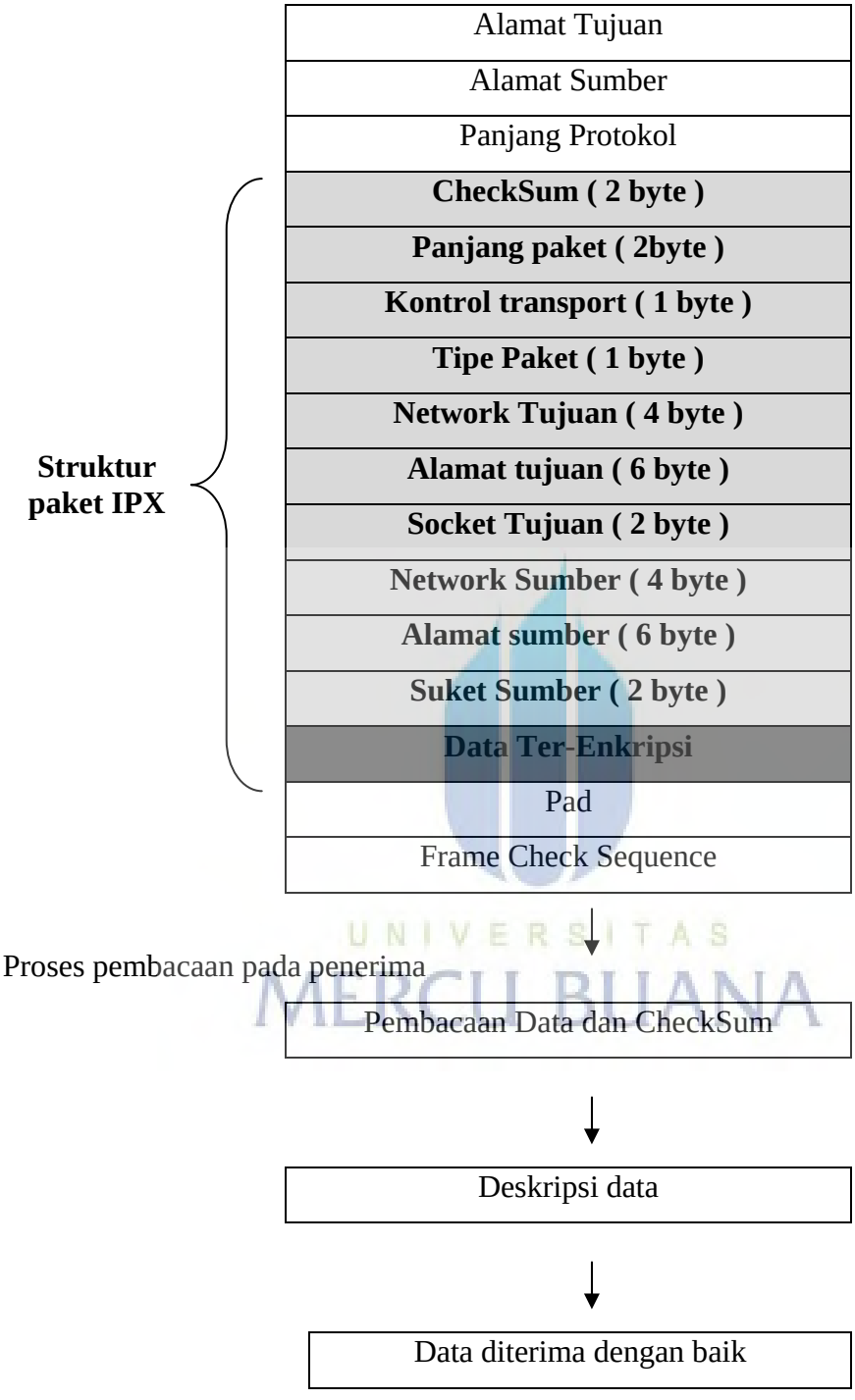
Untuk lebih jelasnya dapat dilihat dalam bagan berikut :

Sisi pengirim



Sisi penerima :

