PENGEMBANGAN ALGORITMA KRYPTOGRAFI KLASIK PADA APLIKASI CHATTING DENGAN METODE CE-VI

SKRIPSI

Disusun oleh:

RENDRA PUGUH W.S
04.12.654

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER & INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011

PENGEMBANGAN ALGORITMA KRYPTOGRAFI KLASIK PADA APLIKASI CHATTING DENGAN METODE CE-VI

SKRIPSI

Disusun oleh:

RENDRA PUGUH W.S
04.12.654

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER & INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2011

ABSTRAK

PENGEMBANGAN ALGORITMA KRYPTOGRAFI KLASIK PADA APLIKASI CHATTING DENGAN METODE CE-VI

Rendra Puguh W.S, NIM 0412654
Dosen Pembimbing : Ir. Yusuf Ismail Nakhoda, MT dan
Dr. Eng. Aryuanto Soetedjo, MT.

Kemajuan teknologi saat ini sudah banyak mempengaruhi kehidupan kita sehari – hari tak terkecuali dalam bidang komunikasi khususnya *chatting*. Namun kemajuan teknologi tersebut harus diimbangi pula dengan jaminan keamanan informasi atau data yang baik, bila tidak maka tingkat kebocoran informasi atau data akan sering terjadi.

Untuk mengatisipasi hal tersebut perlu menerapkan metode yang bisa menjaga kerahasiaan suatu informasi, salah satunya dengan menerapkan kriptografi. Ce-Vi ialah salah satu algoritma yang berasal dari kombinasi pengembangan kriptografi klasik yang sudah ada yaitu algoritma *Caesar* dan algoritma *Vignere*, dimana nantinya algoritma ini akan diterapkan pada aplikasi *chatting*.

Kelebihan dari algoritma Ce-Vi ini selain terletak pada kombinasi penggunaan dua metode kriptografi klasik (*Caesar* dan *Vignere*) yang dijadikan menjadi satu proses juga terletak pada jumlah karakter yang digunakan. Sehingga hasil yang didapatkan algoritma ini lebih luas cakupannya dan efisien daripada menggunakan algoritma *Caesar* atau *Vignere* saja serta dengan kombinasi dua buah metode ini memungkinkan pihak – pihak luar tidak dapat mengetahui algoritma apa yang sedang dipakai.

Kata Kunci : Kriptografi, Caesar, Vignere, Ce-Vi, Chatting.

DAFTAR ISI

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi saat ini sudah banyak mempengaruhi kehidupan kita sehari – hari tak terkecuali dalam bidang komunikasi. Dengan adanya telepon genggam atau *handphone (hp)* maupun internet , komunikasi jarak jauh dapat dilakukan dengan cepat dan mudah. Namun kemajuan teknologi tersebut harus diimbangi dengan jaminan keamanan yang baik, dalam hal ini seperti internet *(chatting)* ataupun telepon genggam juga rawan terhadap penyadapan informasi yang dilakukan oleh pihak – pihak yang tidak seharusnya tidak boleh mengetahui informasi tersebut.

Untuk itu keamanan dan kerahasiaan data atau informasi harus dijaga. Keamanan dan kerahasiaan data pada jaringan komputer saat ini menjadi isu yang sangat penting dan terus berkembang. Hal ini disebabkan karena kemajuan bidang jaringan komputer dengan konsep *open system*-nya sehingga siapapun, dimanapun, dan kapanpun mempunyai kesempatan untuk mengakses informasi – informasi yang disediakan. Salah satunya yang pernah terjadi pada para pengguna salah satu jejaring sosial yang terkenal, diberitakan bahwa para pengguna jejaring sosial tersebut dikejutkan dengan peristiwa bocornya pembicaraan *chat* mereka dan kebocoran data – data lain seperti daftar tunggu dan permintaan pertemanan, serta informasi – informasi yang berpotensi berbahaya lain[3].

Pengamanan data pada prinsipnya berfungsi untuk melindungi data agar tidak dapat dibaca oleh orang – orang yang tidak berhak dan juga mencegah agar orang – orang yang tidak berhak menyisipkan atau menghapus data. Banyak cara untuk melindungi dan menjaga kerahasaiaan suatu informasi, salah satunya dengan membuat aplikasi dengan menggunakan metode kriptografi. Kriptografi merupakan ilmu teknik penyandian yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, authentication dan keaslian data[1]. Diharapkan dengan menerapkan metode atau ilmu kriptografi ini bisa mengurangi kebocoran data sehingga dalam hal ini para pengguna *(user)* bisa mengirimkan data atau informasi ke pihak lain tanpa harus khawatir akan bahaya dari pihak lain. Walaupun sebenarnya teknik kriptografi ini juga tidak bisa menjamin seratus persen keamanan suatu informasi. Karena semakin berkembangnya zaman, teknik kriptografi semakin lama juga akan semakin bisa ditembus oleh pihak lain bilamana teknik tersebut tidak diperbarui dan dikembangkan

lagi. Untuk mengatasi hal tersebut perlu dilakukan upaya pengembangan teknik kriptografi yang sudah ada menjadi lebih baik dengan algoritma – algoritma yang semakin kompleks sehingga kebocoran informasi bisa diminimalisir.

1.2 Rumusan Masalah

Berdasarkan hal di atas ,permasalahan yang timbul adalah bagaimana membuat suatu aplikasi kriptografi, salah satunya dengan cara mengkombinasikan dan mengembangkan dua metode kriptografi yang sudah ada serta menerapkannya pada aplikasi *chatting*.

1.3 Tujuan

Tujuan yang diharapkan dari pembuatan aplikasi ini adalah untuk menjaga kerahasiaan informasi atau data saat percakapan dua pengguna sedang berkomunikasi dengan menyisipkan kombinasi algoritma kriptografi di dalam aplikasi tersebut.

1.4 Batasan Masalah

Agar pembahasan mengarah sesuai dengan tujuan pembuatan aplikasi ini,maka pembahasan dibatasi pada hal – hal sebagai berikut:

1. Aplikasi *Chatting* dengan algoritma kriptografi yang akan dibangun menggunakan *input*-an berupa karakter.
2. Di dalam tugas akhir ini hanya membahas Kriptografi Klasik dengan menggunakan Metode Penyandian Subtitusi serta tidak membahas kriptografi modern.
3. Dalam tugas akhir ini lebih ditekankan pada aplikasi kriptografinya sendiri daripada yang lain.
4. Di dalam tugas akhir ini tidak membahas tentang kriptanalisis.
5. Sistem jaringan yang digunakan tidak termasuk dalam pembahasan.

1.5 Metode Penelitian

1.5.1 Metode Pengumpulan Data

Dengan metode ini data diperoleh langsung dari sumber yang bersangkutan, yang dilakukan dengan cara :

1. Studi Lapangan
   a. *Survey*

      Teknik pengumpulan data dengan cara terjun secara langsung dan mencatat secara sistematis terhadap obyek masalah.

   b. Wawancara / *Interview*

      Teknik pengumpulan data dengan jalan mengadakan komunikasi atau tanya jawab secara langsung .
2. Studi Pustaka/Literatur

   Mencari buku- buku dan literatur yang dapat mendukung dalam penyelesaian skripsi ini.

1.5.2. Metode Pengembangan Sistem

Metode pengembangan sistem perangkat lunak menggunakan metodologi *waterfall,* yaitu sebagai berikut [1]:

1. *Analysis* (Analisis)

   Merupakan tahap menggabungkan hasil studi lapangan dan kebijakan pemakai menjadi spesifikasi kebutuhan sistem dengan menggunakan pemodelan, tahapan ini dibahas pada Bab III.
2. *Design* (Perancangan)

   Perancangan merupakan tahap penerjemahan dari model yang diinginkan pemakai, tahapan ini dibahas pada Bab III.
3. *Coding* (Pengkodean)

   Pada tahap ini dilakukan proses mengimplementasikan dari model atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman komputer, tahapan ini dibahas pada Bab IV.
4. *Testing* (Pengujian)

   Menguji coba spesifikasi program dan sistem secara keseluruhan, tahapan ini dibahas pada Bab IV.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan skripsi ini adalah sebagai berikut:

BAB I : Pendahuluan, meliputi latar belakang, rumusan masalah, batasan masalah, ruang lingkup, tujuan dan sistematika penulisan.

BAB II : Dasar teori, berupa penjelasan tentang pengertian atau definisi Kriptografi, Algoritma *Caesar Cipher*, Algoritma *Vignere Cipher*.

BAB III : Perancangan Sistem dan Analisa Algoritma, berisi perancangan sistem yang akan digunakan serta analisa algoritma *Caesar*, algoritma *Vignere*, algoritma Ce-Vi serta aplikasi *Chatting*.

BAB IV : Pengujian hasil, menyajikan pembuatan obyek uji, hasil pengujian serta pembahasan dari hasil pengujian yang dilakukan.

BAB V : Penutup, berisi kesimpulan dan saran dari hasil pembahasan pada skripsi ini.

BAB II
DASAR TEORI

2.1 Kriptografi

Kriptografi mempunyai peranan penting dalam dunia komputer, khususnya yang berhubungan dengan pengamanan informasi. Banyaknya informasi – informasi rahasia yang dikirimkan melalui media komputer membuat perlu diterapkannya ilmu kriptografi agar bisa dikembangkan setiap saat. Informasi – informasi ini biasanya berisikan informasi atau data penting dari seseorang, perusahaan ataupun instansi yang tidak ingin dibaca oleh orang yang tidak berhak atas informasi tersebut.

2.1.1 Definisi Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu *"cryptos"* yang berarti rahasia, sedangkan *"graphein"* artinya tulisan. Sehingga secara morfologi kriptologi berarti tulisan rahasia. Ada beberapa definisi kriptografi yang telah dikemukakan didalam beberapa literatur. Definisi yang kita pakai di dalam tulisan ini : Kriptografi adalah ilmu dan seni untuk menjaga pesan[1]. Kata "seni" di dalam definisi di atas berasal dari fakta sejarah bahwa pada masa – masa awal sejarah kriptografi, setiap orang mempunyai cara yang unik untuk merahasiakan pesan. Pada perkembangan selanjutnya, kriptografi berkembang menjadi sebuah disiplin ilmu sendiri karena teknik – teknik kriptografi dapat diformulasikan secara matematika menjadi sebuah metode yang formal.

2.1.2 Sejarah Kriptografi

Kriptografi mempunyai sejarah yang panjang. Informasi yang lengkap mengenai sejarah kriptografi dapat ditemukan di dalam buku David Khan yang berjudul *The Codebreakers*. Buku yang tebalnya 1000 halaman ini menulis secara rinci sejarah kriptografi mulai dari penggunaan kriptografi oleh Bangsa Mesir 4000 tahun yang lalu (berupa *hieroglyph* yang tidak standard piramid) hingga penggunaan kriptografi pada abad ke-20[7].

Sejarah kriptografi sebagian besar merupakan sejarah kriptografi klasik, yaitu metode enkripsi yang menggunakan kertas dan pensil atau mungkin dengan bantuan alat mekanik sederhana. Secara umum kriptografi klasik dikelompokkan menjadi dua kategori, yaitu algoritma transposisi (*transposition cipher*) dan algoritma subtitusi (*subtitution cipher*). *Cipher* transposisi mengubah susunan huruf – huruf dalam pesan,

sedangkan *cipher* subtitusi mengganti setiap huruf atau kelompok huruf dengan sebuah huruf atau kelompok huruf lain[7].

Sejarah kriptografi klasik mencatat penggunaan *cipher* transposisi oleh tentara Yunani pada permulaan tahun 400 SM. Mereka menggunakan alat yang namanya *Scytale*. *Scytale* terdiri dari sebuah kertas panjang daun *papyrus* yang dililitkan pada sebuah silinder dari diameter tertentu (diameter silinder menyatakan kunci penyandian). Pesan ditulis secara horizontal, baris per baris. Bila pita dilepaskan, maka huruf – huruf di dalamnya telah tersusun secara acak membentuk pesan rahasia[7]. Dalam gambar 2.1 ditunjukkan gambar sebuah *Scytale* dan pesan yang ditulis secara horisontal dan baris per baris.

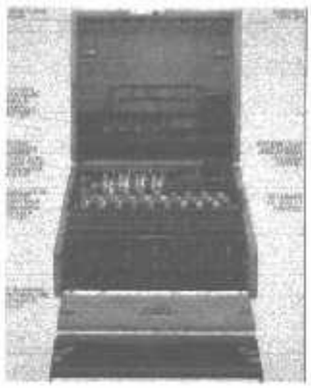
Gambar 2.1 (a) Sebuah *Syctale* (b) Pesan Ditulis Secara Horizontal, Baris Per Baris

Untuk membaca pesan, penerima pesan harus melilitkan kembali kertas tersebut ke silinder yang diameternya sama dengan diameter silinder pengirim. Sedangkan algoritma subtitusi paling awal dan paling sederhana adalah *Caesar Cipher*, yang digunakan oleh Raja Yunani kuno, Julius Caesar. Caranya adalah mengganti setiap karakter di dalam alfabet dengan karakter yang terletak pada tiga posisi berikutnya di dalam susunan alfabet[7].

Kriptografi juga digunakan untuk keamanan seperti pada kalangan agama Kristen yang menggunakan kriptografi untuk menjaga tulisan relijius dari gangguan otoritas politik atau budaya yang dominan itu. Di India, kriptografi digunakan oleh pencinta (*lovers*) untuk berkomunikasi tanpa diketahui orang lain. Pada Abad ke-17, sejarah kriptografi mencatat korban ketika ratu Skotlandia, Queen Mary, setelah surat rahasianya dari balik penjara (surat terenkripsi yang isinya rencana membunuh Ratu Elizabeth I) berhasil dipecahkan oleh seorang pemecah kode[7].

Pada kalangan militer, kriptografi juga digunakan yaitu pada saat Perang Dunia II saat pemerintah Nazi Jerman membuat mesin enkripsi yang dinamakan *Enigma*. Mesin yang menggunakan beberapa buah *rotor* (roda berputar) ini melakukan enkripsi

dengan cara yang sangat rumit. Namun *Enigma Cipher* berhasil dipecahkan oleh pihak sekutu dan keberhasilan memecahkan *Enigma* sering dikatakan sebagai faktor yang memperpendek Perang Dunia II[7]. Adapun bentuk *Enigma* seperti pada gambar 2.2



Gambar 2.2 *Enigma*

Kriptografi modern dipicu oleh perkembangan peralatan komputer digital saat ini. Dengan komputer digital, *cipher* lebih kompleks menjadi sangat mungkin untuk dapat dihasilkan. Tidak seperti yang mengenkripsi karakter per karakter (dengan menggunakan alfabet tradisionil), kriptografi modern beroperasi pada *string biner*[7].

2.1.3 Tujuan Kriptografi

Adapun konsep penggunaan kriptografi antara lain[1] :

1. Kerahasiaan (*Confidentiality*)
   Proses penyembunyian data dari orang – orang yang tidak punya otoritas.
2. Integritas (*Integrity*)
   Proses untuk menjaga agar sebuah data tidak diubah – ubah sewaktu dikirim atau disimpan.
3. Penghindaran penolakan (*Non-repuditation*)
   Proses untuk menjaga bukti – bukti bahwa suatu data berasal dari seseorang. Seseorang yang ingin menyangkal bahwa data tersebut bukan berasal dari dirinya dapat saja melenyapkan bukti – bukti yang ada, Karenanya diperlukan teknik untuk melindungi data – data tersebut.
4. Autentikasi (*Authentication*)
   Proses untuk menjamin keaslian suatu data.

5. Tanda Tangan Data (*Data Signature*)

   Dapat disebut juga tanda tangan digital, berguna untuk menandatangani data digital.

6. Kontrol Akses (Access Control)

   Untuk mengontrol akses terhadap suatu *entity*.

2.1.4 Terminologi

Di dalam kriptografi akan sering menemukan berbagai istilah atau terminologi. Beberapa istilah penting untuk diketahui[7].

a. Plainteks dan Cipherteks

Pesan ialah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain dari pesan adalah Plainteks. Plainteks dapat berupa data atau informasi yang dikirim(melalui kurir, saluran telekomunikasi, dsb) atau yang disimpan di dalam media perekaman. Plainteks yang tersimpan tidak hanya berupa teks, tetapi juga berbentuk citra (*image*), suara atau bunyi (*audio*), dan *video* , atau berkas *biner* lainnya[6].

Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan ke bentuk lain yang tidak dapat dipahami. Bentuk pesan(plainteks) yang tersandi disebut cipherteks (*ciphertext*). Cipherteks harus dapat ditransformasikan kembali menjadi plainteks semula agar pesan yang diterima bisa dibaca[7]. Gambar 2.3 merupakan contoh perbandingan plainteks dan cipherteks.

Untuk program pengembangan selanjutnya perlu dibuat jaringan yang terintegrasi agar lebih memudahkan informasinya

(a) Plainteks

BffbcqJjaTjSZwbRfYReTNfYNfq`] ^Nf\bl`fYq]]dYmqQaTbYflbS_a`T Y`lqS[_qa]dVffR_dNk[lYYNjqY]TV `qZ]_b\SUcS[w[[^a_eS`a`fY

(b) Cipherteks

Gambar 2.3 Perbandingan Plainteks dan Cipherteks

b. Pengirim dan Penerima

Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*Sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*Receiver*) adalah entitas yang menerima pesan. Entitas – entitas ini dapat berupa orang, mesin (komputer), kartu kredit dan

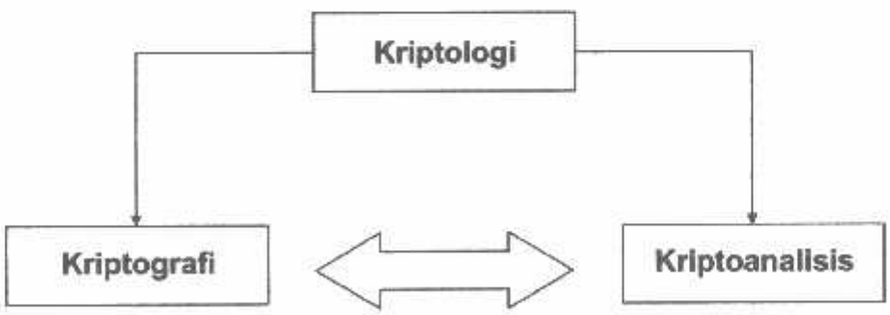
sebagainya. Pengirim ingin mengirimkan pesan dengan aman sampai ke penerima. Solusinya adalah dilakukan penyandian terhadap pesan tersebut agar tidak diketahui pihak – pihak yang tidak berkepentingan terhadap pesan tersebut[7].

c. Enkripsi dan Dekripsi

Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) sedangkan proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi (*decryption*). Enkripsi dan dekripsi dapat diterapkan baik pada pesan yang dikirim maupun pesan yang diterima. Istilah *encryption of data motion* mengacu pada enkripsi pesan yang ditransmisikan melalui saluran komunikasi, contohnya adalah pengiriman nomor PIN dari mesin ATM ke komputer *server* di kantor bank pusat. Sedangkan istilah *encryption of data at-rest* mengacu pada enkripsi dokumen yang disimpan di dalam *storage*, contohnya adalah enkripsi *file basis* data di dalam *hard disk*[7].

d. Kriptoanalisis dan Kriptologi

Kriptoanalisis dapat diartikan sebagai seni atau ilmu untuk memecahkan cipherteks dan plainteks dengan memanfaatkan celah – celah keamanan sebuah sistem kriptografi[3]. Hal inilah yang menjadikan kriptoanalisis dicap sebagai cara illegal untuk menterjemahkan cipherteks. Orang yang melakukan kriptoanalisis disebut **kriptoanalis**, dan usaha - usaha untuk melakukan kriptoanalisis disebut dengan *attack* (serangan). Sedangkan kriptologi (*cryptologi*) adalah studi mengenai kriptografi dan kriptoanalisis. Gambar 2.4 menunjukkan hubungan antara kriptologi, kriptografi dan kriptoanalisis.



Gambar 2.4 Hubungan Antara Kriptologi, Kriptografi Dan Kriptoanalisis

Sebenarnya, melakukan serangan pada sebuah system kriptografi merupakan pekerjaan yang sulit dan membutuhkan logika dan itelegensia yang tinggi, dan didukung dengan alat – alat yang memadai. Karena sistem yang akan diserang tentunya telah dilengkapi dengan program perlindungan serangan yang cukup kuat, serta dengan nilai informasi yang harus dijaganya[8].

Dari kemungkinan di atas , diperoleh beberapa jenis serangan yang bisa dilakukan oleh kriptoanalis, dengan asumsi bahwa kriptoanalis telah mengetahui algoritma kriptografi yang digunakan dalam sistem yang akan diserang , yaitu[3] :

1. **Ciphertext Only Attack**

   Kriptoanalis hanya mempunyai beberapa cipherteks hasil dari penyadapan. Namun ia tidak mengetahui kunci serta plainteksnya. Pekerjaan kriptoanalis adalah mencari kunci dekripsi untuk memperoleh plainteksnya[8].