Struktur paket Ethernet



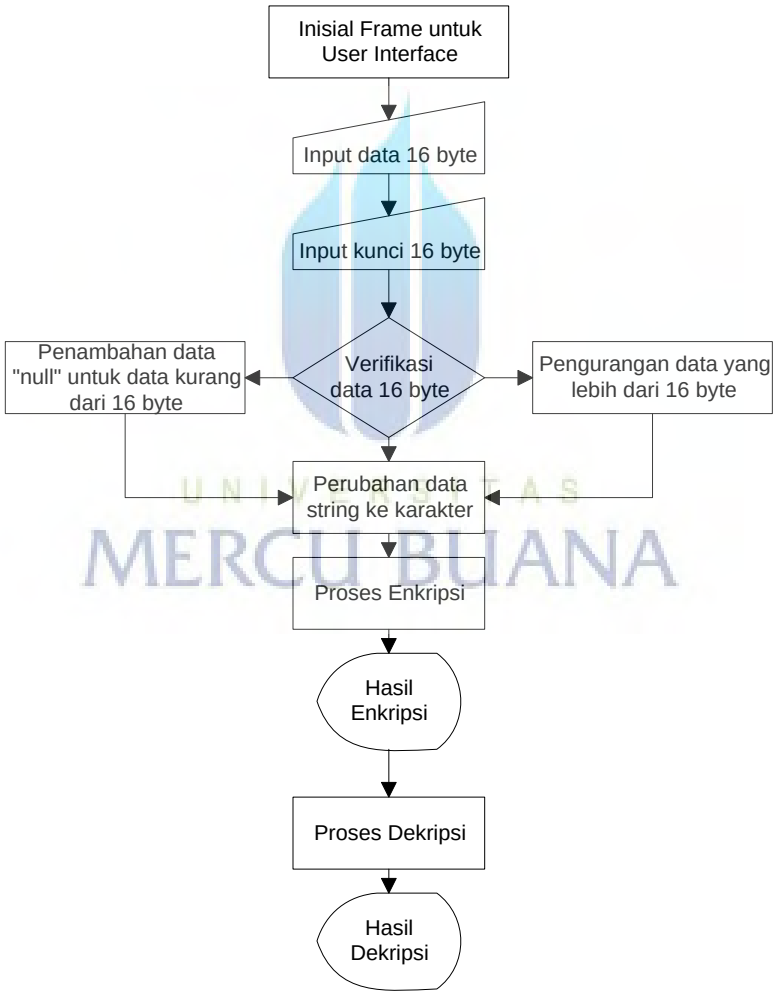
Proses pembacaan pada penerima

Terlihat bahwa proses enkripsi tidak ada pada proses protokol pengiriman data namun pada saat data terkirim bentuk data sudah teracak hingga pada saat ada orang lain yang berniat mengambil data pada saat pengiriman maka data yang diterima dalam keaadan acak. Perlu diingat bahwa proses enkripsi tidak memperlambat proses pengiriman namun menambah waktu dalam pembuatan paket data.

BAB III
PERANCANGAN

III.1 Perancangan flow chart proses enkripsi data.

Pada perancangan program penulis membagi dalam beberapa file untuk mendefinisikan class-class atau prosedur program untuk dipakai pada program utama. Secara umum flow chart pemrograman adalah sebagai berikut :



Gambar 3.1. Flow proses program simulasi Enkripsi dan Dekripsi

Pada start program pertama kali maka program utama akan mengisialsasi Grafik User Interface untuk memudahkan user untuk menjalankan program.Pada program utama tersebut user akan diminta untuk memasukkan data 16 byte atau 16buah karakter sebagai data yangakan diolah dan 16 buah karakter yang digunakan sebagai kata kunci dalam proses enkripsi.

Secara automatis program akan membaca data tersebut dan ketika terjadi kesalahan input seperti data yang dimasukkan kurang dari 16 karakter maka program secara automatis akan menambahkan nilai “null” untuk melengkapi 16 karakter, sedangkan jika data yang dimasukkan lebih dari 16 karakter maka program pun secara automatis hanya mengambil 16 karakter pertama yang diinput. Sistem ini sengaja dibuat agar kesalahan yang mungkin terjadi pada user dapat di antisipasi oleh program.

Setelah data dan kata kunci telah direkam dalam varibel program maka data-data mulai diolah dalam bentuk enkripsi 10 tahap dan begitu pula sebaliknya data-data hasi enkripsi juga akan diolah 10 tahap dekripsi untuk menampilkan kembali data yang pertama kali di masukkan oleh user.

III.2 Perancangan algoritma pemrograman.

Untuk mempermudah dan mempercepat kinerja suatu software maka sebuah program sebaiknya di bagi dalam beberapa class / library yang berfungsi untuk memperkecil kapasitas setiap file yang dipergunakan dalam menjalankan program, dan juga berfungsi untuk mempercepat kinerja software dikarenakan sebuah program utama / main file tidak boleh terlalu besar kapasitasnya disebabkan pada saat pertama kali menjalankan program maka “main program” tersebut akan di masukkan dalam memory komputer. Maka jika sebuah software bekerja pada kapasitas memory komputer yang minim sedangkan menggunakan main software yang cukup besar maka akan sangat memberatkan sekali proses logika yang dijalankan suatu program.

Dalam perancangan suatu software diperlukan algoritma program terlebih dahulu sehingga langkah pembuatan software akan lebih terarah. Pada tugas akhir ini, algoritma yang dipergunakan sebagai berikut :

File I : *Frame program*

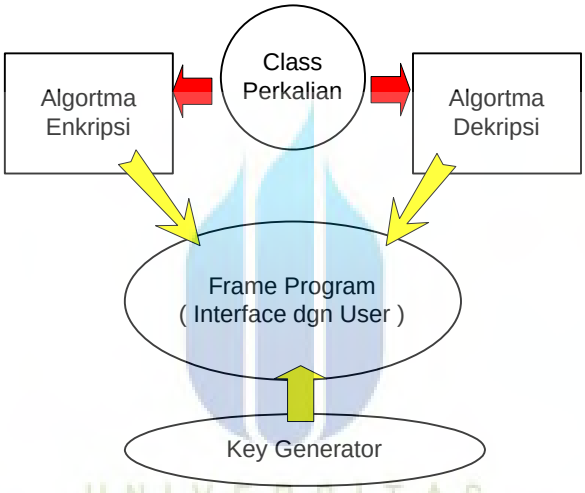
File II : *File Enkripsi*

File III : *File Dekripsi*

File IV : *File Perkalian Hexadecimal*

File V : *File Password / Key Generator*

Untuk lebih jelasnya sinkronisasi kelima file tersebut dapat dilihat pada blok diagram berikut :



Gambar 3.2. Bagan alur program simulasi enkripsi dan dekripsi

Terlihat pada bagan diatas bahwa empat buah file saling bekerja sama untuk memenuhi permintaan user yang dibuat dalam frame program. Untuk lebih jelasnya algoritma program adalah sebagai berikut.

File I (Frame Program)

File ini dimulai dengan inisialisasi paket program dan mendefinisikan variabel-variabel public yang akan dipakai :

```
package JavaAES;
import JavaAES.Enkripsi.*;
public class AES_Frame extends javax.swing.JFrame{
   /** Creates new form AES_Frame */
```

```
char[ ] hasilmixcolumn = new char[16];
char[ ] hasilshiftrows = new char[16];
char[ ] hasilsubbytes = new char[16];
char[ ] hasiladdround = new char[16];
char[ ] hasilenkripsi = new char[16];
```

Diawali dengan mengumumkan paket program yaitu “JavaAES” yang ditempatkan dalam sebuah folder, kemudian program akan mendefinisikan class-class yang akan dipakai adalah seluruh class dalam folder “C:/ ….JavaAES/Enkripsi..”

Oleh karena terdapat beberapa variabel yang akan secara terus menerus dipakai dalam berbagai prosedur / class program maka variabel-variabel tersebut didefinisikan diawal dengan bentuk “public” yang bisa di akses class manapun.

```
public AES_Frame() {
    initComponents();
    setLocation(250,150);
    setSize(790,400);
    setResizable(false);
  }
```


Kemudian frame program di definisikan mulai dari koordinat lokasi pada monitor, ukuran yang diinginkan dan definisi apakah program tersebut dapat diperbesar atau diperkecil.Selebihnya program akan menginisialisasi isi dari frame termasuk button/tombol, label, dan text.

Kemudian dalam beberapa “event” penekanan tombol pada program terdapat beberapa baris program untuk menjalankan proses enkripsi dan dekripsi. Listing program lebih lengkap dapat dilihat pada lampiran.

File II (Enkripsi)

Deklarasi procedure Enkripsi dibagi dalam beberapa prosedur yaitu :

a. Prosedur **“Ubah ke Char”** yang mengubah bentuk String kedalam bentuk 16 karakter yang tergabung dalam sebuah variabnel array.

b. Prosedur **“Konversi”** yang akan mengubah karakter dalam sebuah varibel array kedalam bentuk hexadesimal.
c. Prosuder **“Add Round Key”** yang menjalankan salah satu tahap enkripsi yaitu add round.
d. Prosedur **“Sbox”** yang melakukan tahap sub bytes dalam proses enkripsi.
e. Prosedur **“Shift Row”** yang melakukan tahap shift row dalam proses enkripsi.
f. Prosedur **“Mix Column”** yang akan melakukan tahap mix column pada proses enkripsi.

Prosedur Ubah ke Char

1. Pengambilan data yang diubah ke bentuk karakter array
2. Pengecekan apakah data lebih besar dari 16 karakter, jika lebih dari 16 maka hanya 16 karakter yang akan diambil.
3. Pengecekan apakah data lebih kecil dari 16 karakter, jika lebih kecil maka data akan ditambahkan nilai “null” untuk mencukupkan 16 data karakter.

Flow chart dibawah menggambarkan cara kerja prosedur Ubah ke Char

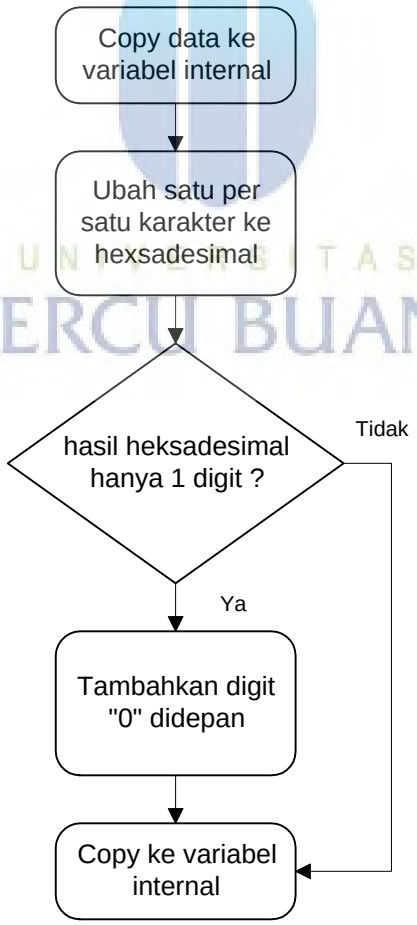
Gambar 3.3 Flow chart prosedur ubah ke chart

Prosedur Konversi

1. Copy data ke variabel internal
2. Ubah satu persatu data ke bentuk hexadesimal.
3. Jika nilai hexadesmal hanya satu digit maka diberikan angka nol didepannya.

```
StringBuffer HasilKonversi = new StringBuffer();
    for( int i=0; i<16; i++)
    {
    String d1 = String.valueOf(data16[i] & 0xff);
    String d2 = Integer.toHexString(Integer.parseInt(d1));
    StringBuffer DataAdd = new StringBuffer();
    DataAdd.append(d2);
    if(d2.length()<2)
    {
       DataAdd.insert(0,"0");
    }
    HasilKonversi.append(DataAdd.toString());
    }
```

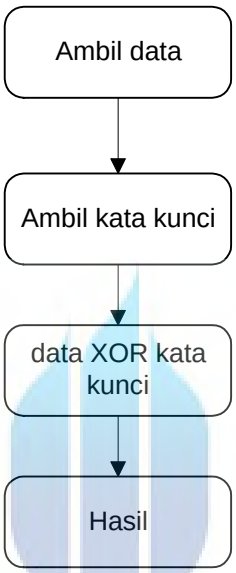
Lebih jelasnya dapat dilihat pada flow chart berikut :



Gambar 3.4. Flow Chart prosedur konversi

Prosedur Add Round Key

1. Ambil data dan copy ke variabel data internal
2. Ambil kata kunci dan copy ke variabel internal
3. Jumlahkan satu persatu data denganlogika XOR
4. Kembalikan hasil Xor ke varibel hasil.