2. **Known Plaintext Attack**

   Kriptoanalis berhasil memperoleh potongan plainteks dan sebuah cipherteks lengkap, namun ia yakin kalau keduanya saling berhubungan[8].

3. **Choosen Plaintext Attack**

   Kriptoanalis tidak hanya mengetahui sebuah plainteks dan cipherteks nya saja, tetapi juga bebas memilih beberapa plainteks yang dianggap sesuai dengan bagian tertentu dari cipherteks. Tugas kriptoanalis selanjutnya menebak kunci[8].

4. **Adaptive Choosen Plaintext Attack**

   Serangan ini merupakan kasus khusus dari serangan ketiganya di atas. Kriptoanalis tidak hanya dapat memilih plainteks yang akan dienkripsi, namun juga dapat memodifikasi pilihannya berdasarkan hasil enkripsi sebelumnya. Dalam *Choosen Plaintext Attack.* mungkin hanya dapat memlih satu blok besar plainteks untuk dienkripsi, sedangkan pada serangan ini dia dapat memilih blok plainteks yang lebih kecil dan kemudian memilih lainnya berdasar kan hasil sebelumnya[8].

5. **Choosen Ciphertext Attack**

   Kriptoanalis dapat memilih cipherteks yang berbeda untuk didekripsi dan mempunyai akses terhadap cipherteks yang dienkripsi. Sebagai contoh, kriptoanalis mempunyai akses ke kotak elektronik yang dapat melakukan proses dekripsi secara otomatis. Pekerjaan kriptoanalis adalah menemukan kunci dekripsi[8].

6. **Choosen Text**

   Merupakan gabungan dari *Choosen Plaintext Attack* dan *Choosen CiphertextAttack*. Di sini kriptoanalis telah mengetahui algoritma enkripsi yang digunakan serta cipherteks yang akan dibaca.Kriptoanalis juga dapat memilih plainteks yang akan dienkripsi bersama cipherteks pasangannya yang dibangkitkan dengan kunci rahasia tertentu[8].

2.1.5 Jenis Kriptografi

Berdasarkan kunci enkripsi dan dekripsinya algoritma kriptografi terbagi menjadi dua bagian :

1. **Kriptografi Simetri**

   Konsep dasar dari kriptografi kunci simetri adalah di mana kunci untuk enkripsi dan dekripsi sama. Istilah lain dari kriptografi simetri ini adalah kriptografi kunci privat (*privat-key cryptography*), kriptografi kunci rahasia (*secret-key cryptography*), atau kripografi konvensional (*conventional cryptography*). Dalam kriptografi kunci simetri dapat diasumsikan bahwa si pengirim dan si penerima telah terlebih dahulu berbagi kunci sebelum pesan dikirimkan[7].

   Kelebihan Kriptografi Simetri adalah :

   1. Proses enkripsi atau dekripsi simetri membutuhkan waktu yang singkat.
   2. Ukuran kunci simetri lebih pendek.
   3. Otentikasi pengiriman pesan langsung diketahui dari cipherteks yang diterima , karena kunci hanya diketahui oleh penerima dan pengirim saja.

Kekurangan Kriptografi Simetri adalah :

1. Kunci simetri harus dikirimkan melalui saluran komunikasi yang aman, dan kedua entitas yang berkomunikasi harus menjaga kerahasiaan kunci.
2. Kunci harus sering diubah, setiap kali melaksanakan komunikasi.

2. Kriptografi Asimetri

Berbeda dengan kriptografi kunci simetri. Kriptografi kunci publik memiliki dua buah kunci yang berbeda pada proses enkripsi dan dekripsinya. Nama lain dari kunci asimetri ini adalah kriptografi kunci-publik (*publik-key kriptografi*). Kunci untuk enkripsi pada kriptografi asimetri ini tidak rahasia (diketahui oleh publik), sedangkan kunci untuk dekripsi bersifat rahasia (kunci privat). Entitas pengirim akan mengenkripsi dengan menggunakan kunci publik, sedangkan entitas penerima akan mendekripsikan menggunakan kunci privat[7].

Kelebihan Kriptografi Asimetri adalah :

1. Hanya kunci privat yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi baik pengirim dan penerima.
2. Pasangan kunci privat dan kunci publik tidak perlu diubah dalam jangka waktu yang lama.
3. Dapat digunakan dalam pengaman pengiriman kunci simetri.
4. Beberapa algoritma kunci publik dapat digunakan untuk memberi tanda tangan digital pada pesan.

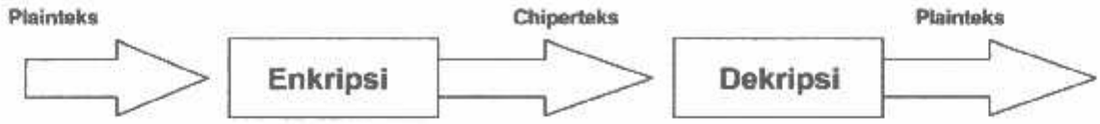
Kelemahan Kriptografi Asimetri adalah :

1. Proses enkripsi dan dekripsi umumnya lebih lambat dari algoritma simetri, karena menggunakan bilangan yang besar dan operasi bilangan yang besar.
2. Ukuran cipherteks lebih besar daripada plainteks.
3. Ukuran kunci relatif lebih besar daripada ukuran kunci simetri.

2.1.6 Prinsip Kerja Kriptografi

Secara umum kritografi merupakan teknik pengamanan informasi yang dilakukan dengan cara mengolah informasi awal (plainteks) dengan suatu kunci tertentu menggunakan suatu metode enkripsi tertentu sehingga menghasilkan suatu informasi baru (cipherteks) yang tidak dapat dibaca secara langsung. Cipherteks tersebut dapat

dikembalikan menjadi informasi awal (plainteks) melalui proses dekripsi. Fungsi – fungsi yang mendasar dalam kriptografi adalah enkripsi dan dekripsi. Enkripsi adalah proses mengubah suatu pesan asli (plainteks) menjadi suatu pesan dalam bahasa sandi (Cipherteks)[1]. Gambar 2.5 menunjukkan urutan proses kriptografi.



Gambar 2.5 Urutan Proses Kriptografi

Secara matematika dapat ditulis :

$$C = E(P) \quad \text{.....(2-1)}$$

dimana : E = proses enkripsi

C = pesan dalam bahasa sandi

P = pesan asli

Sedangkan dekripsi adalah proses mengubah pesan dalam bahasa sandi menjadi pesan asli kembali

$$P = D(C) \quad \text{.....(2-2)}$$

dimana : P = pesan asli

D = proses dekripsi

C = pesan dalam bahasa sandi

2.2 Caesar Cipher

Salah satu metode penyandian yang pernah digunakan pada masa Yunani Kuno adalah Sandi Caesar (*Caesar Cipher*). Sandi ini merupakan algoritma subtitusi tertua, dan proses enkripsinya mengganti (subtitusi) setiap huruf pada plainteks menjadi huruf ke-3 setelahnya. Dengan kata lain, setiap huruf digeser maju sebanyak tiga huruf. Dan untuk mendekripsi cipherteks cukup dengan menggeser mundur sebanyak tiga huruf[1]. Sehingga *Caesar Cipher* ini digolongkan atau dikenal sebagai *monoalphabetic substitution chipper* karena satu huruf tertentu pasti akan berubah menjadi huruf tertentu yang lain.

2.2.1 Dasar Teknik Enkripsi dan Dekripsi Caesar Cipher

Dasar teknik *Caesar Cipher* adalah mengganti (mensubtitusi) setiap karakter dengan karakter lain dalam susunan abjad (alfabet). Misalnya, tiap huruf disubtitusi

dengan huruf ketiga berikutnya dari susunan abjad. Dalam hal ini kuncinya adalah jumlah pergeseran huruf (yaitu k=3) seperti halnya yang ditunjukkan pada Tabel 2.1. Sedangkan untuk mengetahui nilai tiap – tiap karakternya dapat ditunjukkan pada Tabel 2.2

Tabel 2.1 Subtitusi *Caesar Cipher* (26 karakter)

p_i	:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
c_i	:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Tabel 2.2 Nilai Tiap Karakter Pada *Caesar Cipher* (26 karakter)

Kunci	A	B	C	D	E	F	G	H	I
Pergeseran k	0	1	2	3	4	5	6	7	8
Kunci	J	K	L	M	N	O	P	Q	R
Pergeseran k	9	10	11	12	13	14	15	16	17
Kunci	S	T	U	V	W	X	Y	Z	
Pergeseran k	18	19	20	21	22	23	24	25	

Contoh :

Pesan :

INSTITUT TEKNOLOGI NASIONAL MALANG

disamarkan (enkripsi) menjadi

LQVWLWXW WHNQRORJM QDVLRQDO PDODQJ

Penerima pesan men-dekripsi *Cipherteks* tersebut, dengan menggunakan tabel subtitusi (Tabel 2.1)

Cipherteks :

LQVWLWXW WHNQRORJM QDVLRQDO PDODQJ

Sehingga akan dikembalikan ke bentuk semula

INSTITUT TEKNOLOGI NASIONAL MALANG

Karena hanya ada 26 huruf abjad, maka pergeseran huruf yang mungkin dilakukan adalah dari 0 sampai 25. Secara umum, untuk pegeseran huruf sejauh k (dalam hal ini k adalah kunci enkripsi dan dekripsi), fungsi enkripsi adalah

$$C_i = E(P_i) = (P_i + k) \bmod 26 \quad \text{..... (2-3)}$$

Dan fungsi dekripsi adalah

$$P_i = D(C_i) = (C_i - k + 26) \bmod 26 \quad \text{..... (2-4)}$$

Dimana : C_i = Cipherteks

D = Dekripsi

E = Enkripsi

P_i = Plainteks

k = kunci

Sebagai contoh :

Untuk proses enkrpsinya :

Nilai karakter pada P = I (8) dengan pergeseran k(*key*) = 3

bila diterapkan pada persamaan (2-3) maka akan menjadi

C = (P+k) mod 26

C = (8+3) mod 26

C = 11 mod 26

C = **11** atau sama dengan karaker (**L**)

Sedangkan untuk proses dekripsinya dapat menggunakan persamaan (2-4):

P = (C-k+26) mod 26

P = (11-3+26) mod 26

P = 34 mod 26

P = **8** atau sama dengan karakter (**I**)

2.2.2 Teknik Kriptanalisis Caesar Cipher

Caesar Cipher mudah dipecahkan dengan metode *exhaustive key search* karena jumlah kuncinya sangat sedikit (hanya ada 26 kunci). Misalkan kriptanalis menemukan potongan cipherteks **XMZVH**. Diandaikan kriptanalis mengetahui bahwa plainteks disusun dalam Bahasa Inggris dan algoritma kriptografi yang digunakan adalah *Caesar Chiper*. Untuk memperoleh plainteks, harus melakukan dekripsi dari kunci yang terbesar (25) sampai kunci yang terkecil (1), kemudian memeriksa apakah dekripsi tersebut menghasilkan pesan yang mempunyai makna[1]. Dalam tabel 2.3 ditunjukkan metode menggunakan *Exhaustive Key Search*.

Tabel 2.3 Metode Menggunakan *Exhaustive Key Search*

Kunci (k) chipering	'Pesan' hasil dekripsi	Kunci (k) chipering	'Pesan' hasil dekripsi	Kunci (k) chipering	'Pesan' hasil dekripsi
0	XMZVH	17	GVIEQ	8	PERNZ
25	YNAWI	16	HWJFR	7	QFSOA
24	ZOBXJ	15	IXKGS	6	RGTPB
23	ZPCYK	14	JYLHT	5	SHUQC
22	BQDZL	13	KZMIU	4	TIVRD
21	CREAM	12	LANJV	3	UJWSE
20	DSFBN	11	MBOKW	2	VKXTF
19	ETGCO	10	NCPLX	1	WLYUG
18	FUHDP	9	ODQMY		

Dari tabel diatas , kata dalam Bahasa Inggris yang potensial menjadi plainteks adalah **CREAM** dengan menggunakan $k = 21$. Kunci ini digunakan untuk mendekripsikan cipherteks lainnya. Kadang – kadang satu kunci menghasilkan pesan yang bermakna tidak satu buah, untuk itu dibutuhkan informasi yang lain dengan mendekripsikan potongan cipherteks lain untuk memperoleh kunci yang benar[1].

Cara lain digunakan untuk memecahkan cipherteks dengan statistik, yaitu dengan menggunakan tabel kemunculan karaker, yang membantu mengidentifikasi karakter plainteks yang berkoresponden dengan karakter di dalam cipherteks. Dalam hal ini, kriptanalis menggunakan tabel frekuensi kemunculan huruf-huruf dalam teks bahasa Inggris. Tabel 2.4 memperlihatkan frekuensi kemunculan huruf-huruf abjad yang diambil dari sampel yang mencapai 300.000 karakter di dalam sejumlah novel dan surat kabar[1].

Tabel 2.4 Frekuensi Kemunculan (Relatif) Huruf – Huruf
Dalam Teks Bahasa Inggris

Huruf	%	Huruf	%
A	8,2	N	6,7
B	1,5	O	7,5
C	2,8	P	1,9
D	4,2	Q	0,1
E	12,7	R	6,0
F	2,2	S	6,3
G	2,0	T	9,0
H	6,1	U	2,8
I	7,0	V	1,0
J	0,1	W	2,4
K	0,8	X	2,0
L	4,0	Y	0,1
M	2,4	Z	0,1

- Tabel 2.4 di atas pada mulanya dipublikasikan di dalam *Cipher-Systems: The Protection of Communications* dan dikompilasi oleh *H. J. Beker* dan *F.C. Piper*[1].
- Terdapat sejumlah tabel frekuensi sejenis yang dipublikasikan oleh pengarang lain, namun secara umum persentase kemunculan tersebut konsisten pada sejumlah tabel[1].
- Bila *Cipher* abjad-tunggal digunakan untuk meng-enkripsi pesan, maka kemunculan huruf-huruf di dalam plainteks tercermin pada Tabel 2.4 di atas. Misalnya bila di dalam *Cipher* abjad-tunggal huruf **R** menggantikan huruf E, maka frekuensi **R** di dalam cipherteks sama dengan frekuensi E di dalam plainteksnya[1].

2.3 Vignere Cipher

Vignere Cipher ini dipublikasikan oleh diplomat (sekaligus seorang kriptologis) Perancis, Blaise de Vignère pada abad 16 (tahun 1586). Tetapi sebenarnya Giovan Batista Belaso telah menggambarkannya pertama kali pada tahun 1553 seperti ditulis di dalam bukunya *La Cifra del Sig. Giovan Batista Belaso*. Algoritma tersebut baru dikenal luas 200 tahun kemudian yang oleh penemunya *Cipher* tersebut kemudian dinamakan *Vignere Cipher*. *Cipher* ini berhasil dipecahkan oleh Babbage dan Kasiski pada pertengahan abad 19. *Vignere Cipher* digunakan oleh Tentara Konfiderasi (*Confederate Army*) pada Perang Sipil Amerika (*American Civil War*)[1].

2.3.1 Dasar Teknik Enkripsi dan Dekripsi Vignere Cipher

Dasar teknik *Vignere Cipher* ini sama halnya pada *Caesar Cipher* yaitu dengan mengganti (mensubtitusi) setiap karakter dengan karakter lain dalam susunan abjad. Hanya saja pada *Vignere Cipher* ini menggunakan Bujursangkar *Vignere* untuk melakukan enkripsi. Setiap baris di dalam bujursangkar menyatakan huruf – huruf cipherteks yang diperoleh dengan *Caesar Cipher*. Tabel 2.5 menunjukkan Bujursangkar *Vignere*, sedangkan pada Tabel 2.6 menampilkan nilai – nilai tiap karakter pada *Vignere*.

Tabel 2.5 Bujursangkar *Vignere Cipher* (26 karakter)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Tabel 2.6 Nilai Tiap Karakter *Vignere Cipher* (26 karakter)

Kunci	A	B	C	D	E	F	G	H	I
Pergeseran k	0	1	2	3	4	5	6	7	8
Kunci	J	K	L	M	N	O	P	Q	R
Pergeseran k	9	10	11	12	13	14	15	16	17
Kunci	S	T	U	V	W	X	Y	Z	
Pergeseran k	18	19	20	21	22	23	24	25	

Contoh :

Pesan : MALANGKU

Kunci : ITN

Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci diulang secara periodik,sehingga menjadi

Pesan : MALANGKU

Kunci : ITNITNIT

Sedangkan untuk enkripsinya seperti pada tabel 2.7

Tabel 2.7 Enkripsi Huruf M Dengan Kunci I

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Sama halnya dengan persamaan fungsi enkripsi dan dekripsi pada *Caesar Cipher*, maka :

Fungsi enkripsi pada *Vignere Cipher* :

$$C_i = E(P_i) = (P_i + k) \bmod 26 \quad \text{.....(2-5)}$$

Sedangkan fungsi dekripsi :

$$P_i = D(C_i) = (C_i - k + 26) \bmod 26 \quad \text{.....(2-6)}$$

Dimana : C_i = Cipherteks

D = Dekripsi

E = Enkripsi

P_i = Plainteks

k = kunci

Hasil Enkripsi seluruhnya adalah sebagai berikut :

Plainteks : MALANGKU

Kunci : ITNITNIT

Cipherteks : UTYIGTSN

Pada *Vignere Cipher* ini huruf yang sama tidak selalu dienkripsi menjadi huruf cipherteks yang sama pula. Contoh : huruf plainteks **A** dapat dienkripsi menjadi **T** dan **I** dan huruf cipherteks **T** mempresentasikan huruf plainteks **A** dan **G**. Hal tersebut merupakan karakteristik dari *Vignere Cipher* karena setiap huruf cipherteks dapat memiliki kemungkinan banyak huruf plainteks atau yang sering dikenal *Cipher* abjad majemuk (*polyalpabetic substitution Cipher*).

Jika dihitung secara matematis menurut persamaan (2-5) , dengan nilai karakter pada P = M (12) serta nilai k(*key*) = I(8) maka :

Untuk proses enkripsi :

C = (P+k) mod 26

C = (12+8) mod 26

C = 20 mod 26

C = **20** atau sama dengan karakter (**U**)

Sedangkan untuk proses dekripsinya dapat menggunakan persamaan (2-6):

P = (C-k+26) mod 26

P = (20-8+26) mod 26

P = 38 mod 26

P = **12** atau sama dengan karakter (**M**)

2.3.2 Teknik Kriptanalisis Vignere Cipher

Salah satu kelebihan *Vignere Cipher* daripada *Caesar Cipher* adalah pada *Vignere Cipher* dapat mencegah frekuensi huruf – huruf di dalam cipherteks yang mempunyai pola tertentu yang sama seperti *Cipher* abjad tunggal. Jika periode kunci diketahui dan tidak terlalu panjang, maka kunci dapat ditentukan dengan menulis program komputer untuk melakukan *exhaustive key search* seperti pada Tabel 2.3

2.4 Pengertian Diagran Alir (Flowchart)

Flowchart adalah tabel keputusan dengan jalur yang terpisah melalui *flowchart* yang mampu menghasilkan aturan keputusan. Menurut Alton. R. Kindred (1985), *flowchart* adalah :

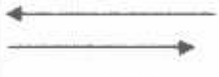
"Menjelaskan karakter dan data yang digunakan oleh *programmer* komputer untuk menjelaskan bagan alur dari program."

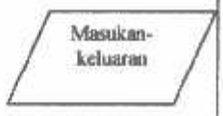
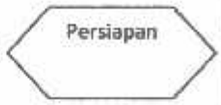
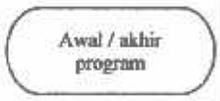
Menurut Jogiyanto H.M (1999), *flowchart* atau juga disebut dengan bagan alir program (program *flowchart*), adalah :

"Merupakan bagan alir yang menjelaskan secara rinci langkah – langkah dari proses program."

Berikut simbol – simbol yang digunakan untuk pembuatan *flowchart* menurut Jogiyanto H.M (1999) pada tabel 2.8 sebagai berikut :

Tabel 2.8 Simbol – Simbol *Flowchart*

SIMBOL	URAIAN
	Gambar kotak persegi menunjukkan proses yang digunakan untuk mewakili suatu proses.
 Garis Alir	Gambar anak panah menunjukkan aliran data atau arus dari proses.
 Penghubung	Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang sama atau lainnya

	Menunjukkan input atau output yang digunakan untuk mewakili data input atau output.
	Digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.
	Digunakan untuk suatu penyelesaian kondisi di dalam program.
	Digunakan untuk memberikan nilai awal suatu besaran
	Atau disebut juga simbol titik terminal yang digunakan untuk menunjukkan awal dan akhir dari suatu proses.

Sehingga dapat disimpulkan bahwa *flowchart* adalah penggambaran aliran data dan karakter dari bagan alir program yang menjelaskan secara rinci langkah – langkah dari proses program.

2.5 Bahasa Pemrograman Borland Delphi 2010

Borland Delphi merupakan salah satu IDE untuk membangun aplikasi *desktop* menggunakan bahasa pemrograman *Pascal*. *Pascal* adalah bahasa pemrograman yang cukup populer karena nyaris memiliki struktur seperti bahasa manusia. [4]

Delphi 2010 adalah aplikasi utama mengembangkan lingkungan secara cepat dengan membangun aplikasi berkinerja tinggi. *Delphi 2010 Architect* dirancang untuk para pengembang dan tim membangun data-intensif klien / *server* GUI dan aplikasi *web* dengan sistem *database* yang besar atau kompleks. *Delphi Architect* menggabungkan pengembangan aplikasi berbasis data yang cepat dan *heterogen* kecepatan akses data yang tinggi dengan pemodelan data *visual* yang kaya untuk membantu para pengembang memahami, desain, dan mendapatkan nilai yang banyak dari *database* perusahaan yang ada dan struktur data[2].

Borland Delphi atau biasa disebut *Delphi* merupakan perangkat lunak pengembangan aplikasi yang sangat populer di lingkungan *Windows*. Perangkat lunak ini dapat digunakan untuk membuat aplikasi apa saja, dari permainan hingga ke aplikasi basis data[4].

Semenjak versi 6, *Delphi* telah dilengkapi dengan sejumlah komponen yang tergolong sebagai *dbExpress*, yang memungkinkan koneksi ke *MySQL* ataupun *Oracle* dilakukan dengan mudah, sehingga *Delphi* dapat digunakan sebagai aplikasi *front-end* yang berhubungan dengan *database server*. Sedangkan pada versi 7, komponen yang tergolong sebagai *dbExpress* sedikit berubah.[4]

Perubahan umum pada IDE (*Integrated Development Environment*) adalah sebagai berikut :[4]

- Menu *View* | *Additional Message Info* menampilkan jendela *message hint*, yaitu informasi tambahan tentang *compiler message* yang dapat di *download* dari *website Borland*.
- Perubahan pada *component pallete*.
- Jika membuat aplikasi baru untuk CLX, pada *page system* akan ditampilkan beberapa komponen yang berhubungan dengan file dan direktori.
- Penambahan page *Indy Intercepts* dan *Indy I/O Handler* Open Source untuk komponen *Internet Protocol* pada edisi Professional dan *Enterprise*.

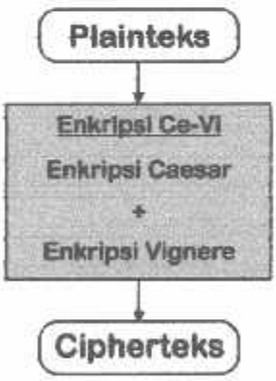
- Penambahan page *IW Standart, IW Data, IW Client Side, IW Control page* yang menyediakan komponen – komponen *IntraWeb* untuk pengembangan aplikasi berbasis *web*.
- Penambahan page *Rave* yang menyediakan komponen – komponen untuk keperluan pembuatan *report*.
- *Code Completion* yang lebih cepat. *Customize code completion manager* menggunakan OpenTools API.

BAB III
ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Masalah

Dalam merancang suatu sistem diperlukan analisis terhadap sistem yang akan dirancang tersebut terlebih dahulu. Tujuan dari analisis ini sendiri adalah agar sistem yang dirancang menjadi tepat guna dan ketahanan sistem tersebut akan lebih terjaga. Di samping itu dengan dilakukannya analisis mempermudah pekerjaan dalam membuat sistem, dan jika suatu saat nanti ada perbaikan atau penambahan dalam sistem tersebut, maka akan mudah diselesaikan.

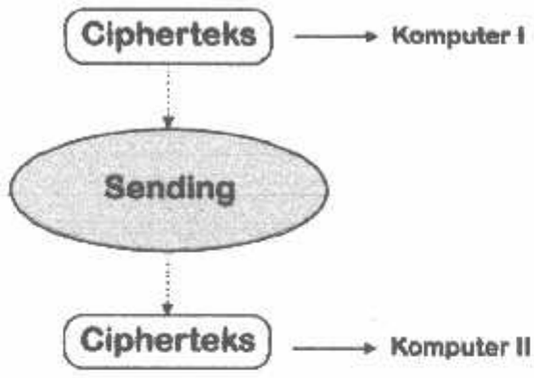
Sistem yang dirancang terdiri dari dua proses secara garis besar, yaitu proses enkripsi ataupun dekripsi dan pengiriman data (*Sending*). Dimana untuk proses enkripsi atau dekripsi menggunakan algoritma kriptografi Ce-Vi. Kriptografi Ce-Vi sendiri merupakan kombinasi dan pengembangan dua algoritma klasik yaitu algoritma kriptografi *Caesar* dan algoritma kriptografi *Vignere*. Serta penggunaan karakter yang digunakan menjadi lebih luas lagi, yang semula hanya mencakup 26 karakter menjadi 95 karakter yang meliputi karakter huruf besar (A..Z), huruf kecil (a..z), angka(0..9) dan simbol – simbol (~! ... +\). Sedangkan untuk pengiriman data, akan menggunakan komponen *Socket* (pada Bahasa Pemrograman *Delphi*) meliputi *ServerSocket* dan *ClientSocket*, yang mana berfungsi sebagai media komunikasi antara komputer yang satu dengan yang lain (*Chatting*). Skema global untuk proses Enkripsi dapat dilihat pada gambar 3.1.



Gambar 3.1 Skema Enkripsi Algoritma Kriptografi Ce-Vi (*Caesar - Vignere*)

Dari gambar di atas dapat dilihat bahwa plainteks akan dienkripsi menggunakan algoritma Ce-Vi, dimana nanti di dalam algoritma Ce-Vi ini ada dua proses enkripsi

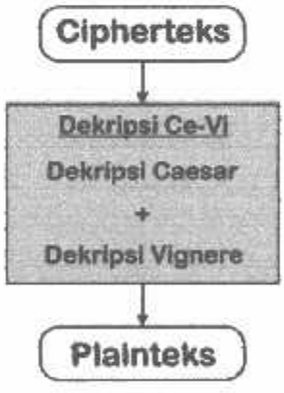
yaitu enkripsi *Caesar* dan enkripsi *Vignere* yang sudah menjadi satu. Sehingga nantinya akan menghasilkan suatu chiperteks yang akan dikirimkan ke komputer lain. Skema global untuk proses pengiriman seperti 3.2



Gambar 3.2 Skema Pengiriman (*Sending*)

Dari gambar di atas dapat dilihat bahwa hasil proses enkripsi tadi akan dikirimkan (*sending*) ke komputer lain atau yang dituju. Sehingga data atau informasi yang melewati suatu jaringan berupa pesan rahasia yang memungkinkan pihak lain tidak mengerti makna pesan tersebut.

Di samping kedua proses di atas juga dilengkapi proses Dekripsi. Hal ini bertujuan agar data atau informasi yang sudah terenkripsi tadi dapat kembali menjadi informasi awal yang dapat dibaca dan dimengerti oleh pihak yang berhak seperti gambar 3.3.



Gambar 3.3 Skema Proses Dekripsi

3.1.1 Analisa Algoritma Caesar

Prinsip kerja algoritma *Caesar* ini adalah mengganti (mensubtitusi) setiap karakter dengan karakter lain sebanyak *key(k)*, dimana *k* pada *Caesar Cipher* ini bernilai tiga (*k=3*). Sehingga setiap karakter akan digeser sebanyak tiga kali. Namun perkembangannya sekarang pemakaian nilai *key* sudah tidak terpaku lagi pada *key Caesar Cipher*, bisa juga bernilai k=1, k=2, k= 3....dst. Dalam hal ini terjadi perubahan karakter menjadi 95 karakter sehingga menghasilkan persamaan enkripsi dan dekripsi sebagai berikut :

- Proses Enkripsi :

  C = 32+(a+3)mod 95 ..... **(3-1)**

- Proses Dekripsi :

  P = (b – 3 + 95) mod 95 + 32 ..... **(3-2)**

  dimana :

  P = Plainteks

  C = Cipherteks

  k = pergeseran *key*

  a = P – 32

  b = C – 32

  mod 95 = modulus 95 karakter

Tabel 3.1 menampilkan hasil lengkap karakter pada *Caesar Cipher*, sedangkan Tabel 3.2 menunjukkan nilai - nilai tiap karakter yang akan digunakan. Nilai – nilai karakter ini disesuaikan dengan nilai pada ANSI karakter.

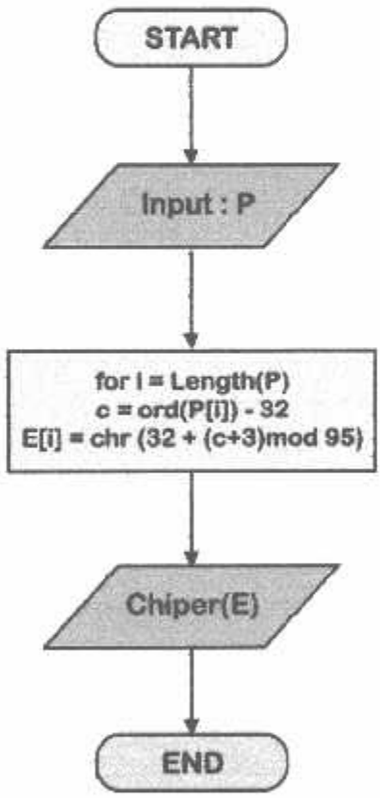
Tabel 3.1 Subtitusi *Caesar Cipher* (95 karakter)

Plainteks	' '	!	"	#	$	%	&	'	(	)	*	+	,
Cipherteks	#	$	%	&	'	(	)	*	+	,	-	.	/
Plainteks	-	.	/	0	1	2	3	4	5	6	7	8	9
Cipherteks	0	1	2	3	4	5	6	7	8	9	:	;	<
Plainteks	:	;	<	=	>	?	@	A	B	C	D	E	F
Cipherteks	=	>	?	@	A	B	C	D	E	F	G	H	I
Plainteks	G	H	I	J	K	L	M	N	O	P	Q	R	S
Cipherteks	J	K	L	M	N	O	P	Q	R	S	T	U	V
Plainteks	T	U	V	W	X	Y	Z	[	\	]	^	_	`
Cipherteks	W	X	Y	Z	[	\	]	^	_	`	a	b	c
Plainteks	a	b	c	d	e	f	g	h	i	j	k	l	m
Cipherteks	d	e	f	g	h	i	j	k	l	m	n	o	p
Plainteks	n	o	p	q	r	s	t	u	v	w	x	y	z
Cipherteks	q	r	s	t	u	v	w	x	y	z	{	|	}
Plainteks	{	|	}	~									
Cipherteks	~	' '	!	"									

Tabel 3.2 Nilai Tiap Karakter Pada *Caesar Cipher* (95 Karakter)

Kunci	' '	!	"	#	$	%	&	'	(	)	*	+	,
Pergeseran k	32	33	34	35	36	37	38	39	40	41	42	43	44
Kunci	-	.	/	0	1	2	3	4	5	6	7	8	9
Pergeseran k	45	46	47	48	49	50	51	52	53	54	55	56	57
Kunci	:	;	<	=	>	?	@	A	B	C	D	E	F
Pergeseran k	58	59	60	61	62	63	64	65	66	67	68	69	70
Kunci	G	H	I	J	K	L	M	N	O	P	Q	R	S
Pergeseran k	71	72	73	74	75	76	77	78	79	80	81	82	83
Kunci	T	U	V	W	X	Y	Z	[	\	]	^	_	`
Pergeseran k	84	85	86	87	88	89	90	91	92	93	94	95	96
Kunci	a	b	c	d	e	f	g	h	i	j	k	l	m
Pergeseran k	97	98	99	100	101	102	103	104	105	106	107	108	109
Kunci	n	o	p	q	r	s	t	u	v	w	x	y	z
Pergeseran k	110	111	112	113	114	115	116	117	118	119	120	121	122
Kunci	{	|	}	~									
Pergeseran k	123	124	125	126									

Berikut *flowchart* enkripsi *Caesar Cipher* seperti gambar 3.4.

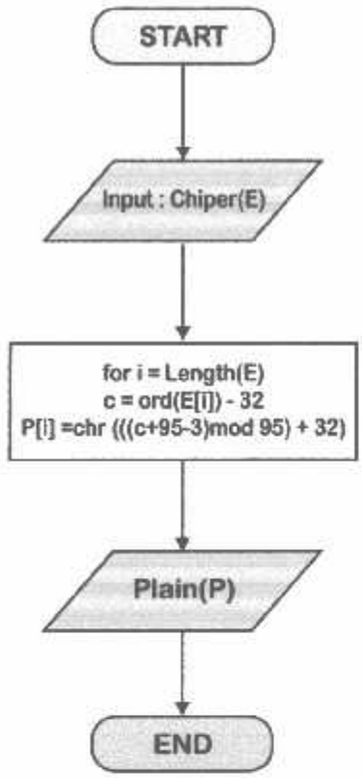


Gambar 3.4 *Flowchart* Enkripsi *Caesar Cipher*

Dalam algoritma *Caesar Cipher* terdapat variabel – variabel dan fungsi sebagai berikut :

1. **P** dan **E** merupakan suatu masukan atau inputan yang akan digunakan sebagai pesan (plainteks) ataupun pesan rahasia (cipherteks) berupa karakter bebas (sesuai dengan jumlah karakter pada *keyboard*),tipe dari variabel ini berupa *string*.
2. **for i = Length(P)**. Menghitung panjangnya variabel pesan (plainteks), sedangkan variabel *i* bertipe *integer*;
3. **for i = Length(E)**. Menghitung panjangnya variabel pesan rahasia (cipherteks), sedangkan variabel *i* bertipe *integer*;
4. **c = ord(P[i]) – 32**, Mengubah *char* menjadi kode ASCII, variabel *k* bertipe *byte*. 32 merupakan batasan dari pengambilan karakter pada ASCII.
5. **E[i] = chr (32+(c+3)mod 95)**. Melakukan proses enkripsi dengan mengubah kode ASCII ke bentuk *char*. Mod 95 merupakan nilai sisa dari 95, sedangkan 95 ialah banyaknya karakter yang digunakan.
6. **P[i] = chr (((c+95-3)mod 95) + 32).** Melakukan proses dekripsi dengan mengubah kode ASCII ke bentuk *char*. Mod 95 merupakan nilai sisa dari 95, sedangkan 95 ialah banyaknya karakter yang digunakan

Proses pendekripsian, dilakukan dengan cara mengembalikan pergeseran *key* ke bentuk semula. Jika pada waktu mengenkripsi dengan menggeser karakter ke arah kanan sebanyak *key*, maka pada proses dekripsi menggeser karakter ke arah berlawanan sebanyak *key* yang sama juga, begitu pula sebaliknya. Berikut *flowchart* dekripsi *Caesar Cipher* seperti gambar 3.5.



Gambar 3.5 *Flowchart* Dekripsi *Caesar Cipher*

3.1.2 Analisa Algoritma Vignere

Prinsip kerja algoritma *Vignere Cipher* hampir sama dengan algoritma *Caesar Cipher* yaitu mensubtitusi atau mengganti setiap karakter dengan karakter lain. Perbedaan terletak pada pergeseran *key*, pergeseran *key* pada algoritma *Vignere Cipher* tergantung dari nilai desimal tiap – tiap karakter pada *key*, sehingga setiap huruf cipherteks dapat memiliki kemungkinan banyak huruf plainteks. Dalam hal ini terjadi perubahan karakter menjadi 95 karakter sehingga menghasilkan persamaan enkripsi dan dekripsi sebagai berikut :

- Proses Enkripsi :

  C = 32+(a+l)mod 95 ..... **(3-3)**

- Proses Dekripsi :

  P = (b – l + 95) mod 95 + 32 ..... **(3-4)**

  dimana :

  P = Plainteks

  C = Cipherteks

  k = pergeseran *key*

  a = P – 32

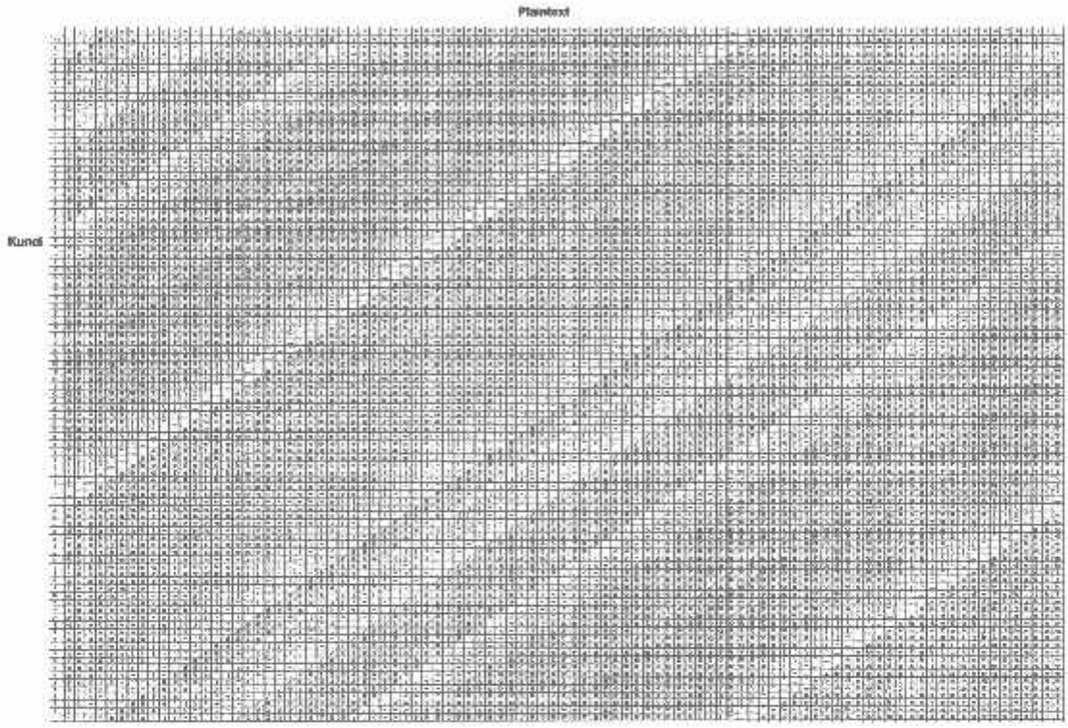
b = C – 32

l = k - 32

mod 95 = modulus 95 karakter

Tabel 3.3 menampilkan hasil lengkap karakter pada *Vignere Cipher*, sedangkan Tabel 3.4 menunjukkan nilai - nilai tiap karakter yang akan digunakan. Nilai – nilai karakter ini disesuaikan dengan nilai pada ANSI karakter.

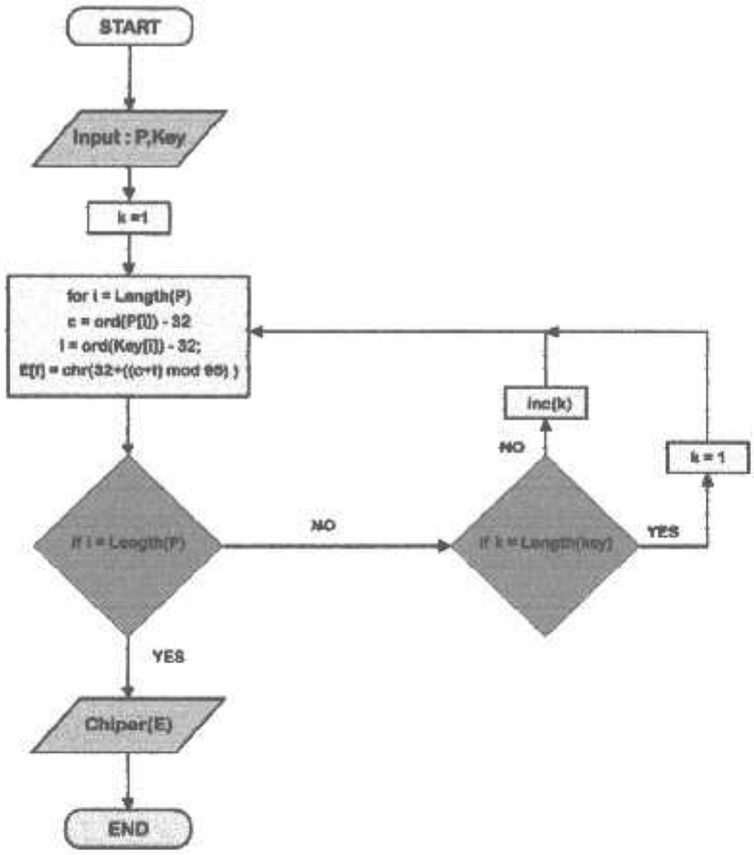
Tabel 3.3 Subtitusi *Vignere Cipher* (95 karakter)



Tabel 3.4 Nilai Tiap Karakter Pada *Vignere Cipher* (95 Karakter)

Kunci	' '	!	"	#	$	%	&	'	(	)	*	+	,
Pergeseran k	32	33	34	35	36	37	38	39	40	41	42	43	44
Kunci	-	.	/	0	1	2	3	4	5	6	7	8	9
Pergeseran k	45	46	47	48	49	50	51	52	53	54	55	56	57
Kunci	:	;	<	=	>	?	@	A	B	C	D	E	F
Pergeseran k	58	59	60	61	62	63	64	65	66	67	68	69	70
Kunci	G	H	I	J	K	L	M	N	O	P	Q	R	S
Pergeseran k	71	72	73	74	75	76	77	78	79	80	81	82	83
Kunci	T	U	V	W	X	Y	Z	[	\	]	^	_	`
Pergeseran k	84	85	86	87	88	89	90	91	92	93	94	95	96
Kunci	a	b	c	d	e	f	g	h	i	j	k	l	m
Pergeseran k	97	98	99	100	101	102	103	104	105	106	107	108	109
Kunci	n	o	p	q	r	s	t	u	v	w	x	y	z
Pergeseran k	110	111	112	113	114	115	116	117	118	119	120	121	122
Kunci	{	|	}	~									
Pergeseran k	123	124	125	126									

Berikut *flowchart* enkripsi *Vignere Cipher* seperti 3.6.