Gambar 3.5. Flowchart prosedur Add round key

Prosedur Sbox

1. Copy data ke variabel internal
2. Ambil data lalu ubah ke bentuk hexadesimal
3. Tentukan koordinat hexadesimal pada SBox
4. Cari hasil Sbox pada tabel
5. Kembalikan data ke variabel hasil

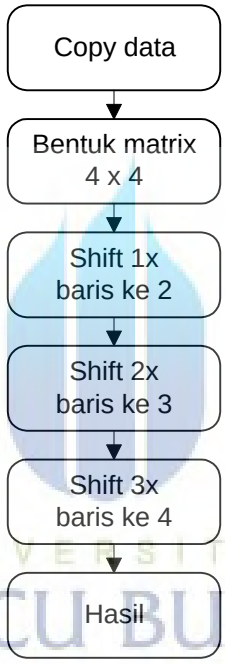
```
d2 = Integer.toHexString(Integer.parseInt(satudata));
            StringBuffer DataAdd = new StringBuffer();
            DataAdd.append(d2);
            if(d2.length()<2)
            {
               DataAdd.insert(0,"0");
            }
            hexa.append(DataAdd.toString());
            char arrayrow,arraycolom;
            arrayrow = hexa.charAt(0);
            arraycolom = hexa.charAt(1);
            int baris1,baris2;
            caribariskolom baris = new caribariskolom();
            baris1 = baris.bariskolom(arrayrow);
            baris2 = baris.bariskolom(arraycolom);
            char sboxawal;
            sboxawal = sbox[baris1][baris2];
            Hasil.append(sboxawal);
        }
```

Gambar 3.6. Flowchart prosedur Sub bytes

Prosedur Shift Row

1. Copy data ke varibel internal
2. Bentuk matrik 4 x 4
3. Shift satu kali baris ke dua
4. Shift dua kali baris ke tiga
5. Shift tiga kali baris ke empat.
6. Kembalikan hasil ke varibel hasil

Pada flow chart berikut memperlihatkan tahap dari baris program:



Gambar 3.7. Flowchart prosedur shift row

Prosedur Mix Column

1. Copy data ke variabel internal
2. Lakukan perkalian hexadesimal dengan rumus yang sudah di tentukan

Listing program dapat dilihat sebagai berikut :

```
for(int i=0;i<16;i+=4)
    {
        arrayhasil[i] = (char)(kali.mul(arraydata[i] & 0xff,0x02)^kali.mul(arraydata[i+1]&
                        0xff,0x03)^(arraydata[i+2] & 0xff)^(arraydata[i+3] & 0xff));
        arrayhasil[i+1] = (char)((arraydata[i] & 0xff)^kali.mul(arraydata[i+1]&
                          0xff,0x02)^kali.mul(arraydata[i+2]& 0xff, 0x03)^(arraydata[i+3] &
                          0xff));
```

```
            arrayhasil[i+2] = (char)((arraydata[i] & 0xff)^(arraydata[i+1] &
                        0xff)^kali.mul(arraydata[i+2]& 0xff, 0x02)^kali.mul(arraydata[i+3]&
                        0xff, 0x03));
            arrayhasil[i+3] = (char)(kali.mul(arraydata[i]& 0xff, 0x03)^(arraydata[i+1] &
                        0xff)^(arraydata[i+2] & 0xff)^kali.mul(arraydata[i+3]& 0xff, 0x02));

        }
        return arrayhasil;
```

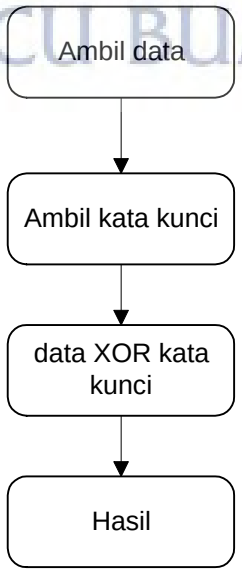
File III (Dekripsi)

Deklarasi procedureDekripsi dibagi dalam beberapa prosedur yaitu :

a. Prosedur **"Inv Add Round Key"** yang melakukan inversi dari proses Add Round Key.
b. Prosedur **"Inv Sbox"** yang melakukan proses inversi Sub Bytes.
c. Prosedur **"Inv Shift Row"** yang melakukan inversi pada proses Shift Row.
d. Prosedur **"Inv Mix Column"** yang melakukan inversi proses Mix Column

Prosedur Inversi Add Round Key

1. Ambil data dan copy ke variabel data internal
2. Ambil kata kunci dan copy ke variabel internal
3. Jumlahkan satu persatu data denganlogika XOR
4. Kembalikan hasil XOR ke varibel hasil.



Gambar 3.8. Flowchart prosedur Inv Add round key

Jika diperhatikan secara seksama algoritma pemrograman Inversi Add round key sama persis dengan proses add round key dalam proses enkripsi, hal ini disebabkan hasil XOR jika dilakukan kembali hasil XOR dari hasil dan nilai pasangan XOR maka akan menghasilkan nilai sebelumnya. Perhatikan ilustrasi berikut :

A = B XOR C

B = A XOR C

Prosedur Inv Sbox

1. Copy data ke variabel internal
2. Ambil data lalu ubah ke bentuk hexadesimal
3. Tentukan koordinat hexadesimal pada SBox
4. Cari hasil Sbox pada tabel
5. Kembalikan data ke variabel hasil

Perhatikan kembali bahwa listing program sama persis dengan proses sub bytes pada proses enkripsi, yang membedakan hanya referensi tabelnya yang mengacu pada tabel InvSbox.

Prosedur Inv Shift Row

1. Copy data ke varibel internal
2. Bentuk matrik 4 x 4
3. Shift satu kali ke belakang baris ke dua
4. Shift dua kali ke belakang baris ke tiga
5. Shift tiga kali ke belakang baris ke empat.
6. Kembalikan hasil ke varibel hasil

Kembali algoritma hampir sama dengan algoritma Shift Row pada proses enkripsi yang membedakan hanya proses sifting dibalik arahnya.

Prosedur Inv Mix Column

1. Copy data ke variabel internal
2. Lakukan perkalian hexadesimal dengan rumus yang sudah di tentukan

Listing program dapat dilihat sebagai berikut :

```
for(int i=0;i<16;i+=4)
      {
            arrayhasil[i] = (char)(kali.mul(arraydata[i] & 0xff,0x0e)^kali.mul(arraydata[i+1]& 0xff,0x0b)^kali.mul(arraydata[i+2] & 0xff,0x0d)^kali.mul(arraydata[i+3] & 0xff,0x09));
            arrayhasil[i+1] = (char)(kali.mul(arraydata[i] & 0xff,0x09)^kali.mul(arraydata[i+1]& 0xff,0x0e)^kali.mul(arraydata[i+2]& 0xff, 0x0b)^kali.mul(arraydata[i+3] & 0xff,0x0d));
            arrayhasil[i+2] = (char)(kali.mul(arraydata[i] & 0xff,0x0d)^kali.mul(arraydata[i+1] & 0xff,0x09)^kali.mul(arraydata[i+2]& 0xff, 0x0e)^kali.mul(arraydata[i+3]& 0xff, 0x0b));
            arrayhasil[i+3] = (char)(kali.mul(arraydata[i]& 0xff, 0x0b)^kali.mul(arraydata[i+1] & 0xff,0x0d)^kali.mul(arraydata[i+2] & 0xff,0x09)^kali.mul(arraydata[i+3]& 0xff, 0x0e));

      }
      return arrayhasil;
```

Pada listing program diatas dapat terlihat bahwa yang membedakan antara proses enkripsi Mix Column dengan proses dekripsi Inv Mix Column hanyalah formula yang dipergunakan dalam proses perkalian.

File IV (Class Perkalian)

Prosedur perkalian hexadesimal di buat untuk membantu proses mix column dan inv mix column dalam algoritma enkripsi dan dekripsi. Untuk mempermudah proses perkalian digunakan dua buah table perkalian hexadesimal yaitu Logtable dan Antilogtable.

Algoritma perkalian tersebut adalah sebagai berikut :

1. Deklarasi LogTable perkalian
2. Deklarasi AntiLog perkalian
3. Class operasi Multiplikasi

Listing programnya dapat dilihat sebagai berikut :

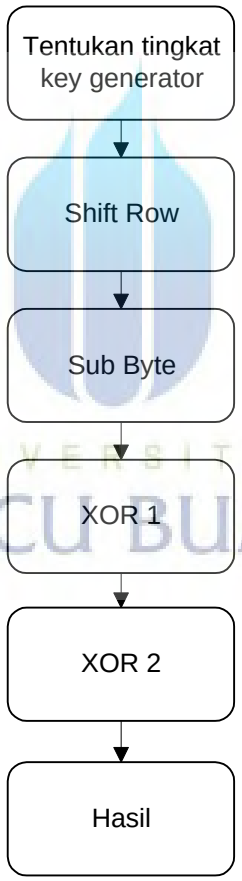
```
public char mul(int a, int b)
   {
      int hasil;
      char hasilkali;
      hasil = (a != 0 && b != 0) ? AntilogTable[(LogTable[a & 0xFF] + LogTable[b & 0xFF]) % 255] : 0;
      hasilkali = (char)(hasil);
      return hasilkali;
   }
```

File V (Key Generator)

Prosedur key generator mendefinisikan tahap key generator yang diinginkan oleh user dari tahap 1 sampai sepuluh.

1. Deklarasi pilihan 10 tingkat pengacakan kata kunci
2. Logika empat tahap pengacakan kata kunci

Flow char pemrograman adalah sebagai berikut :



Gambar 3.9. Flowchart prosedur key generator

Pada bab sebelumnya telah dijelaskan pengertian dan formula dari empat tahap enkripsi yaitu SubBytes, Shift Rows, Add Round dan Mixcolumn. Untuk

memformulasikan empat tahap tersebut dalam bentuk algoritma pemrograman java maka tahap-tahap yang dilakukan adalah sebagai berikut.

SubBytes yang memiliki formula matematika GF (2^8) akan sangat rumut jika di tuangkan dalam bentuk algoritma pemrograman, maka untuk lebih mempermudah proses pembuatan dipakai tabel Sbox dimana sebelumnya setiap kemungkinan proses hexadesimal yang akan diolah dalam program dihutung terlebih dahulu untuk kemudian dituliskan dalam bentuk table x,y yang masing masing mewakili 16 angka heksadesimal (0 sampai F). Dengan tabel ini setiap kali akan melakukan transformasi SubBytes cukup merujuk pada table tersebut. Hal inipun dilakukan pada proses Inverse SubBytes.

Shift Rows merupakan penggeseran data biasa yang memiliki tiga tingkat penggeseran sehingga dalam algoritma pemrograman cukup dilakukan penyimpanan satu data dalam varibel temporary untuk kemudian di tempatkan pada area penggeserannya.

AddRound Key merupakan proses XOR biasa antar data dengan kata kunci.