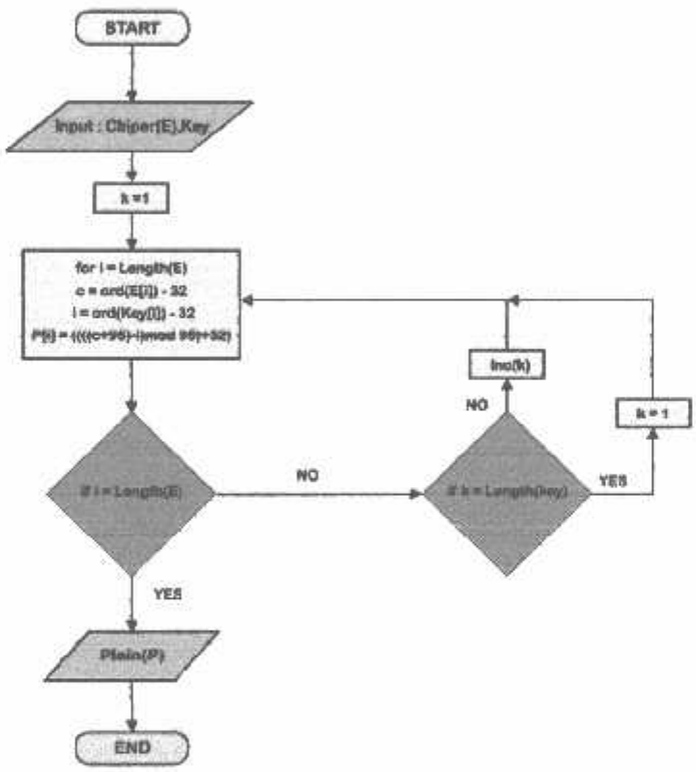
Gambar 3.6 *Flowchart* Enkripsi *Vignere Cipher*

Dalam algoritma *Vignere* terdapat variabel dan fungsi sebagai berikut :

1. **P,Key** dan E. Merupakan imputan yang berupa karakter bebas bertipe *string*.
2. **k = 1**. Merupakan nilai awal yang digunakan untuk menghitung panjang *key*, *k* bertipe *integer*.
3. **for i = Length(P)**. Menghitung panjang karakter (P), sedangkan *i* bertipe *integer*
4. **for i = Length(E)**. Menghitung panjang karakter (E), sedangkan *i* bertipe *integer*
5. **c = ord (P[i]) – 32.** Mengubah *char* menjadi kode ASCII, variabel *c* bertipe *byte*. 32 merupakan batasan dari pengambilan karakter pada ASCII.
6. **l = ord(Key[i]) – 32**Mengubah char menjadi kode ASCII,, variabel *l* bertipe *byte*. 32 merupakan batasan dari pengambilan karakter pada ASCII.

7. **E[i] = chr (32+((c+l)mod 95)).** Melakukan proses enkripsi dengan mengubah kode ASCII ke bentuk char. Mod 95 merupakan nilai sisa dari 95, sedangkan 95 ialah banyaknya karakter yang digunakan.
8. **P[i] =chr((((c+95)-l)mod 95)+32).** Melakukan proses dekripsi dengan mengubah kode ASCII ke bentuk char. Mod 95 merupakan nilai sisa dari 95, sedangkan 95 ialah banyaknya karakter yang digunakan.
7. **inc(k).** Merupakan penambahan nilai variabel *k* sebanyak satu kali (*k=k+1*).

Proses pendekripsian pada algoritma *Vignere* ini, dilakukan dengan cara mengembalikan pergeseran *key* ke bentuk semula sesuai dengan nilai desimal tiap – tiap karakter pada *key* yang dimasukan. Berikut *flowchart* dekripsi *Vignere Cipher* seperti gambar 3.7.



Gambar 3.7 *Flowchart* Dekripsi *Vignere Cipher*

3.1.3 Analisa Algoritma Ce-Vi

Prinsip kerja algoritma Ce-Vi ini mengkombinasikan dua metode diatas yaitu algoritma *Caesar* dan algoritma *Vignere*, Secara umum dapat dituliskan persamaan gabungan sebagai berikut dengan menggunakan 26 karakter :

- Proses Enkripsi

$$E_{Ce\text{-}Vi} = E_{Vigenere}(E_{Caesar}) \quad \text{(3-5)}$$

$$E_{Ce\text{-}Vi} = (((P+3) \bmod 26)+k) \bmod 26 \quad \text{(3-6)}$$

- Proses Dekripsi

$$D_{Ce\text{-}Vi} = D_{Caesar}(D_{Vigenere}) \quad \text{(3-7)}$$

$$D_{Ce\text{-}Vi} = (((C-k+26) \bmod 26) - k + 26) \bmod 26 \quad \text{(3-8)}$$

Dikarenakan dalam hal ini menggunakan 95 karakter maka akan berubah menjadi persamaan gabungan sebagai berikut :

- Proses Enkripsi

$$E_{Ce\text{-}Vi} = E_{Vigenere}(E_{Caesar}) \quad \text{(3.9)}$$

$$E_{Ce\text{-}Vi} = 32+((32+(a+3) \bmod 95)+l) \bmod 95 \quad \text{(3.10)}$$

Dengan catatan a = P – 32 dan l = k - 32

- Proses Dekripsi

$$D_{Ce\text{-}Vi} = D_{Caesar}(D_{Vigenere}) \quad \text{(3-11)}$$

$$D_{Ce\text{-}Vi} = (((b - l + 95) \bmod 95 + 32) - 3 + 95) \bmod 95 + 32 \quad \text{(3-12)}$$

Dengan catatan b = C – 32 dan l = k - 32

Adapun persamaan secara lengkap pada *coding* sebagai berikut untuk penggunaan 95 karakter :

$$E = chr\,(32 + ((c + 3)) \bmod 95) \quad \text{(3-13)}$$

Merupakan persamaan enkripsi algoritma *Caesar*.

$$E = chr\,(32+((c+k) \bmod 95)) \quad \text{(3-14)}$$

Merupakan persamaan enkripsi algoritma *Vignere*

Apabila kedua persamaan diatas digabungkan menjadi

$$E = chr\,(32 + ((c+k+3) \bmod 95)) \quad \text{(3-15)}$$

Langkah pertama pada algoritma Ce-Vi ini ialah memasukkan suatu masukan berupa karakter bebas yang mana akan dijadikan pesan (plainteks) yang akan dikirim, kemudian masukkan juga *key* (berupa karakter bebas) yang akan digunakan untuk proses enkripsi maupun dekripsi. Sehingga nantinya akan menghasilkan suatu cipherteks, begitu pula sebaliknya untuk proses dekripsi

Sedangkan untuk mendekripsikan algoritma Ce-Vi ini ialah proses sebaliknya, menggeser tiap–tiap karakter pada *key*, sehingga membentuk persamaan sebagai berikut untuk penggunaan 95 karakter :

P = chr (((c+95-3)mod 95)+32) ..... **(3-16)**

Merupakan persamaan dekripsi algoritma *Caesar*.

P = chr((((c+95)-k)mod 95)+32) ..... **(3-17)**

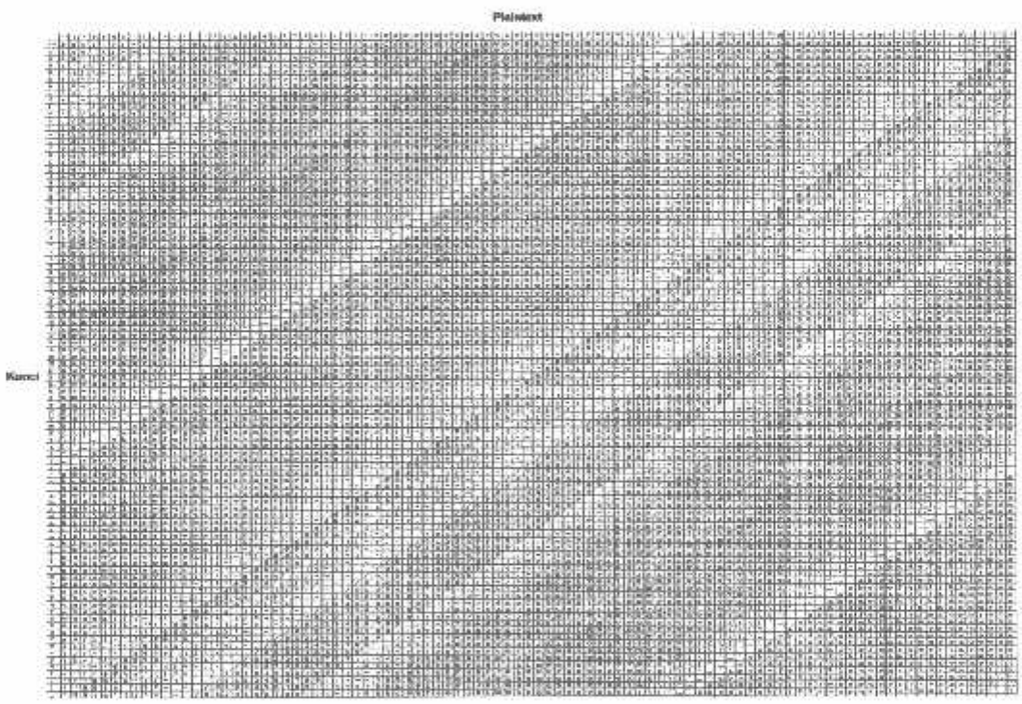
Merupakan persamaan dekripsi algoritma *Vignere*.

Apabila kedua persamaan diatas digabungkan menjadi

P = chr ((((c+95)-k-3)mod 95)+32) ..... **(3-18)**

Untuk mempermudah dalam penggunaan metode Ce-Vi ini maka dibuatlah Tabel 3.5 yang menampilkan hasil keseluruhan karakter yang digunakan pada metode ini serta Tabel 3.6 yang menunjukkan nilai–nilai tiap karakter. Nilai–nilai karakter ini disesuaikan dengan nilai pada ANSI karakter.

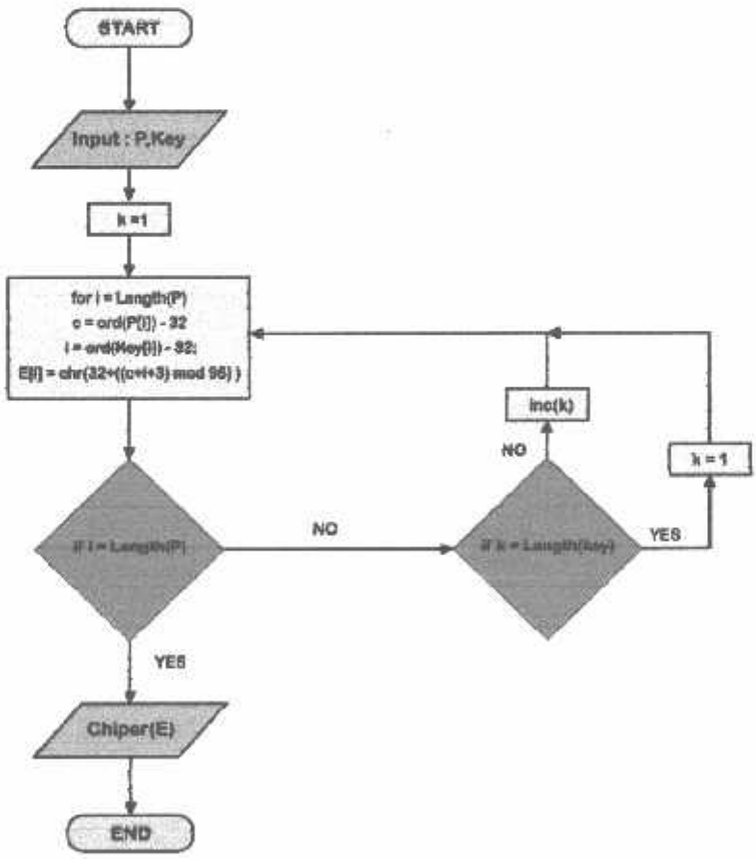
Tabel 3.5 Subtitusi Ce-Vi *Cipher* (95 karakter)



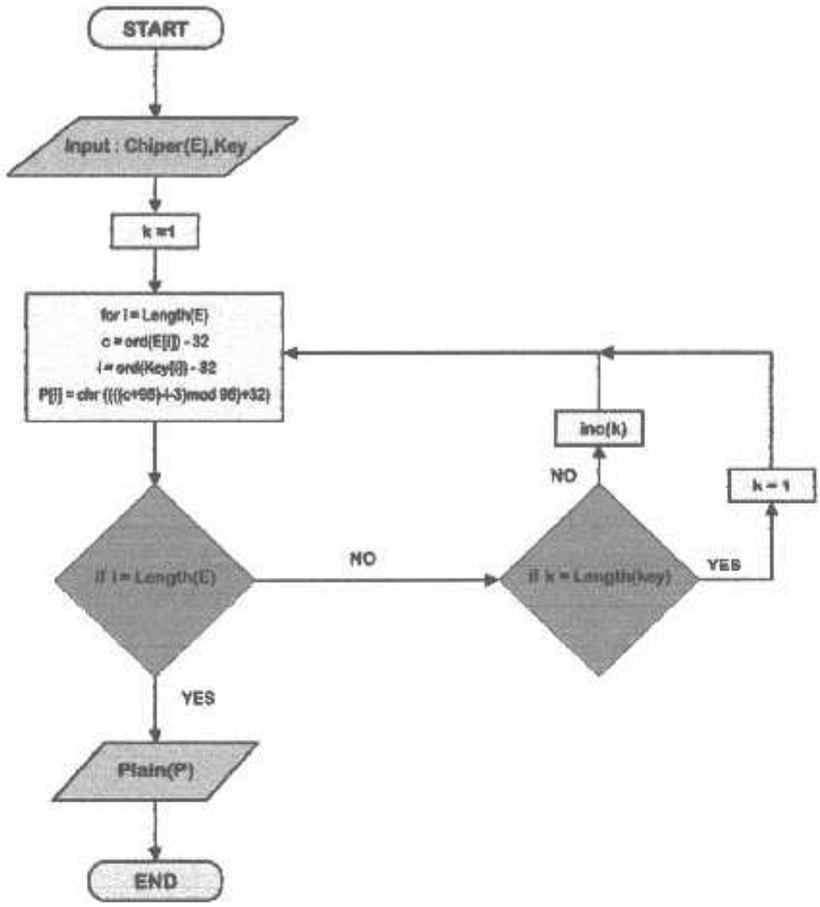
Tabel 3.6 Nilai Tiap Karakter Pada Ce-Vi *Cipher* (95 Karakter)

Kunci	' '	!	"	#	$	%	&	'	(	)	*	+	,
Pergeseran k	32	33	34	35	36	37	38	39	40	41	42	43	44
Kunci	-	.	/	0	1	2	3	4	5	6	7	8	9
Pergeseran k	45	46	47	48	49	50	51	52	53	54	55	56	57
Kunci	:	;	<	=	>	?	@	A	B	C	D	E	F
Pergeseran k	58	59	60	61	62	63	64	65	66	67	68	69	70
Kunci	G	H	I	J	K	L	M	N	O	P	Q	R	S
Pergeseran k	71	72	73	74	75	76	77	78	79	80	81	82	83
Kunci	T	U	V	W	X	Y	Z	[	\	]	^	_	`
Pergeseran k	84	85	86	87	88	89	90	91	92	93	94	95	96
Kunci	a	b	c	d	e	f	g	h	i	j	k	l	m
Pergeseran k	97	98	99	100	101	102	103	104	105	106	107	108	109
Kunci	n	o	p	q	r	s	t	u	v	w	x	y	z
Pergeseran k	110	111	112	113	114	115	116	117	118	119	120	121	122
Kunci	{	|	}	~									
Pergeseran k	123	124	125	126									

Berikut *flowchart* algoritma enkripsi Ce-Vi dan algoritma dekripsi Ce-Vi seperti gambar 3.8 dan gambar 3.9.



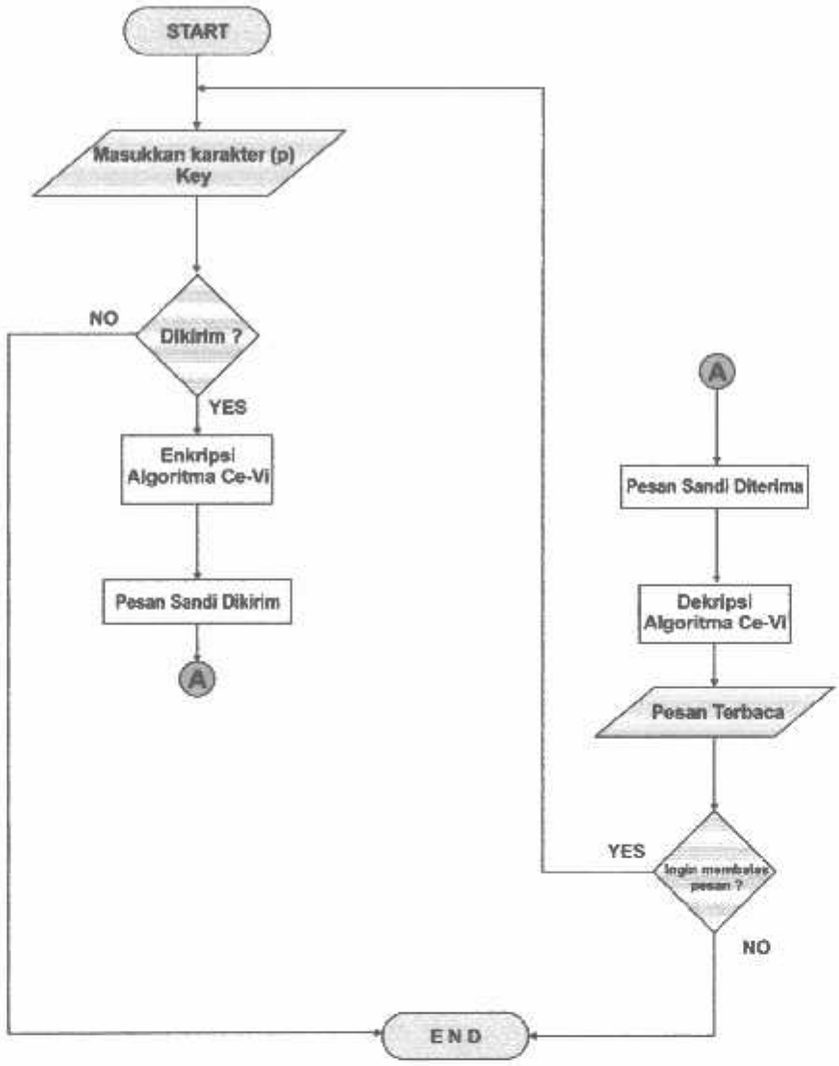
Gambar 3.8 *Flowchart* Enkripsi Ce-Vi *Cipher*



Gambar 3.9 Flowchart Dekripsi Ce-Vi *Cipher*

3.1.4 Analisa Chatting

Berikut algoritma kerja program *chatting* yang sudah dimasukkan algoritma Ce-Vi. Pesan yang berisi beberapa karakter ini bisa berupa huruf alfabet kecil (a...z) atau huruf alfabet besar (A...Z), angka (0..9) ataupun simbol – simbol (~! ... +\). Sebelum pesan dikirim, pesan terlebih dahulu di enkripsi, setelah selesai menjadi chiperteks maka cipherteks tersebut akan dikirim ke penerima. Pada penerima, cipherteks tersebut akan di dekripsi menghasilkan pesan asli (plainteks) sehingga dapat dibaca kembali. Pada program ini tidak bisa menerima masukkan berupa suara (*audio*), *video* ataupun berupa *file*. Berikut *flowchart* pada *chatting* seperti gambar 3.10



Gambar 3.10 *Flowchart Chatting*

Proses selanjutnya adalah mengirimkan informasi atau data (*sending*). Langkah – langkah dari proses pengiriman (*sending*) adalah :

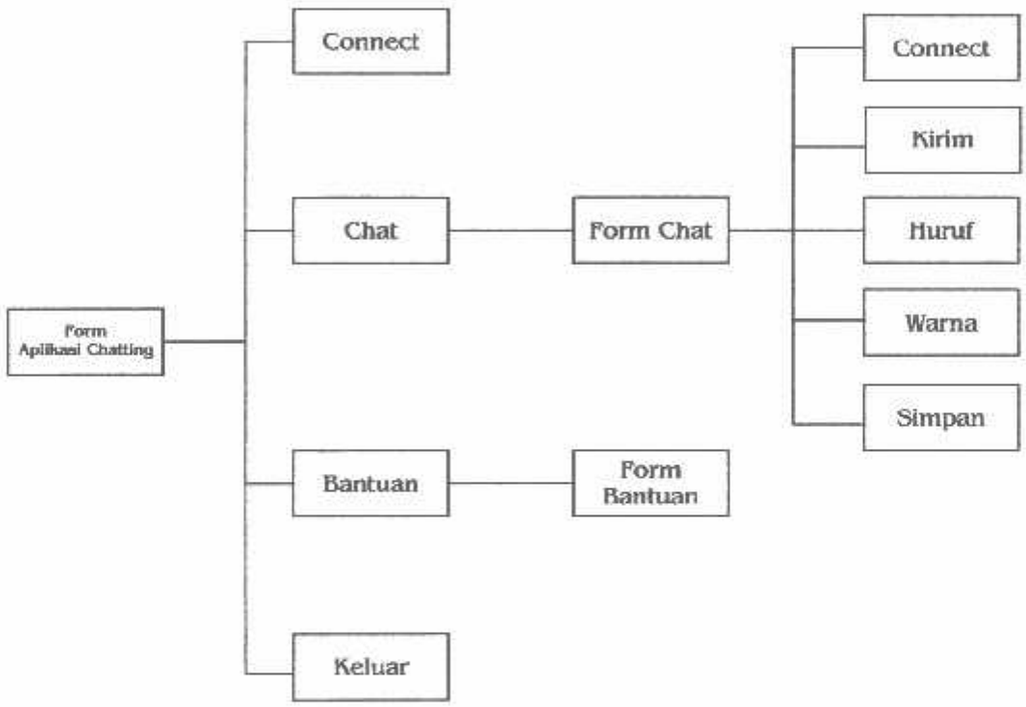
1. Memasukkan pesan yang dikirim dan sebuah *key*.
2. Setelah dikirim, pesan tersebut akan di enkripsi menggunakan *key* yang sudah dimasukkan sehingga menghasilkan pesan rahasia (cipherteks).
3. Cipherteks tersebut akhirnya dikirim ke penerima.
4. Pada penerima pesan rahasia (cipherteks) tersebut akan dikembalikan lagi ke pesan awal sehingga bisa terbaca dan dimengerti isi pesan tersebut oleh penerima.
5. Bila si penerima ingin membalas pesan maka akan melakukan proses yang sama.