MixColumn merupakan sebuah sistem perkalian heksadesimal dalam bentuk polinomial GF(2^8), proses ini tidak sama dengan proses SubByte yang bisa dituangkan dalam bentuk table, namun persamaan polinomial harus diformulasikan dalam bentuk algoritma pemrograman. Pada perkalian dengan menngunakan persamaan pemangkatan dan modulasi pada formula matematik akan sangat menyulitkan sekali hingga algoritma pemrograman sebaiknya dilakukan dalam bentuk binner. Perkalian dalam bentuk binner akan lebih mempermudah proses perkalian karena pada proses perkalian dan pembagian manual pada angka desimal digunakan proses penjumlahan desimal namun pada proses perkalian dan pembagian dalam bentuk binner proses penjumlahan dan pembagian memakai formula XOR.

Contoh :

Perkalian desimal : 23 X 25

Akan dijabarkan :

```
  23
  25 x
 115
 46  +
 575
```

Perhatikan bahwa ada proses penambahan dalam proses perkalian

Pada **perkalian heksadesimal** digunakan metode sebagai berikut :

{53} • {CA} = {01} dalam proses Rijndael

$(x^6 + x^4 + x + 1)\ (x^7 + x^6 + x^3 + x) =$

$\Leftrightarrow x^{13} + x^{12} + x^9 + \mathbf{x^7} + x^{11} + x^{10} + \mathbf{x^7} + x^5 + x^8 + \mathbf{x^7} + x^4 + x^2 + \mathbf{x^7} + x^6 + x^3 + x =$

$\Leftrightarrow x^{13} + x^{12} + x^9 + x^{11} + x^{10} + x^5 + x^8 + x^4 + x^2 + x^6 + x^3 + x =$

$\Leftrightarrow x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x$

Kemudian

$x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x$ modulo $x^8 + x^4 + x^3 + x + 1$ menghasilkan nilai hexa {01}.

Cara diatas akan sangat murumitkan dalam proses pembuatan program maka proses diatas diuah ke bentuk binner :

{53} = 0101 0011

{CA} = 1100 1010

```
 01010011
 11001010 x
 01010011
  01010011
    00000000
     00000000
      01010011
       00000000
        01010011
      ^  00000000
      ---------
      0111111110111110
```

Hasil diatas akan di mod dengan bilangan 100011011, menjadi :

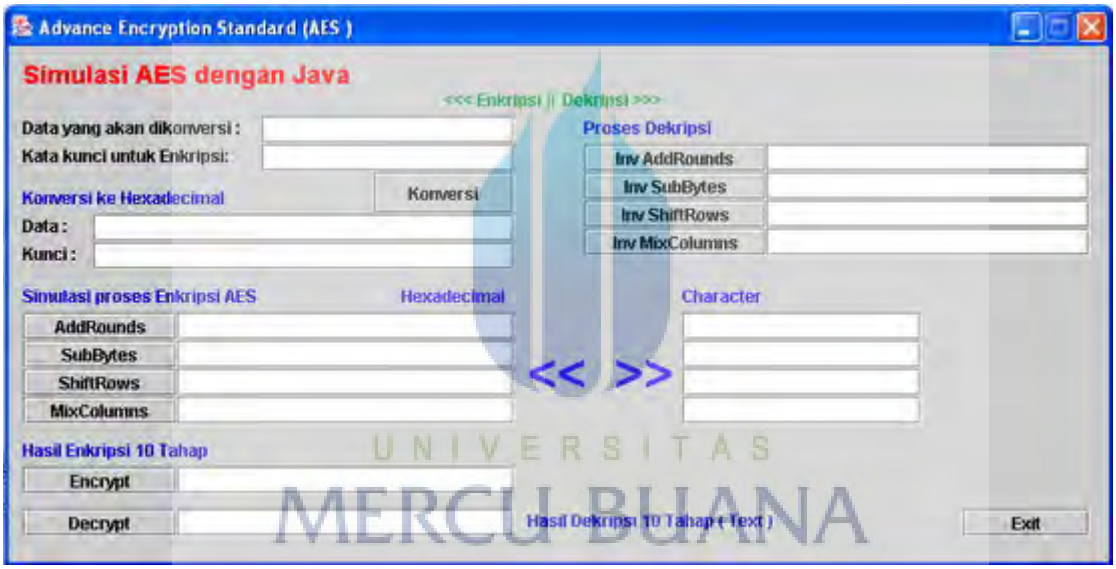
```
                    111101
---------------------------
100011011)11111101111110
          ^100011011
           ----------
            1110000011110
           ^100011011
            ----------
             110110101110
            ^100011011
             ----------
              10101110110
             ^100011011
              ----------
               0100011010
              ^000000000
               ----------
                100011010
               ^100011011
                ----------
                 00000001
```

Terlihat bhwa hasil akhir yang didapat adalah binner 00000001 = {01}

Sistem perkalian inilah yang dipakai dalam algoritma pemrograman perkalian MixColumn.

III.3 Performansi hasil pemrograman.

Simulasi pemorograman ditampilkan pada dua aspek yaitu Enkripsi dan Dekripsi. Pada bagian lain dari program disimulasikan bagaimana tahap demi taham enkripsi dan dekripsi terbentuk. Pada user interface program dibawah terdapat beberapa label input diantaranya:



Gambar.3.10. Tampilan Interface program

Data yang akan diinput adalah tempat user mengisi data yang akan di enkripsi

Kata kunci untuk Enkripsi adalah tempat user mengisi kata kunci yang dinginkan.

Terdapat **tombol konversi** untuk memperjelas kepada user bagaimana proses enkripsi terjadi maka data dan kunci di ubah menjadi bentuk heksadesimal

Data yanng dimasukkan oleh user akan diubah kebentuk heksa desimal dalam variabel karakter, kemudian dilakukan perulangan 16 kali untuk mendapatkan setiap byte data. Khusus untuk hasil konversi bernilai satu digit akan ditambahkan angka “nol” didepannya agar user masih dapat melihat 16 data heksadesimal.

Tombol AddRound adalah proses XOR data dengan Kunci dalam satu tahap. Data dan kunci diambil satu persatu kemudian dilakukan proses XOR selama 16 kali kemudian hasilnya ditampilkan pada label add round yang masih dalam bentuk heksadesimal.
Proses ini juga dilakukan pada subbytes, shiftrow dan mixcolumn yang dapat dilihat listing program lebih lengkapnya pada lampiran.

Kemudian terdapat tombol Enkripsi yang akan mengekripsi secara lengkap proses AES 16 tahap dari data dan kata kunci yang di input oleh user, output yang dihasilkan dalam bentuk heksadesimal agar user dapat melihat perubahan mendasar dari data ke hasil enrkripsi.

Tombol enkripsi pada bagian bawah program simulasi berguna untuk mengembalikan hasil enkripsi ke data yang sebenarnya dengan asumsi bahwa proses ini terjadi pada sisi penerima data.

BAB IV
IMPLEMENTASI

IV.1 Hasil uji implementasi AES pada Jaringan.

Setelah melakukan beberapa test terhadap program enkripsi dengan memakai VPN3015 dengan tujuan untuk mengatehui kemampuan enkripsi AES terhadap DES yang diterapkan pada workstation Pentium 4 dengan memory (RAM) 256MB dan Operating sistem Windows 2000 profesional, mengunakan data 700MB.

Dapat dinilai hasil enkripsi AES dapat menaikkan kecepatan enkripsi dibanding menggunakan DES.

Tabel. 4.1. 3015 ke VPN Client (IPSEC Tunnel Mode)
[Server ke client]

Jenis Enkripsi	Kecepatan Tranfer data [Mbps]
Triple DES	5.005
DES	8.048
AES	14.228

Tabel. 4.2. 3015 to 3005 VPN Concentrator Lan-to-Lan (IPSEC Tunnel Mode)
[Server A ke Server B]

Jenis Enkripsi	Kecepatan Tranfer data [Mbps]
Triple DES	2.948
DES	4.927
AES 128 bit	13.315
AES 192 bit	12.754
AES 256 bit	12.526

Dapat dilihat diatas bahwa AES masih memiliki kecepatan transfer data yang lebih baik dibandingkan denga DES dan Triple DES

Namun dalam tingkat kemanan Triple DES memang lebih baik dibandingkan dengan AES tapi perlu diketahui tingkat keamanan AES masih jauh lebih baik jika dibandingkan dengan sistem enkripsi pendahulunya yaitu DES. Dapat dilihat pada tabel berikut untuk tingkat kemanan antara Triple DES dengan AES:

Tabel.4.3. Perbandingan AES vs Triple-DES

AES vs. Triple-DES		
	AES	**Triple-DES**
Key size (in bits)	128, 192, 256	112 or 168
Speed	Tinggi	Rendah
Waktu crack (asumsi mesin bekerja dengan 255 key/detik — NIST)	149 milyar years	4.6 trilyun years
Resource consumption	Rendah	Medium

Sangat jelas dengan kecepatan yang tinggi yang dimiliki AES masih dibutuhkan 149 milyar tahun untuk memecahkan kode dari enkripsi AES dan ini sangat cukup untuk mengamankan sebuah file dengan kecepatan pengiriman data yang tinggi.

Untuk menganalisa lebih lanjut tentang kecepatan pengiriman data terhadap proses enkripsi yang diperlakukan pada program dapat dilihat pada tabel berikut :

Tabel 4.4. Perbandingan kecepatan pengiriman data dalam berbagai kondisi

Tidak ada proses Enkripsi				
Besar File	**0.1KB**	**1KB**	**100KB**	**1MB**
Test 1 [ms]	90	91	403	390
Test 2 [ms]	90	91	411	490
Test 3 [ms]	89	90	407	440
Rata-rata [ms]	89.67	90.67	407.00	440.00

Enkripsi dengan DES				
Besar File	**0.1KB**	**1KB**	**100KB**	**1MB**
Test 1 [ms]	90	160	521	1352
Test 2 [ms]	90	101	460	1022
Test 3 [ms]	100	100	471	1081
Rata-rata [ms]	93.33	120.33	484.00	1151.67

Enkripsi dengan AES-128Bit				
Besar File	**0.1KB**	**1KB**	**100KB**	**1MB**
Test 1 [ms]	90	90	461	801
Test 2 [ms]	90	90	480	801
Test 3 [ms]	90	100	490	752
Rata-rata [ms]	90.00	96.67	477.00	785.00

Enkripsi dengan AES-192Bit				
Besar File	**0.1KB**	**1KB**	**100KB**	**1MB**
Test 1 [ms]	90	92	411	791
Test 2 [ms]	90	97	431	771
Test 3 [ms]	100	93	430	751
Rata-rata [ms]	93.33	94.00	424.00	771.00

Enkripsi dengan AES-192Bit				
Besar File	**0.1KB**	**1KB**	**100KB**	**1MB**
Test 1 [ms]	101	100	441	751
Test 2 [ms]	90	90	481	781
Test 3 [ms]	100	100	441	771
Rata-rata [ms]	97.00	96.67	454.33	767.67

Dapat terlihat pada table diatas bagaimana kecepatan AES bekerja tanpa begitu banyak mempengaruhi kecepatan pengiriman data dibandingkan dengan generasi enkripsi sebelumnya yaitu DES, dapat disimpulkan bahwa AES masih jauh lebih baik untuk digunakan pada komunikasi jaringan.