3.2 Dekripsi Sistem

Sistem Algoritma Ce-Vi pada aplikasi *Chatting* ini ialah salah satu solusi untuk mengamankan kerahasiaan suatu informasi dalam percakapan antara dua entitas (pengirim dan penerima). Sistem ini merupakan kombinasi serta pengembangan algoritma *Caesar* dan algoritma *Vignere*. Dimana penggabungan dua algoritma tersebut akan diintegrasikan atau diterapkan pasa aplikasi *chatting*. Tujuan dari sistem ini adalah menjaga kerahasiaan informasi percakapan (*chat*) yang lebih baik daripada penggunaan satu algoritma kriptografi saja.

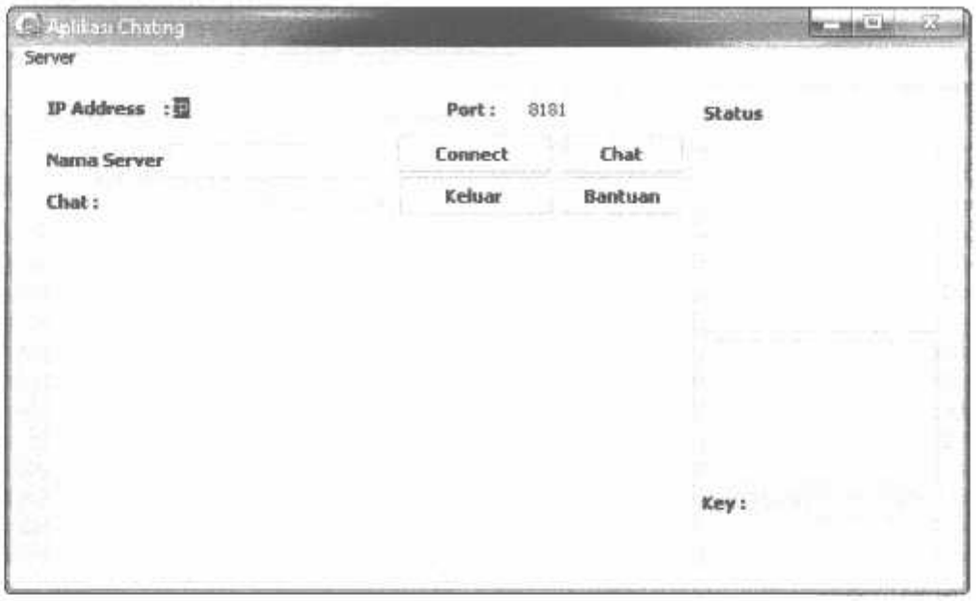
3.3 Perancangan Sistem

3.3.1 Desain Menu



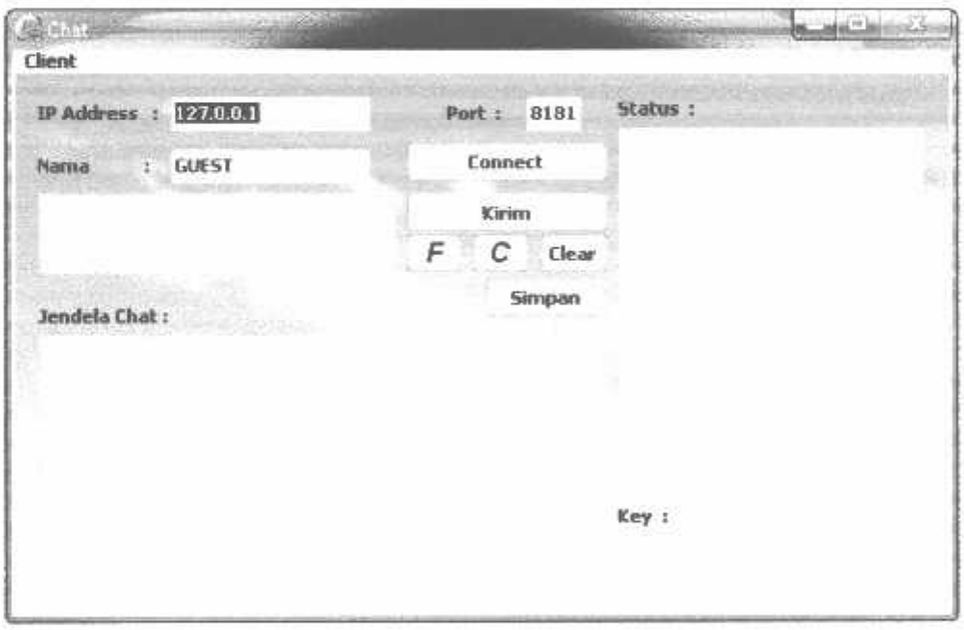
Gambar 3.11 Desain Form Aplikasi *Chatting* Ce-Vi

3.3.2 Desain Form Aplikasi Chatting



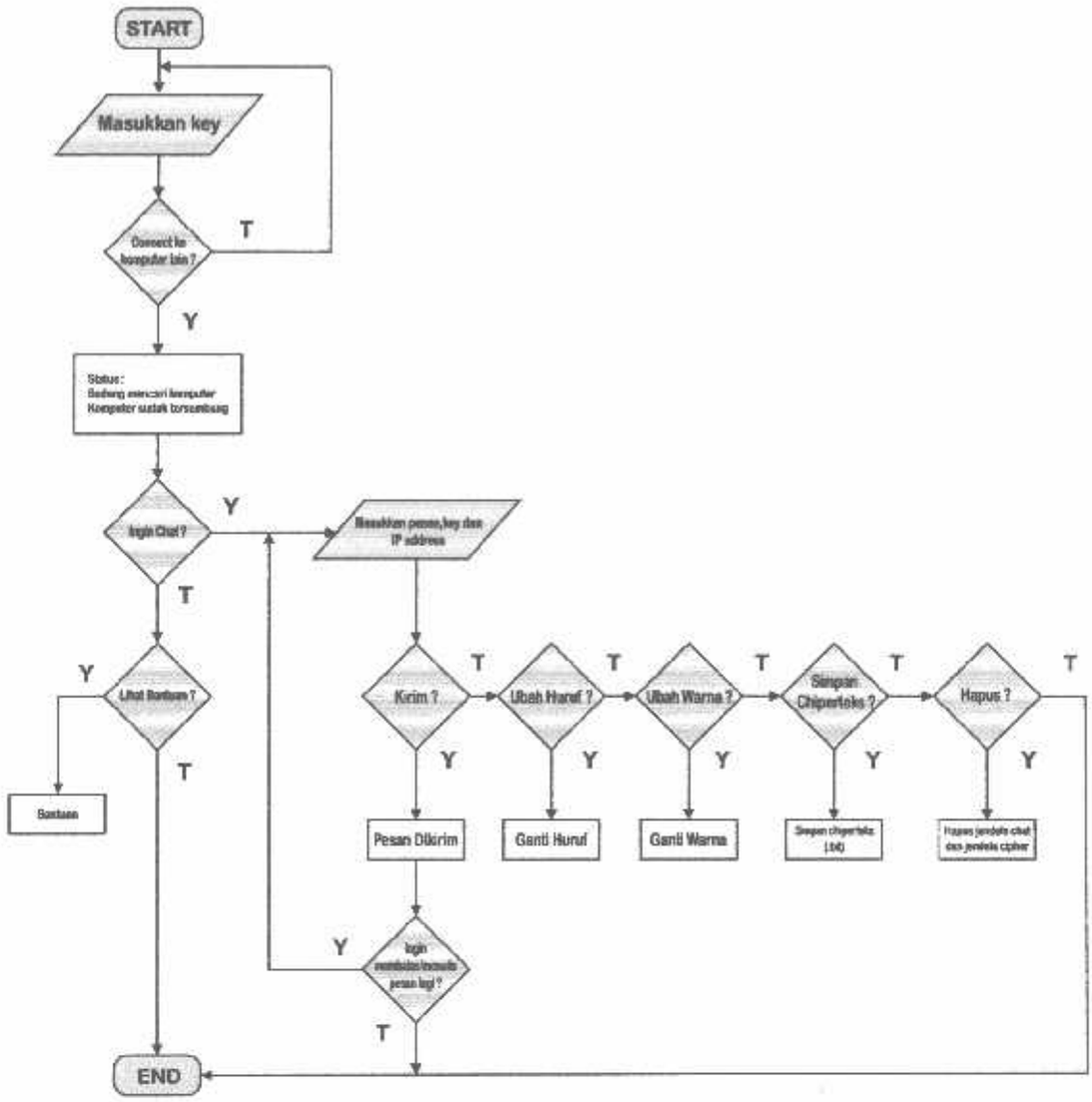
Gambar 3.12 *Form* Aplikasi *Chatting*

3.3.3 Desain Form Chat



Gambar 3.13 *Form Chat*

3.3.4 Flowchart



Gambar 3.14 *Flowchart* Aplikasi *Chatting* Ce-Vi

BAB IV
IMPLEMENTASI

4.1 Kebutuhan Hardware

Perangkat keras (*hardware*) adalah semua alat komputer yang objeknya *real* seperti monitor, CPU dll. Perangkart keras yang dibutuhkan harus di-*install* terlebih dahulu agar Sistem Operasi *Windows* bisa digunakan. Rincian *Software* dan *Hardware* pendukung seperti pada tabel 4.1 sebagai berikut :

Tabel 4.1 Spesifikasi Implementasi Perlengkapan

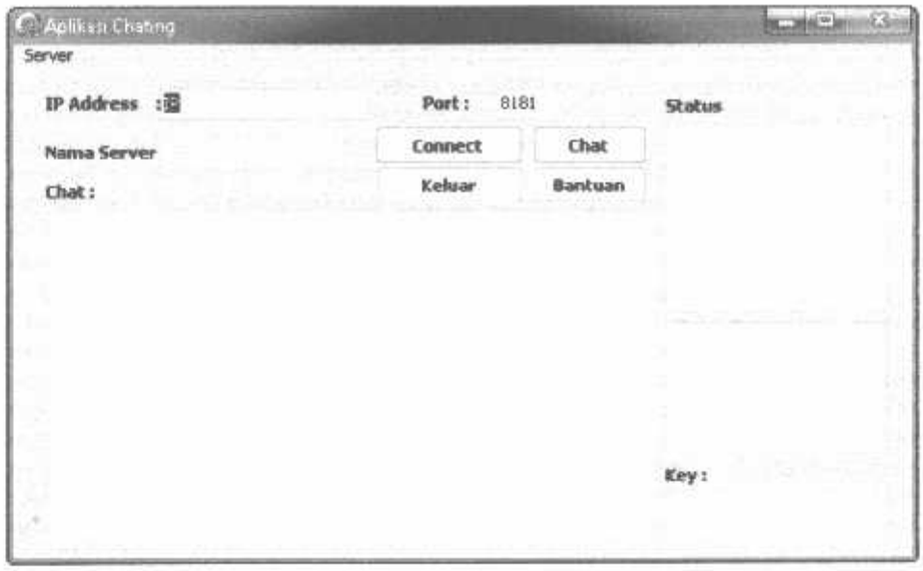
NO	Perlengkapan	Spesifikasi	Keterangan
1	Software	Sistem Operasi	Windows XP Service Pack 2
		Bahasa Pemrograman	Borland Delphi 2010 Architect
2	Personal Komputer	Processor	Pentium (R) Dual. 2.2 GHz
		Memori	2 Gb DDR2
		Hardisk	40 Gb
		Kartu Grafis (VGA Card)	Min. 32 Mbyte

4.2. Implementasi Sistem

Implementasi dilakukan dengan menerapkan hasil desain yang telah dbuat ke dalam bahasa pemrograman (*Coding*) *Borland Delphi 2010 Architect*, sehingga prosedur – prosedur yang telah dibuat dapar dimengerti oleh mesin schingga menghasilkan keluaran seperti yang diharapkan.

4.2.1 Form Aplikasi Chatting

Form ini bentindak sebagai form utama, jadi ketika program pertama kali dijalankan akan muncul Form Aplikasi *Chatting*. Gambar 4.1 menunjukkan tampilan *form* pada aplikasi *chatting*.

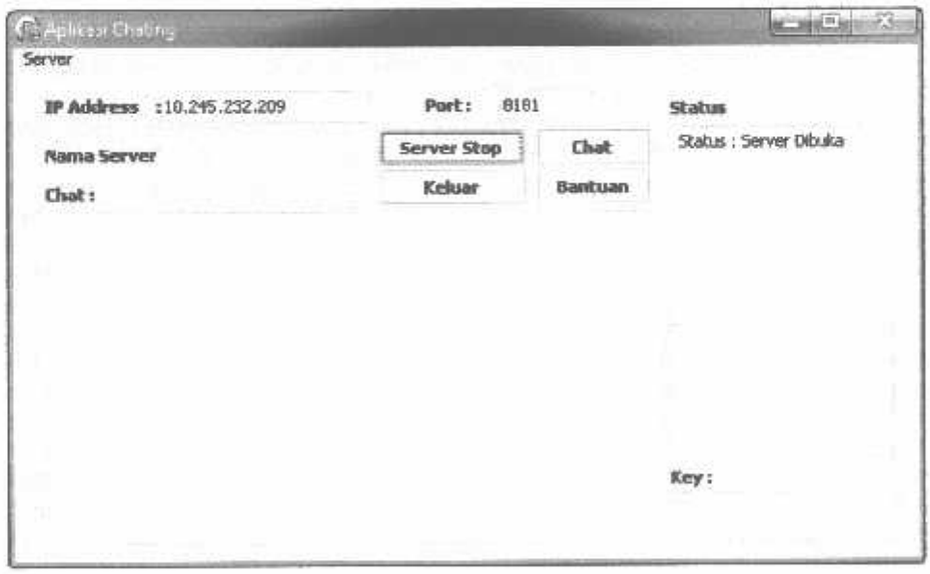


Gambar 4.1 Tampilan *Form* Aplikasi *Chatting*

Adapun fungsi masing – masing tombol (*button*) pada *Form* Aplikasi *Chatting* sebagai berikut :

1. **Connect** Connect

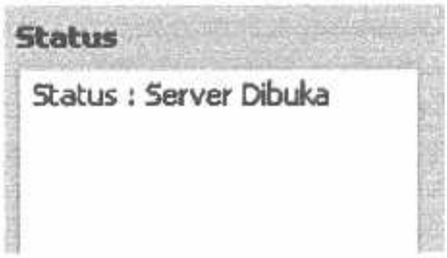
   Tombol ini berguna untuk membuka *port* pada komputer yang membuka koneksi ke jaringan agar komputer lain bisa terhubung. Berikut tampilan saat tombol *Connect* ditekan seperti pada gambar 4.2



Gambar 4.2 Tampilan *Form* Saat Tombol *Connect* Ditekan

Kerika tombol *Connect* ditekan, maka *port* 8181 (dengan alamat IP : xxx.xxx.xxx.xxx)akan dibuka, *port* ini digunakan sebagai jalur pintu keluar masuknya informasi atau data. Pada saat *port* dibuka, muncul pesan pada

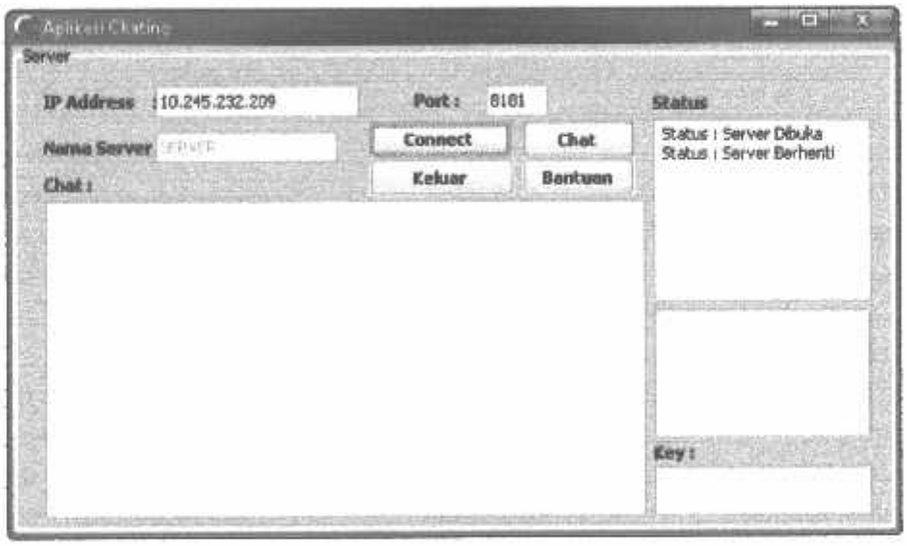
jendela status yang menyatakan bahwa *port* pada komputer pertama dibuka, lalu menunggu komputer – komputer lain yang ingin berkoneksi seperti pada gambar dibawah ini.



Gambar 4.3 Status Ketika *Connect* Ditekan

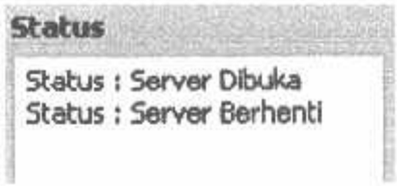
2. **Server Stop** 

Tombol ini digunakan untuk menutup *port* sekaligus memutuskan hubungan atau koneksi ke komputer lain. Berikut tampilan *form* saat tombol *Server Stop* ditekan seperti pada gambar 4.4.



Gambar 4.4 Tampilan *Form* Saat Tombol *Server Stop* Ditekan

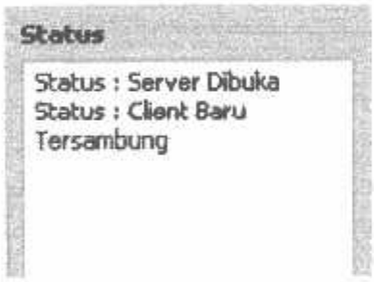
Ketika *Server Stop* ditekan, maka hubungan antara dua komputer akan berhenti sekaligus menutup *port* 8181 seperti yang ditampilkan pada gambar 4.5.