Jika ingin membandingkan kemungkin terpecahkannya kode yang terkirim dalam sistem enkripsi AES dapat digambarkan sebagai berikut :

- Untuk AES, terdapat 3.4 x 10^{38} kemungkinan pada sistem AES-128-bit , 6.2 x 10^{57} kemungkinan pada AES-192-bit, dan 1.1 x 10^{77} kemungkinan pada AES-256-bit.
- DES hanya mempunyai kemungkinan 7.2 x 10^{16} kemungkinan pada DES-56bit.
- Pada proses enkripsi 255 key/detik maka DES membutuhkan waktu 9 juta tahun untuk dipecahkan kodenya sedangkan AES dibutuhkan 149 milyar tahun untuk dipecahkan, namun untuk enkripsi Triple DES dibutuhkan 4.6 trilyun tahun.

Untuk saat ini jika ingin mengamankan jaringan komunikasi maka hanya ada dua kemungkinan enkripsi yang dapat dipakai yaitu AES dan Triple DES hal ini dikarenakan oleh alasan kecepatan dan kemungkinan dipecahkannya kode enkripsi, sedangkan untuk sistem enkripsi DES sudah sangat jarang digunakan apalagi pada pasar gelap dunia cyber sudah tersedia software untuk meng-crack enkripsi DES dengan harga berkisar $250.000 US dollar hingga dengan software ini kode enkripsi DES yang tadinya membutuhkan 9 juta tahun untuk memecahkan kodenya dengan software ini hanya dibutuhkan waktu kurang dari 4(empat) hari.

BAB V KESIMPULAN

V.1. Kesimpulan

Berdasarkan dari hasil penelitian dan pembuatan program maka ada beberapa kesimpulan yang dapat diambil pada perancangan enkripsi tugas akhir ini:

a. AES (Advance Encryption Standard) merupakan salah satu pilihan terbaik dalam enkripsi data pada komunikasi jaringan public dibandingkan dengan DES dan Triple-DES karena selain sangat rumit untuk diterjemahkan oleh pihak ketiga, AES juga memiliki kecepatan pengiriman data yang dapat diandalkan tanpa begitu banyak mempengaruhi kecepatan pengiriman data tanpa proses enkripsi.
b. AES memiliki banyak pilihan enkripsi sesuai dengan tingkat keamanan yang diinginkan oleh pengguna yaitu ; 128 bit, 192 bit dan 256 bit.

V.2. Saran

Pengembangan aplikasi Enkripsi pada sistem komunikasi pada jaringan web server masih memerlukan analisa yang lebih mendalam untuk lebih mengamankan data yang dikirim hal ini dikarenakan jaringan web server adalah jaringan publik dimana setiap orang yang terkoneksi keinternet pasti menggunakan jaringan web server, maka penulis mempunyai beberapa saran dalam hal ini :

a. Maksimalkan tingkat pengacakan pada sistem enkripsi namun jangan lupa untuk menggunakan listing program sesederhana mungkin karena semakin rukit program yang dibuat akan memperlambat proses komunikasi data.
b. Penggunaan software JAVA sangat memudahkan pembuat program untuk membuat suatu aplikasi, terlebih dengan dukungan class-class java untuk security terus dikembangkan untuk sistem enkripsi DES, Triple DES dan AES.
c. Lakukan aplikasi enkripsi pada data-data tertentu saja karena walaupun program dibuat seminimal mungkin dalam mempengaruhi kecepatan jaringan tentu akan sangat berpengaruh jika sudah menyangkut pengiriman data dengan ukuran besar.
d. Walaupun web server adalah jaringan publik yang mudah diakses namun dengan sistem enkripsi rumit yang bertingkat akan sangat cukup untuk mengamankan data yang dikirim

e. Dari pada membuat / membangun jaringan sendiri dalam sebuah perusahaan yang membutuhkan koneksi di berbagai daerah sebaiknya memakai jaringan public / web server saja, karena selain murah web server juga tidak perlu membangun infrastruktur yang baru, jaringan web server sudah dikenal banyak orang sehingga para user dapat dengan mudah memakainya.

f. Kerena biasanya software yang dibuat pada Program JAVA dibuat menjadi open source maka sebaiknya dalam pembuatan prosedur program setiap aplikasi khusus dibuat ke dalam satu file dengan nama yang spesifik sehingga jika terdapat user yang ingin memodifikasi atau mengupgrade software akan mudah dilakukan akan sangat mudah jika pada main program class-class yang dipakai pada main program sudah mengacu kepada penamaan algoritma yangm mudah dimengerti.

g. Pengolahan data pada aplikasi enkripsi dibuat pada dua tingkat pengakcakan yaitu pada tingkat karakter ASCII dan tingkat Binner, pengacaka tingkat ASCII dipakai untuk pengacakan,pergeseran dan operasi logika, sedangkan tingkat Binner dipakai untuk mempermudah proses matematik / aljabar yang diterapkan pada sistem enkripsi.

Daftar Pustaka

Eric Amstrong, Jenifer Ball, Sthephani Bodoff, *The J2EE™ 1.4 Tutorial*, Sun Micro Siytem 2005.

Isak Rickyanto, ST, *Dasar pemrograman Orientasi Object Java2 (JDK1.4)*, Andy Yogyakarta 2003.

Java Security

Onno W Purbo, *Trik Pemrograman JAVA untuk Jaringan dan Internet*, Elex Media Komputindo 200