Gambar 4.5 Status Ketika *Server* Berhenti

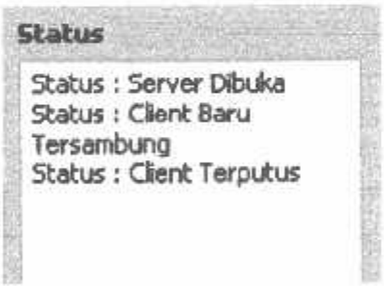
Berikut beberapa kondisi atau keterangan pada jendela status saat terjadinya hubungan (*Connect*) atau tidak terhubung (*disconnect*) ke komputer penerima :

- Kondisi ketika terjadi saat komputer penerima sudah terhubung ke komputer pengirim, maka jendela status akan menampilkan pesan seperti gambar 4.6



Gambar 4.6 Kondisi *Client* Terhubung Ke *Server*

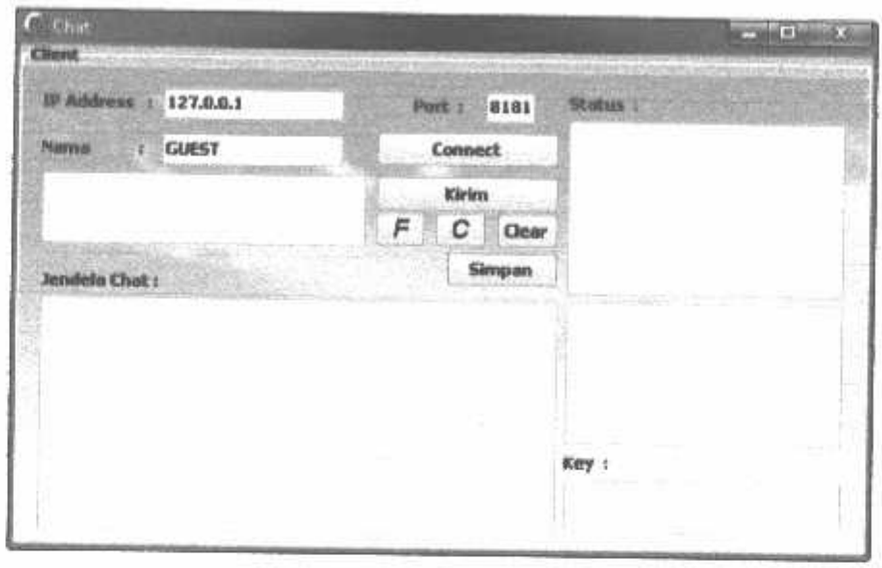
- Kondisi terjadi saat komputer penerima memutuskan hubungan (*disconnect*) ke *Server*, maka jendela status akan menampilkan pesan seperti gambar 4.7



Gambar 4.7 Kondisi *Client* Memutuskan Hubungan (*Disconnect*) Ke *Server*

3. **Chat** Chat

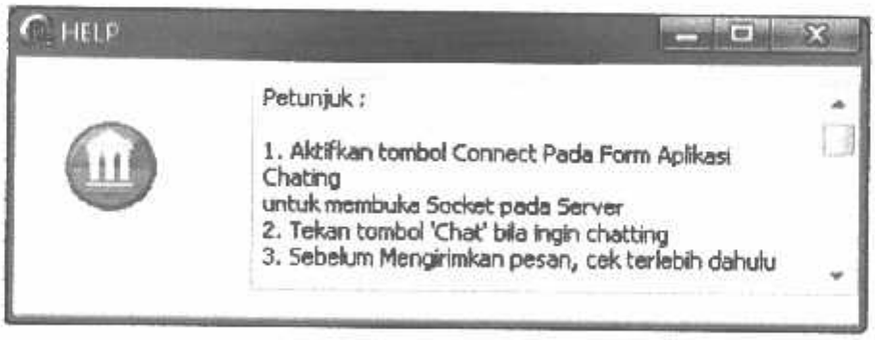
Tombol *Chat* digunakan sebagai sarana untuk mengirim atau membalas pesan ke komputer lain, dengan kata lain sebagai media untuk berkomunikasi antar dua entitas (pengirim dan penerima). Berikut tampilan form ketika tombol *Chat* ditekan seperti gambar 4.8



Gambar 4.8 Tampilan *Form* Ketika Tombol *Chat* Ditekan

4. **Bantuan** Bantuan

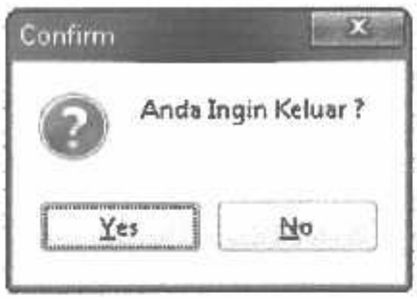
Tombol ini sebagai pemberitahuan mengenai petunjuk penggunaan aplikasi ini. Ketika tombol ditekan, maka akan mucul *form* Bantuan seperti gambar 4.9



Gambar 4.9 Tampilan *Form* Bantuan

5. **Keluar** 

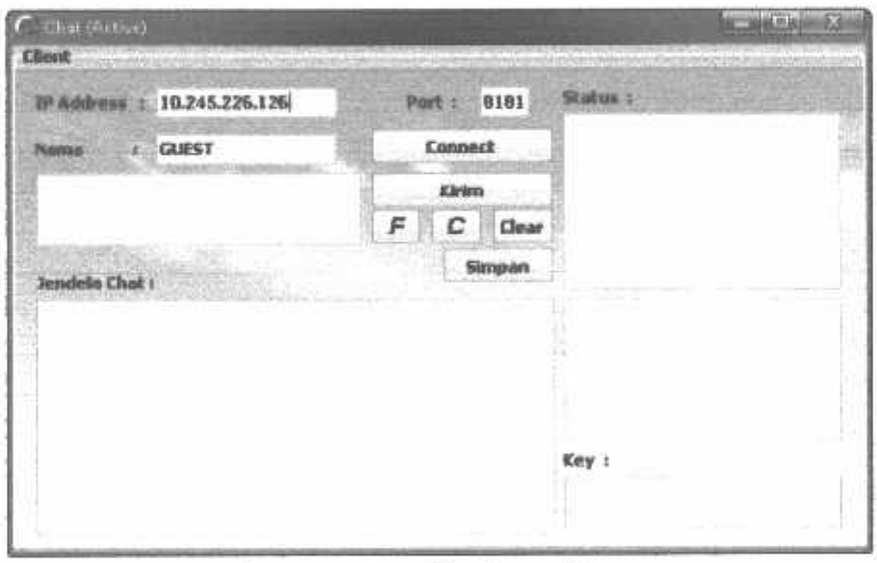
Tombol ini digunakan untuk mengakhiri program ini (*exit*) seperti gambar 4.10



Gambar 4.10 Tampilan Pemberitahuan Keluar Dari Program

4.2.2 Form Chat

Form ini digunakan untuk mengirimkan pesan ataupun membalas pesan kepada komputer yang dituju. Berikut tampilan *Form Chat* seperti gambar 4.11

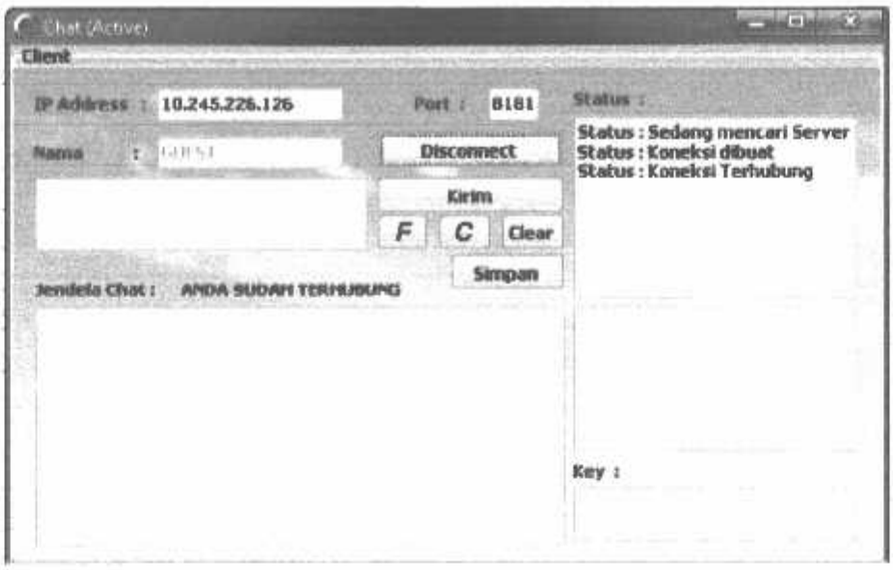


Gambar 4.11 Tampilan Form Chat

Adapun fungsi masing – masing tombol (*button*) pada *Form Chat* sebagai berikut :

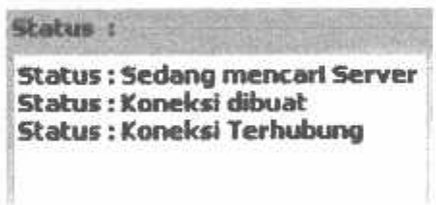
1. **Connect** 

Tombol ini berfungsi untuk menghubungkan koneksi ke komputer yang dituju seperti yang ditampilkan pada gambar 4.12



Gambar 4.12 Tampilan *Form Chatting* Saat *Connect*

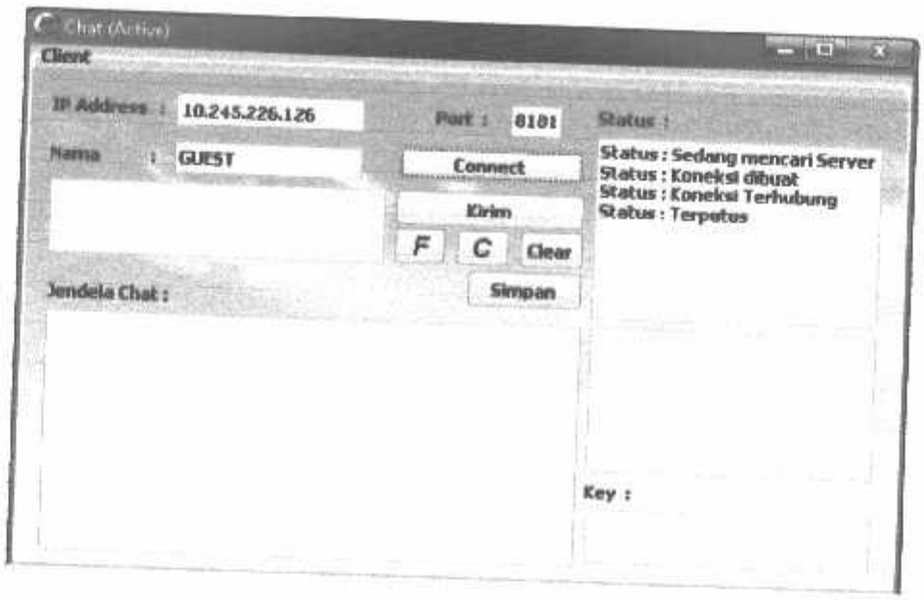
Saat tombol *Connect* ditekan, maka komputer akan mencari alamat komputer yang akan dituju dengan IP (xxx.xxx.xxx.xxx), ketika sudah menemukan komputer yang diinginkan lalu komputer akan memberitahu lewat jendela statusnya bahwa koneksi sudah terhubung dan bisa mulai untuk berkomunikasi. Berikut tampilannya seperti gambar 4.13



Gambar 4.13 Jendela Status *Connect*

2. Disconnect [Disconnect]

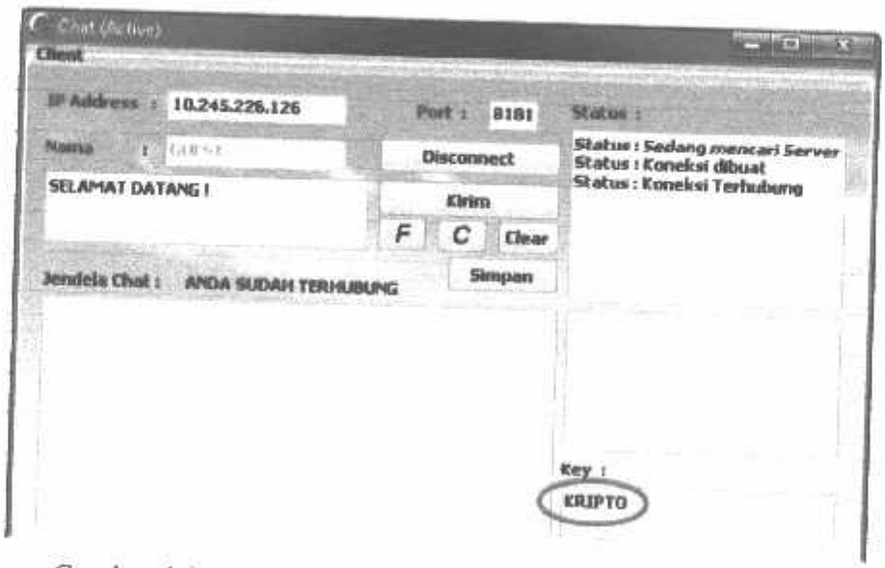
Tombol ini berguna untuk memutuskan hubungan atau koneksi kepada komputer yang dituju seperti yang ditunjukkan pada gambar 4.14



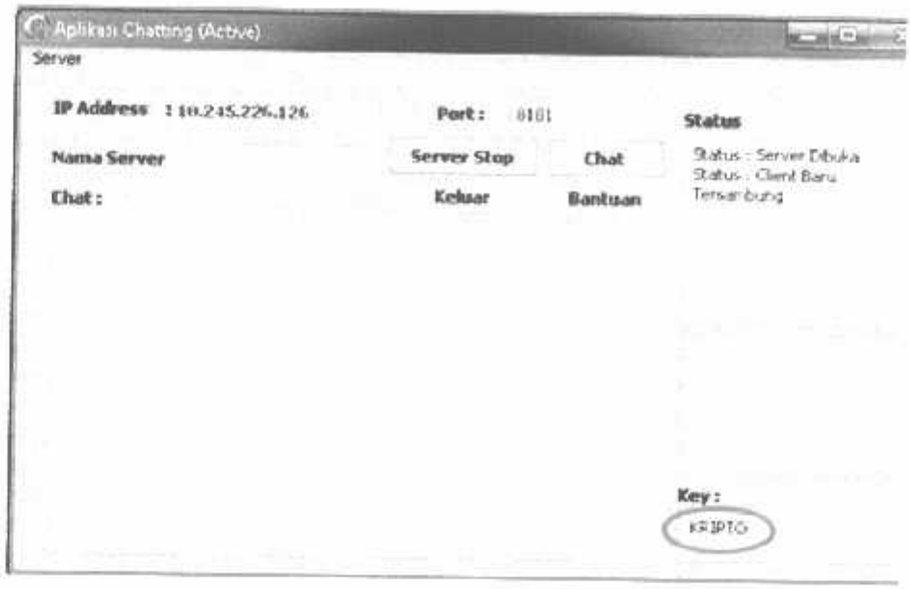
Gambar 4.14 Tampilan *Form Chat* Saat *Disconnect*

3. **Kirim** 

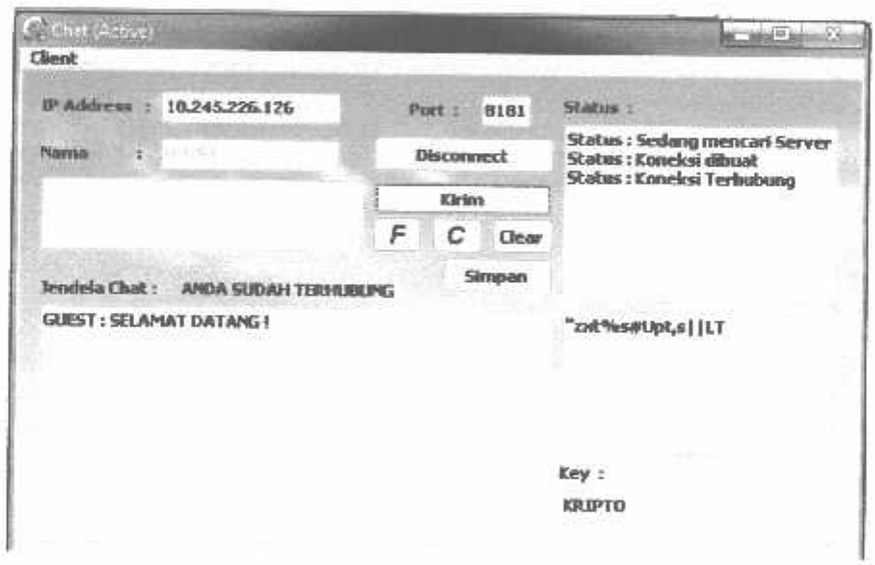
Tombol ini berfungsi untuk mengirimkan pesan yang ketik, yang nantinya pada komputer yang dituju akan menerima pesan tersebut. Ada satu hal yang perlu diperhatikan bila ingin mengirimkan pesan yaitu pengisian *key* pada pengirim dan penerima haruslah sama, jika tidak maka keluaran (*output*) tidak akan sesuai dengan pesan yang diharapkan seperti yang ditunjukkan pada gambar 4.15, gambar 4.16, gambar 4.17 dan gambar 4.18



Gambar 4.15 Tampilan *Form Chat* Saat Pada Pengirim



Gambar 4.16 Tampilan Form Aplikasi *Chatting* Pada Penerima

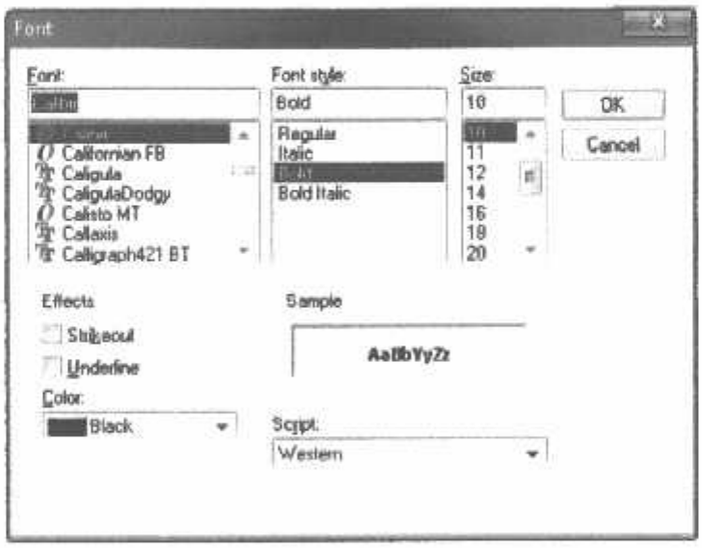


Gambar 4.17 Tampilan *Form Chat* Saat Mengirim Pesan

Gambar 4.18 Tampilan *Form* Aplikasi *Chatting* Saat Menerima Pesan

4. Huruf *F*

Berfungsi untuk mengubah jenis, ukuran dan warna huruf (*font*). Adapun tampilan pengaturan huruf seperti pada gambar 4.19



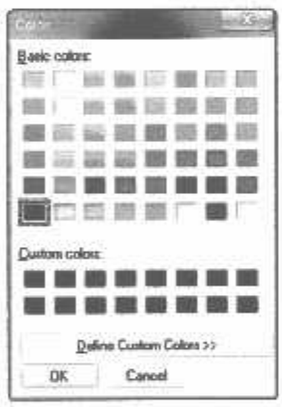
Gambar 4.19 Tampilan Pengaturan Huruf (*Font*)

Sedangkan Gambar 4.20 menunjukkan tampilan *Form Chat* saat ditekan

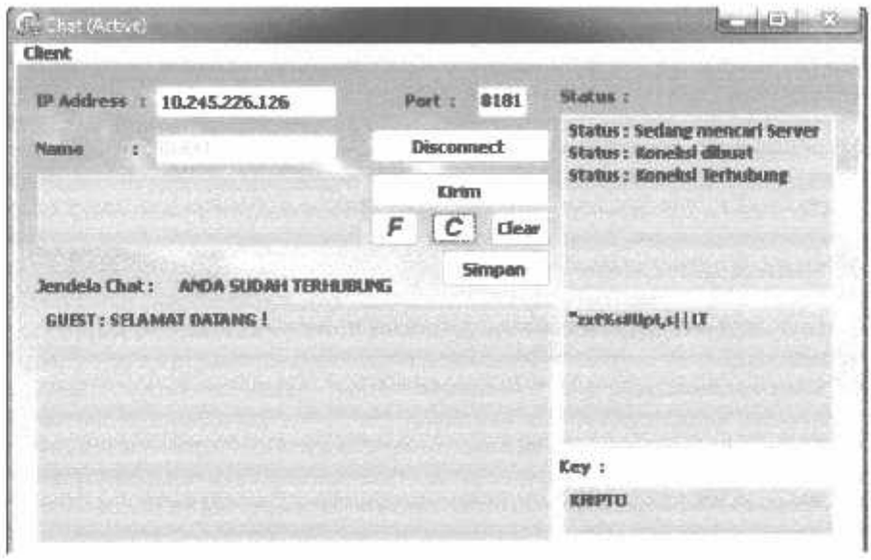
Gambar 4.20 Tampilan *Form Chat* Setelah Huruf Diganti

5. **Warna** `C`

Berfungsi untuk mengubah jenis warna pada masing – masing jendela. Gambar 4.21 menampilkan pengaturan warna dan gambar 4.22 menampilan *Form Chat* setelah warna diganti



Gambar 4.21 Tampilan Pengaturan Warna



Gambar 4.22 Tampilan *Form Chat* Setelah Warna Diganti

6. Hapus Clear

Berfungsi untuk menghapus pesan pada jendela dan pesan rahasia(cipherteks) pada jendela *Cipher* seperti pada gambar 4.23 .

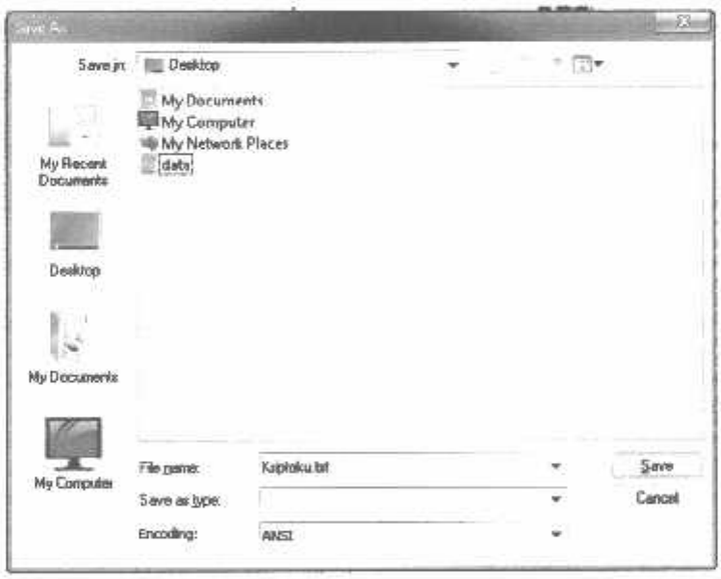
Gambar 4.23 Tampilan Ketika Jendela *Chat*
dan Jendela *Cipher* Dihapus

7. **Simpan** Simpan

Tombol ini berfungsi untuk menyimpan hasil dari proses enkripsi yang berupa pesan rahasia (cipherteks). Pesan rahasia akan disimpan ke dalam suatu direktori. Gambar 4.24 menampilkan pesan rahasia yang sudah diproses dan akan disimpan

Gambar 4.24 Pesan Rahasia (Cipherteks)

Gambar 4.25 menampilkan letak penyimpanan *File* yang berisi pesan rahasia

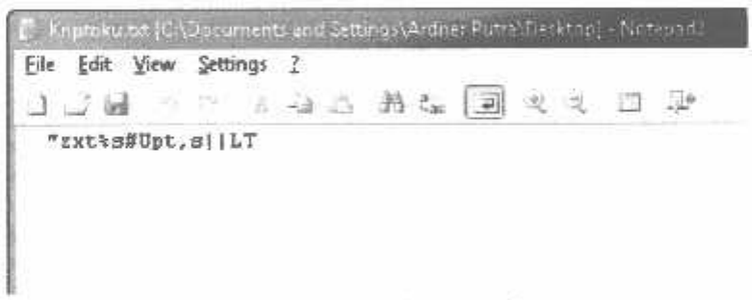


Gambar 4.25 Mencari Letak Penyimpanan *File*

Gambar 4.26 menampilkan hasil penyimpanan *File* yang berupa pesan rahasia tadi dengan format **.txt*

Gambar 4.26 Hasil Penyimpanan *File* berupa (*.txt)

Gambar 4.27 menampilkan isi dari *File* pesan rahasia



Gambar 4.27 Isi dari *File Kriptoku.txt*

4.2.3 Coding

Secara garis besar penulisan program (*coding*) pada aplikasi *chatting* ini terbagi menjadi dua, yakni :

- Enkripsi/Dekripsi.
- *Chat*.

1. Coding Enkripsi/Dekripsi

Coding Enkripsi/Dekripsi terbagi menjadi dua , yaitu :

- **Enkripsi**

  Digunakan untuk mengacak suatu pesan menjadi pesan yang sulit dimengerti atau biasa juga disebut pesan rahasia (cipherteks). Berikut penggalan *coding* untuk Enkripsi :

```
function EkCeVi(Text,Key : string):string;
var
  pos1,pos2 : integer;
  c,l: byte;
begin
    result:= Text;
     if Length(result) > 0 then
     begin
      pos2 := 1;
       for pos1 := 1 to Length(result) do
       begin
           c := ord(result[pos1]) - 32;
           l := ord(Key[pos2]) - 32;
           result[pos1] := chr(32+((c+l+3) mod 95) );
           if pos2 = Length(Key) then
              pos2:=1
           else
              inc(pos2);
       end;
     end;
end;
```

- **Dekripsi**

  Digunakan untuk mengembalikan pesan rahasia (cipherteks) menjadi pesan yang bisa dimengerti. Berikut penggalan *coding* untuk Enkripsi :

```
function DkCeVi(Chiper,Key : string): string;
var
  pos1,pos2 : integer;
 c,l : byte;
begin
  result:= Chiper;
  if Length(result) > 0 then
  begin
   pos2 := 1;
    for pos1 := 1 to Length(result) do
     begin
      c := ord (result[pos1]) - 32;
      l := ord (Key[pos2]) - 32;
      result[pos1] := chr ((((c+95)-l-3)mod 95)+32);
     if pos2 = Length(Key) then
           pos2:=1
       else
           inc(pos2);
     end;
  end;
end;
```

2. **Coding Chat**

Meliputi dua bagian , yaitu

- **Send (Pengiriman)**

Berikut penggalan *coding* untuk mengirimkan pesan

```
procedure TfrmClient.btKirimClick(Sender: TObject);
var
 Pesan : TChatPesan;
 Chiper : string;
begin
 try
 Chiper := EkCeVi(MemKirim.Text,MemKey2.Text);
 Pesan.Nickname := ednamaclient.Text;
 Pesan.Text := Chiper;
 MemPesanClient.Lines.Add(ednamaclient.Text + ' : ' +
                  MemKirim.Text);
  try
  ClientSocket.Socket.SendBuf(Pesan, SizeOf(Pesan));
  except
  end;
  if (MemKirim.Text<>'') then
   begin
     MemChiper.Lines.Add(Pesan.Text);
   end;
  if ednamaclient.Enabled = true then
  else
   MemKirim.Text := '';
 except
 showmessage('Periksa Key , apakah sudah dimasukkan ?');
 end;
end;
```

- **Receive (Penerimaan)**

Berikut penggalan *coding* untuk menerima pesan

```
procedure TFrmChating.ServerSocketClientRead(Sender: TObject;
 Socket: TCustomWinSocket);
var
 Pesan : TChatPesan;
 plain: string;
begin
  try
  Socket.ReceiveBuf(Pesan, SizeOf(Pesan));
  plain := DkCeVi(Pesan.Text,MemKey1.Text);
  MemPesan.Lines.Add('Dari ' + Pesan.Nickname + ':' + plain);
   except
  showmessage('Key Pada Server Belum Terisi !!');
  end;
end;
end.
```

4.3 Pengujian Sistem

4.3.1 Pengujian Terhadap Algoritma Enkripsi Atau Dekripsi

Pengujian dilakukan untuk mengetahui kebenaran hasil algoritma enkripsi maupun dekripsi

1. Pengujian Enkripsi

Pengujian ini dilakukan dengan membandingkan algoritma Caesar dan Vignere dengan algoritma Ce-Vi dengan mengambil satu buah karakter untuk dijadikan contoh (*sample*).

Misal :

Plainteks : SELAMAT DATANG !

Key : KRIPTO

Perhitungan menurut algoritma Caesar dan Vignere, dengan menggunakan persamaan (3-1) dan persamaan (3-3)

Algoritma Caesar :

C = 32+(a+3)mod 95

C = 32+(51+3) mod 95

C = 32 + 54 mod 95

C = **86**

Algoritma Vignere :

C = 32+(a+l)mod 95

C = 32+(54+43) mod 95

C = 32 + 97 mod 95

C = **34** atau sama dengan karakter (")

Perhitungan menurut algoritma Ce-Vi, berdasarkan persamaan (3-10) maka :

C = 32+((32+(a+3)mod 95)+l)mod 95

C = 32+((32(51+3)mod 95)+1) mod 95

C = 32 + (86+l) mod 95

C = 32 + (54+43) mod 95

C = 32 + (97) mod 95

C = **34** atau sama dengan karakter (")

2. Pengujian Dekripsi

Pengujian ini dilakukan dengan mengambil satu buah karakter untuk dijadikan contoh (*sample*).

Contoh ciphertcks diambil dari hasil enkripsi keseluruhan dari plainteks.

Misal :

Cipher : "zxt%s#Upt,s||LT

Key : KRIPTO

Perhitungan menurut algoritma Caesar dan Vignere, dengan menggunakan persamaan (3-2) dan persamaan (3-4)

Algoritma Vignere :

P = (b – 1 + 95) mod 95 + 32

P = (2-43+95) mod 95 + 32

P = 54 mod 95 + 32

P = 54 + 32

P = 86

Algoritma Caesar :

P = (b – 3 + 95) mod 95 + 32

P = (54-3+95) mod 95 + 32

P = 146 mod 95 + 32

P = **83** atau sama dengan karakter (**S**)

Perhitungan menurut algoritma Ce-Vi, berdasarkan persamaan (3-12) maka :

P = (((b – I + 95) mod 95 + 32) – 3 + 95) mod 95 + 32

P = (((2-43+95) mod 95) + 32) – 3 + 95) mod 95 + 32

P = ((54 mod 95) + 32) – 3 + 95) mod 95 + 32

P = (86 – 3 + 95) mod 95 + 32

P = (54-3+95) mod 95 + 32

P = 146 mod 95 + 32

P = **83** atau sama dengan karakter (**S**)

4.3.2 Pengujian Terhadap Waktu

Pengujian ini dilakukan untuk membandingkan waktu pengolahan plainteks menjadi cipherteks dan membandingkan waktu yang dibutuhkan untuk mengirimkan pesan rahasia tersebut ke pihak lain serta mengetahui waktu yang dibutuhkan agar hubungan atau koneksi dua entitas dapat terjadi. Dalam hal ini penguji menggunakan dua buah modem *usb* yang terpasang pada masing – masing komputer. Berikut perbandingan waktu cipherteks seperti pada tabel 4.2

Tabel 4.2 Perbandingan Waktu Cipherteks

NO	JUMLAH KARAKTER PLAINTEKS	JUMLAH KARAKTER KEY	WAKTU YANG DIBUTUHKAN
1	50 karakter	10 karakter	± 1 detik
2	100 karakter	15 karakter	± 1 detik
3	1000 karakter	20 karakter	± 1 detik

Dilihat dari tabel diatas, maka rata – rata waktu yang dibutuhkan untuk mengolah pesan asli (plainteks) menjadi pesan rahasia (cipherteks) kurang lebih hanya 1 detik saja. Sedangkan waktu yang digunakan agar komputer pengirim dan komputer penerima bisa terhubung kurang lebih 3 detik .Untuk mengirimkan pesan ke komputer lain dibutuhkan kurang lebih 3 detik saja. Berikut perbandingan waktu pengiriman pesan seperti pada tabel 4.3

Tabel. 4.3 Perbandingan Waktu Pengiriman Pesan

NO	JUMLAH KARAKTER CIPHER	WAKTU YANG DIBUTUHKAN
1	50 karakter	± 3 detik
2	100 karakter	± 3 detik
3	1000 karakter	± 3 detik

4.4 Pemeliharaan / Maintenance

Apabila program ini dijalankan perlu diatur jadwal untuk melakukan pemeliharaan *Hardware* dan *Software*. Khususnya bidang *Software*, pemeliharaan program ini dilakukan secara rutin yang meliputi melakukan perubahan *key* secara rutin (minimal satu bulan sekali). Ini dilakukan agar mencegah orang lain tidak bisa mengetahui secara pasti *key* yang digunakan pada program tersebut. Pengiriman *key* diusahakan hanya diketahui oleh pihak – pihak yang berkepentingan saja dan tidak menyebarkan kepada orang yang tidak berhak. Pengembangan algoritma program perlu dilakukan yaitu dengan menambah atau mengembangkan alghoritma yang sudah ada agar menjadi program yang lebih kompleks dan handal daripada sebelumnya.

BAB V
PENUTUP

5.1 Kesimpulan

Dari hasil penelitian dan pengujian yang dilakukan dapat dibuat beberapa kesimpulan sebagai berikut :

1. Menggabungkan serta mengembangkan dua algoritma kriptografi atau lebih merupakan salah satu metode yang cukup berguna untuk mengamankan data atau informasi bila dibandingkan menggunakan satu buah algoritma kriptografi saja dikarenakan algoritma Ce-Vi ini lebih efisien.
2. Penggabungan dua buah kriptografi akan membuat cipherteks semakin susah untuk dipecahkan karena seorang kriptanalis tidak akan mengetahui algoritma apa yang digunakan.
3. Lama atau tidaknya hubungan komunikasi (*Connect*) antara pengirim dan penerima tergantung dari sinyal atau jaringan itu sendiri.
4. Waktu yang dibutuhkan untuk melakukan proses enkripsi atau dekripsi serta proses pengiriman maupun penerimaan pada program ini cukup cepat, sehingga untuk membangun program ini tidak memerlukan spesifikasi peralatan komputer yang lengkap dengan biaya yang mahal.

5.2 Saran

Berikut adalah saran – saran untuk pengembangan lebih lanjut terhadap penelitian ini:

1. Untuk program pengembangan selanjutnya perlu dibuat tidak hanya menggunakan algoritma simetri saja, bisa juga menggunakan algoritma asimetri atau perpaduan algoritma simetri maupun algoritma asimetri.
2. Kelemahan menggunakan algoritma simetri ini adalah pada penyebaran kuncinya yang rawan, sehingga kedepannya dibutuhkan suatu metode khusus untuk melindungi kerahasiaan kunci.
3. Untuk kedepannya agar sistem yang dibuat tidak hanya penggunaan teks saja, tetapi juga dapat digunakan untuk *image*,*audio*, *video*, dan lain – lain.

DAFTAR PUSTAKA

Anonim. *Bab 1 Pengantar Kriptografi.* URL:http://www.informatika.org/.../Kriptografi/Bab-1_Pengantar%20Kriptografi.pdf.

Anonim.*Delphi 2010 Architect.*2008. URL:http://shop.embarcadero.com/.

Anonim.*Tiga Kebohongan Facebook Tentang Privasi.* 2010. http://unik-supopersicum.com. tiga- kebohongan-facebook-tentang-privasi.

Betha Sidik, Ir., 2005.*MySQL Untuk Pengguna, Administrator, dan Pengembang Aplikasi Web.*Informatika.

Pakereng, Ineke, M.A, & Teguh Wahyono. 2004. *Sistem Basis Data*, Graha Ilmu.

Whitten, Jeffery L. & Bentley, Lonnie D. 2004. *System Analysis and Design Methods 6th Ed*, The McGraw-Hill Education. New York.

Yuli Andri, M. 2009. *Implementasi Algoritma Kriptografi DES, RSA, dan Algoritma Kompresi LZW pada berkas digital.*

Zakimath.2008. *Klub Sandi (GMU Cryptology Club).* URL:http://sandi.math.web.id.

LAMPIRAN

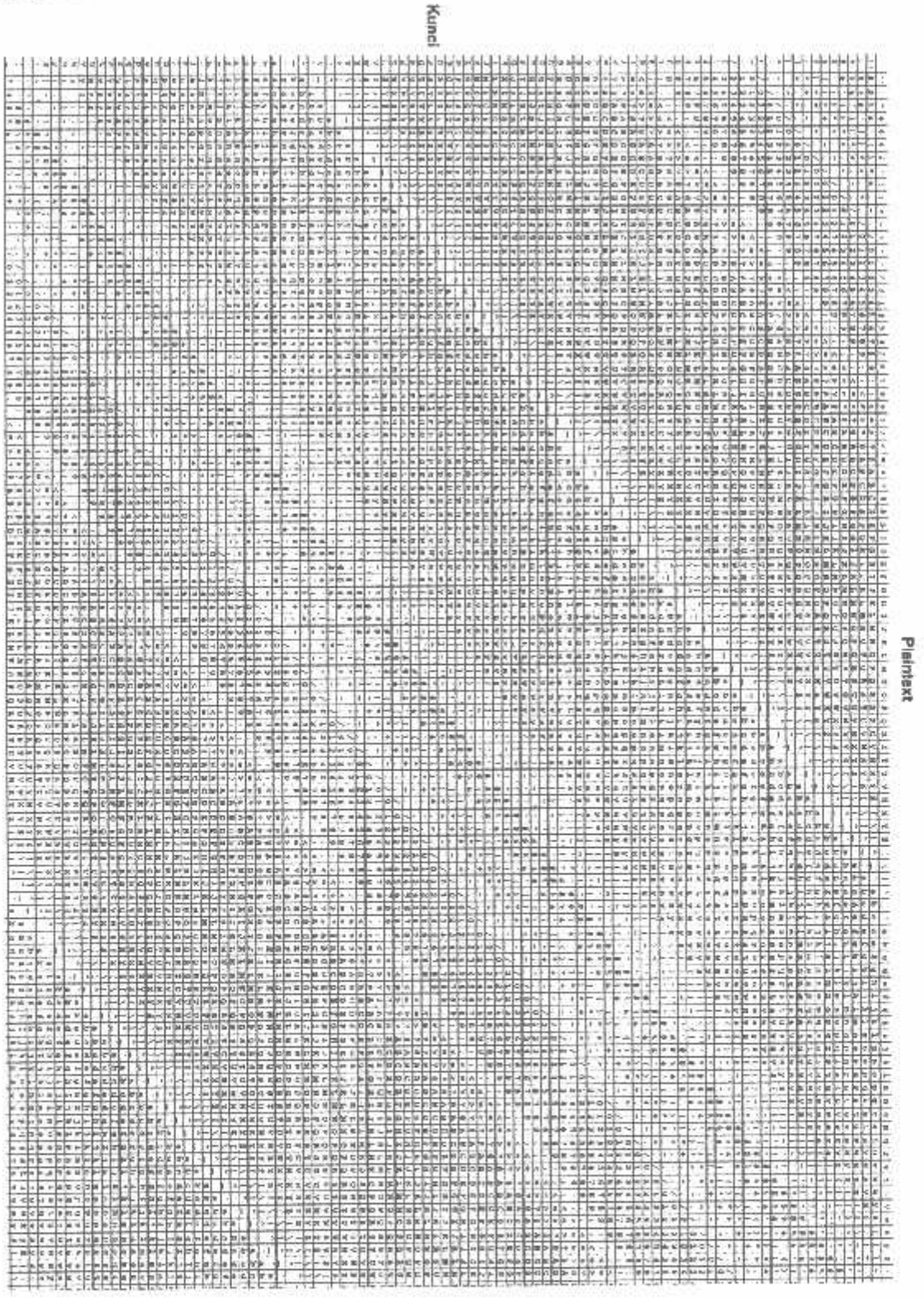
LAMPIRAN

Lampiran 1. Tabel ANSI

The Ansi character set

Char	Code	Description	Char	Code	Description
' '	32	Space	V	86	Latin capital letter V
!	33	Exclamation mark	W	87	Latin capital letter W
"	34	Quotation mark	X	88	Latin capital letter X
#	35	Numbesign	Y	89	Latin capital letter Y
$	36	Dollar sign	Z	90	Latin capital letter Z
%	37	Percent sign	[	91	Left square bracket
&	38	Ampersand	\	92	Reverse solidus
'	39	Apostrophe	]	93	Right square bracket
(	40	Left parenthesis	^	94	Circumflex accent
)	41	Right parenthesis	_	95	Low line
*	42	Asterisk	`	96	Grave accent
+	43	Plus sign	a	97	Latin small letter a
,	44	Comma	b	98	Latin small letter b
-	45	Hypherminus	c	99	Latin small letter c
.	46	Full stop	d	100	Latin small letter d
/	47	Solidus	e	101	Latin small letter e
0	48	Digit zero	f	102	Latin small letter f
1	49	Digit one	g	103	Latin small letter g
2	50	Digit two	h	104	Latin small letter h
3	51	Digit three	i	105	Latin small letter i
4	52	Digit four	j	106	Latin small letter j
5	53	Digit five	k	107	Latin small letter k
6	54	Digit six	l	108	Latin small letter l
7	55	Digit seven	m	109	Latin small letter m
8	56	Digit eight	n	110	Latin small letter n
9	57	Digit nine	o	111	Latin small letter o
:	58	Colon	p	112	Latin small letter p
;	59	Semicolon	q	113	Latin small letter q
<	60	Less-than sign	r	114	Latin small letter r
=	61	Equals sign	s	115	Latin small letter s
>	62	Greater-than sign	t	116	Latin small letter t
?	63	Question mark	u	117	Latin small letter u
@	64	Commerciabt	v	118	Latin small letter v
A	65	Latin capital letter A	w	119	Latin small letter w
B	66	Latin capital letter B	x	120	Latin small letter x
C	67	Latin capital letter C	y	121	Latin small letter y
D	68	Latin capital letter D	z	122	Latin small letter z
E	69	Latin capital letter E	{	123	left curly bracket
F	70	Latin capital letter F	|	124	Vertical line
G	71	Latin capital letter G	}	125	Right curly bracket
H	72	Latin capital letter H	~	126	Tilde
I	73	Latin capital letter I			
J	74	Latin capital letter J			
K	75	Latin capital letter K			
L	76	Latin capital letter L			
M	77	Latin capital letter M			
N	78	Latin capital letter N			
O	79	Latin capital letter O			
P	80	Latin capital letter P			
Q	81	Latin capital letter Q			
R	82	Latin capital letter R			
S	83	Latin capital letter S			
T	84	Latin capital letter T			
U	85	Latin capital letter U			

Lampiran 2. Tabel Ce-Vi 95 Karakter



Lampiran 3 Source Code

1. Chat.pas

```
unit Chat;

interface

uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, ScktComp, IdBaseComponent, IdComponent, IdIPWatch, jpeg,
 ExtCtrls;

 type
 TChatPesan = Record
 Nickname: string[255];
 Text:    string[255];
 end;

type
 TFrmChating = class(TForm)
  Server: TGroupBox;
  lbIPadrs: TLabel;
  edIP: TEdit;
  lbPort: TLabel;
  edPort: TEdit;
  lbNamaServer: TLabel;
  edNameServer: TEdit;
  btStartStop: TButton;
  MemPesan: TMemo;
  MemStatus: TMemo;
  lbChat: TLabel;
  lbStatus: TLabel;
  ServerSocket: TServerSocket;
  IdIPWatch: TIdIPWatch;
  btChat: TButton;
  lbKey1: TLabel;
  MemKey1: TMemo;
  btExit: TButton;
  bthelp: TButton;
  MemChiper2: TMemo;
  procedure btStartStopClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure ServerSocketClientConnect(Sender: TObject;
   Socket: TCustomWinSocket);
```

```
  procedure ServerSocketClientDisconnect(Sender: TObject;
    Socket: TCustomWinSocket);
  procedure ServerSocketClientRead(Sender: TObject; Socket:
TCustomWinSocket);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure btChatClick(Sender: TObject);
  procedure btExitClick(Sender: TObject);
  procedure bthelpClick(Sender: TObject);
  procedure FormActivate(Sender: TObject);
  procedure FormDeactivate(Sender: TObject);

 private

 public

 end;

var
 FrmChating: TFrmChating;

implementation

uses UnitClient, UnitHelp;

{$R *.dfm}
function DkCeVi(Chiper,Key : string): string;
var
  pos1,pos2 : integer;
  c,l : byte;
begin
  result:= Chiper;
  if Length(result) > 0 then
  begin
    pos2 := 1;
     for pos1 := 1 to Length(result) do
      begin
        c := ord (result[pos1]) - 32;
        l := ord (Key[pos2]) - 32;
        result[pos1] := chr ((((c+95)-l-3)mod 95)+32);
        if pos2 = Length(Key) then
              pos2:=1
         else
              inc(pos2);
      end;
```

```
 end;
end;
procedure TFrmChating.btExitClick(Sender: TObject);
var
 res : word;
begin
  serversocket.Close;
  res := MessageDlg ('Anda Ingin Keluar ?', mtConfirmation, [mbYes,mbNo], 0);
  if res = mrYes then
  begin
   MessageDlg ('Terima Kasih !!!',mtInformation,[mbOK],0);
   close;
  end
  else
   frmchating.Show;
end;

procedure TFrmChating.bthelpClick(Sender: TObject);
begin
 frmbantuan.show;
end;

procedure TFrmChating.btStartStopClick(Sender: TObject);
begin
 if serversocket.Active then
  begin
   serversocket.Close;
   BtStartStop.Caption := 'Connect';
   MemStatus.Lines.Add('Status : Server Berhenti');
  end
 else
  begin
   serversocket.Open;
   serversocket.Port := strtoint(edPort.Text);
   BtStartStop.Caption := 'Server Stop';
   MemStatus.Lines.Add('Status : Server Dibuka');
   edIP.Text := IdIPWatch.LocalIP;
  end
end;

procedure TFrmChating.btChatClick(Sender: TObject);
begin
  frmClient.show;
end;
```

```
procedure TFrmChating.FormActivate(Sender: TObject);
begin
 FrmChating.Caption := 'Aplikasi Chatting (Active)';
end;

procedure TFrmChating.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  ServerSocket.Close;
end;

procedure TFrmChating.FormCreate(Sender: TObject);
begin
  MemStatus.ReadOnly := true;
  MemPesan.ReadOnly := true;
  edIP.ReadOnly := true;
end;

procedure TFrmChating.FormDeactivate(Sender: TObject);
begin
FrmChating.Caption := 'Aplikasi Chatting';
end;

procedure TFrmChating.ServerSocketClientConnect(Sender: TObject;
 Socket: TCustomWinSocket);
begin
  MemStatus.Lines.Add('Status : Client Baru Tersambung');
end;

procedure TFrmChating.ServerSocketClientDisconnect(Sender: TObject;
 Socket: TCustomWinSocket);
begin
   MemStatus.Lines.Add('Status : Client Terputus');
end;

procedure TFrmChating.ServerSocketClientRead(Sender: TObject;
 Socket: TCustomWinSocket);
var
 Pesan : TChatPesan;
 plain: string;
begin
  try
  Socket.ReceiveBuf(Pesan, SizeOf(Pesan));
  MemChiper2.Lines.Add(Pesan.Text);
  plain := DkCeVi(Pesan.Text,MemKey1.Text);
```

```
MemPesan.Lines.Add('Dari ' + Pesan.Nickname + ':' + plain);
    except
   showmessage('Key Pada Server Belum Terisi !!');
   end;
end;
end.
```

2. UnitCaesar.pas

```
unit UnitCaesar;

interface

uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, Menus;

type
 TFormCaesar = class(TForm)
  btEnkrip: TButton;
  btDekrip: TButton;
  mPlain: TMemo;
  mChipher: TMemo;
  mPlain2: TMemo;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  procedure btEnkripClick(Sender: TObject);
  procedure btDekripClick(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure mPlainKeyPress(Sender: TObject; var Key: Char);

 private

 public

 end;
var
 FormCaesar: TFormCaesar;
implementation

{$R *.dfm}
```

```
function EnCaesar(Plain : string): string;
var
 pos : integer;
 k : byte;
begin
 result := Plain;
 for pos := 1 to length(result) do
     begin
      k := ord(result[pos])-32;
      result[pos] := chr(32+(k+3)mod 95);
     end;
end;

function DkCaesar(Chiper : string): string;
var
 pos : integer;
 k : byte;
begin
 result := Chiper;
 for pos := 1 to length(result) do
     begin
      k := ord(result[pos])-32;
      result[pos] := chr(((k+95-3)mod 95)+32);
     end;
end;

procedure TFormCaesar.btDekripClick(Sender: TObject);
begin
 if mChipher.Text = '' then
 begin
  MessageDlg('Pesan Sandi Belum ADA, Enkripsi Dahulu !!',mtError,[mbOK],0);
 end
 else
 mPlain2.Text := DkCaesar(mChipher.Text);
end;

procedure TFormCaesar.btEnkripClick(Sender: TObject);
begin
 btDekrip.Enabled := true;
 if mPlain.Text = '' then
 begin
  MessageDlg ('Pesan Belum Terisi !!!',mtError,[mbOK],0);
  FormCaesar.Show;
 end
```

```
  else
    mChipher.Text := EnCaesar(mPlain.Text);
end;

procedure TFormCaesar.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  mPlain.Lines.Clear;
  mPlain2.Lines.Clear;
  mChipher.Lines.Clear;
end;

procedure TFormCaesar.mPlainKeyPress(Sender: TObject; var Key: Char);
begin
  btDekrip.Enabled := false;
end;

end.
```

3 UnitClient.pas

```
unit UnitClient;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ScktComp, ExtCtrls, jpeg, ExtDlgs;

type
  TChatPesan = Record
  Nickname: string[255];
  Text:    string[255];

end;
type
  TfrmClient = class(TForm)
    GroupBox1: TGroupBox;
    lbIPClient: TLabel;
    edIPClient: TEdit;
    lbNamaClient: TLabel;
    edNamaClient: TEdit;
    lbPortClient: TLabel;
```

```
    edPortClient: TEdit;
    btConnect: TButton;
    MemKirim: TMemo;
    btKirim: TButton;
    MemPesanClient: TMemo;
    MemStatusClient: TMemo;
    MemChiper: TMemo;
    lbKey2: TLabel;
    MemKey2: TMemo;
    lbStatusClient: TLabel;
    ClientSocket: TClientSocket;
    lblWindow: TLabel;
    lblwelcome: TLabel;
    FontDialog: TFontDialog;
    btFont: TButton;
    ColorDialog: TColorDialog;
    btClear: TButton;
    btColor: TButton;
    Image1: TImage;
    SaveTextFileDialog: TSaveTextFileDialog;
    btSimpan: TButton;
    procedure btConnectClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure ClientSocketLookup(Sender: TObject; Socket: TCustomWinSocket);
    procedure ClientSocketConnecting(Sender: TObject; Socket:
TCustomWinSocket);
    procedure ClientSocketConnect(Sender: TObject; Socket: TCustomWinSocket);
    procedure ClientSocketError(Sender: TObject; Socket: TCustomWinSocket;
      ErrorEvent: TErrorEvent; var ErrorCode: Integer);
    procedure ClientSocketDisconnect(Sender: TObject; Socket:
TCustomWinSocket);
    procedure btKirimClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btFontClick(Sender: TObject);
    procedure btColorClick(Sender: TObject);
    procedure btClearClick(Sender: TObject);
    procedure btSimpanClick(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormDeactivate(Sender: TObject);
    private
    Connect : Boolean;

  public
```

```
end;
var
 frmClient: TfrmClient;

implementation

{$R *.dfm}
function EkCeVi(Text,Key : string):string;
var
  pos1,pos2 : integer;
  c,l: byte;
begin
    result:= Text;
     if Length(result) > 0 then
     begin
      pos2 := 1;
       for pos1 := 1 to Length(result) do
       begin
           c := ord(result[pos1]) - 32;
           l := ord(Key[pos2]) - 32;
           result[pos1] := chr(32+((c+l+3) mod 95) );
           if pos2 = Length(Key) then
              pos2:=1
           else
              inc(pos2);
       end;
     end;
end;

procedure TfrmClient.btClearClick(Sender: TObject);
begin
MemPesanClient.Text := '';
MemChiper.Text := '';
end;

procedure TfrmClient.btColorClick(Sender: TObject);
begin
   Colordialog.Execute;
   MemKirim.Color := colordialog.Color;
   MemPesanClient.Color := colordialog.Color;
   MemStatusClient.Color := colordialog.Color;
   MemChiper.Color := colordialog.Color;
   MemKey2.Color := colordialog.Color;
end;
```

```
procedure TfrmClient.btConnectClick(Sender: TObject);
begin
 if Connect then
    begin
     Connect := false;
     ClientSocket.Close;
     btConnect.Caption := 'Connect';
     edNamaClient.Enabled := true;
    end
 else
    begin
     Connect := true;
     edNamaClient.Enabled := false;
     ClientSocket.Host := edIPClient.Text;
     ClientSocket.Port := strtoint(edPortClient.Text);
     ClientSocket.Open;
     edNamaClient.Enabled := true;
     showmessage('Server Tidak Ditemukan');
     btConnect.Caption := 'Disconnect';
    end;
end;

procedure TfrmClient.btFontClick(Sender: TObject);
begin
   fontdialog.Execute;
   MemKirim.Font := fontdialog.Font;
   MemPesanClient.Font := fontdialog.Font;
   MemStatusClient.Font := fontdialog.Font;
   MemChiper.Font := fontdialog.Font;
   MemKey2.Font := fontdialog.Font;
end;

procedure TfrmClient.btKirimClick(Sender: TObject);
var
 Pesan : TChatPesan;
 Chiper : string;
begin
 try
 Chiper := EkCeVi(MemKirim.Text,MemKey2.Text);
 Pesan.Nickname := ednamaclient.Text;
 Pesan.Text := Chiper;
 MemPesanClient.Lines.Add(ednamaclient.Text + ' : ' + MemKirim.Text);
 ClientSocket.Socket.SendBuf(Pesan, SizeOf(Pesan));
  if (MemKirim.Text<>'') then
```

```
begin
      MemChiper.Lines.Add(Pesan.Text);
    end;
   if ednamaclient.Enabled = true then
   else
    MemKirim.Text := '';
 except
 showmessage('Periksa Key , apakah sudah dimasukkan ?');
 end;
end;

procedure TfrmClient.btSimpanClick(Sender: TObject);
begin
 SaveTextFileDialog.Execute;
 MemChiper.Lines.SaveToFile(SaveTextFileDialog.FileName + '.txt');
end;

procedure TfrmClient.ClientSocketConnect(Sender: TObject;
 Socket: TCustomWinSocket);
begin
   MemStatusClient.Lines.Add('Status : Koneksi Terhubung');
   lblwelcome.Caption := 'ANDA SUDAH TERHUBUNG ';
end;

procedure TfrmClient.ClientSocketConnecting(Sender: TObject;
 Socket: TCustomWinSocket);
begin
   MemStatusClient.Lines.Add('Status : Koneksi dibuat');
end;

procedure TfrmClient.ClientSocketDisconnect(Sender: TObject;
 Socket: TCustomWinSocket);
begin
 MemStatusClient.Lines.Add('Status : Terputus');
 lblwelcome.Caption := '';
  try
    begin
     ClientSocket.Host := edIPClient.Text;
     ClientSocket.Port := strtoint(edPortClient.Text);
     ClientSocket.Open;
     btConnect.Caption := 'Disconnect from Server';
     ednamaclient.Enabled := false;
    end;
  except
```

```
showmessage('Tidak Bisa Dihubungi');
  end;
end;

procedure TfrmClient.ClientSocketError(Sender: TObject;
 Socket: TCustomWinSocket; ErrorEvent: TErrorEvent; var ErrorCode: Integer);
begin
 MemStatusClient.Lines.Add('Status : Tidak Adak Koneksi ke Server');
 ednamaclient.Enabled := true;
end;

procedure TfrmClient.ClientSocketLookup(Sender: TObject;
 Socket: TCustomWinSocket);
begin
  MemStatusClient.Lines.Add('Status : Sedang mencari Server');
end;

procedure TfrmClient.FormActivate(Sender: TObject);
begin
 frmClient.Caption := 'Chat (Active)';
end;

procedure TfrmClient.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  ClientSocket.Close;
end;

procedure TfrmClient.FormCreate(Sender: TObject);
begin
  MemPesanClient.ReadOnly := true;
  MemStatusClient.ReadOnly := true;
  MemChiper.ReadOnly := true;
end;

procedure TfrmClient.FormDeactivate(Sender: TObject);
begin
    frmClient.Caption := 'Chat';
end;

end.
```

4. UnitHelp.pas

```
unit UnitHelp;

interface

uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, ExtCtrls, GIFImg, Menus;

type
 Tfrmbantuan = class(TForm)
  Image1: TImage;
  MemPetunjuk: TMemo;
  MainMenu1: TMainMenu;
  MenuAnalisa: TMenuItem;
  MenuMetode: TMenuItem;
  MenuAlgoritmaCaesar: TMenuItem;
  MenuAlgoritmaVignere: TMenuItem;
  MenuKeluar: TMenuItem;
  procedure MenuAlgoritmaCaesarClick(Sender: TObject);
  procedure MenuAlgoritmaVignereClick(Sender: TObject);
  procedure MenuKeluarClick(Sender: TObject);
 private

 public

 end;

var
 frmbantuan: Tfrmbantuan;

implementation

{$R *.dfm}

uses UnitCaesar, UnitVignere;
procedure Tfrmbantuan.MenuAlgoritmaCaesarClick(Sender: TObject);
begin
 FormCaesar.Show;
end;
```

```
procedure Tfrmbantuan.MenuAlgoritmaVignereClick(Sender: TObject);
begin
  FormVignere.Show;
end;

procedure Tfrmbantuan.MenuKeluarClick(Sender: TObject);
begin
 frmBantuan.Close;
end;

end.
```

5. UnitVignere.pas

```
unit UnitVignere;

interface

uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, Menus;

type
 TFormVignere = class(TForm)
  btEnkrip: TButton;
  btDekrip: TButton;
  mKey: TMemo;
  mPlain: TMemo;
  mChiper: TMemo;
  mPlain2: TMemo;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  procedure btEnkripClick(Sender: TObject);
  procedure btDekripClick(Sender: TObject);
  procedure mPlainKeyPress(Sender: TObject; var Key: Char);
  procedure mKeyKeyPress(Sender: TObject; var Key: Char);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);

 private
```

```
public
  { Public declarations }
 end;
var
 FormVignere: TFormVignere;

implementation

{$R *.dfm}

function EkVignere(Plain,Key : string): string;
var
 i,k : integer;
 c,l : byte;
begin
 result:=Plain;
 if Length(result)>0 then
 begin
  k:=1;
  for i:=1 to Length(result) do
  begin
   c:=Ord(Plain[i])-32;
   l:=Ord(Key[k])-32;
   result[i]:=Chr(32+((c+l) mod 95) );
   if k=Length(Key) then k:=1 else inc(k);
  end;
 end;
end;

function DkVignere(Chiper,Key : string):string;
var
  k,pos : integer;
  c,l: byte;
begin
   result := Chiper;
   k:=1;
   for pos:=1 to Length(result) do
   begin
    c:=Ord(Chiper[pos])-32;
    l:=Ord(Key[k])-32;
    result[pos]:=Chr((((c+95)-l)mod 95)+32);
    if k=Length(Key) then k:=1 else inc(k);
   end;
end;
```

```
procedure TFormVignere.btDekripClick(Sender: TObject);
begin
if mChiper.Text = '' then
 begin
  MessageDlg('Pesan Sandi Belum ADA, Enkripsi Dahulu !!!',mtError,[mbOK],0);
 end
 else
mPlain2.Text := DkVignere(mChiper.Text,mKey.Text);
end;

procedure TFormVignere.btEnkripClick(Sender: TObject);
begin
 btDekrip.Enabled := true;
 if (mPlain.Text = '') or (mKey.Text = '') then
 begin
  MessageDlg ('Pesan Atau Key Belum Terisi !!!',mtError,[mbOK],0);
  FormVignere.Show;
 end
 else
 mChiper.Text := EkVignere(mPlain.Text,mKey.Text);
end;

procedure TFormVignere.FormClose(Sender: TObject; var Action: TCloseAction);
begin
 mPlain.Lines.Clear;
 mPlain2.Lines.Clear;
 mChiper.Lines.Clear;
 mKey.Lines.Clear;
end;

procedure TFormVignere.mKeyKeyPress(Sender: TObject; var Key: Char);
begin
 btDekrip.Enabled := false;
end;

procedure TFormVignere.mPlainKeyPress(Sender: TObject; var Key: Char);
begin
 btDekrip.Enabled := false;
end;

end.
```

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Rendra Puguh W.S

NIM : 04.12.654

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Judul Skripsi : **PENGEMBANGAN ALGORITMA KRYPTOGRAFI KLASIK PADA APLIKASI CHATTING DENGAN METODE CE-VI**

Dipertahankan di hadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Jum'at

Tanggal : 18 Februari 2011

Dengan Nilai : **84 (A)**

Ketua Majelis Penguji

Ir. Yusuf Ismail Nahkoda, MT.
NIP.Y. 1018800189

Sekretaris Majelis Penguji

Dr. Eng. Aryanto Soetedjo, ST, MT.
NIP.Y. 1030800417

Penguji I

Joseph Dedy Irawan,ST. MT.
NIP. 19740416 200501 1 002

Penguji II

Sonny Prasetio, ST. MT.
NIP.P 1031000433

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 20 Desember 2010

Nomor : ITN-087/I.TA/2/10
Lampiran : -
Perihal : [illegible]

Kepada : Yth. Sdr/i. **IR. YUSUF ISMAIL NAKHODA, MT**
[illegible]

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: RENDRA PUGUH. W. S
Nim	: 0412654
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai tanggal :

02 Agustus 2010 s/d 02 Februari 2011

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y. 1018800189

Tembusan Kepada Yth :
1. Mahasiswa Yang Bersangkutan
2. Arsip
3. Coret yang tidak perlu

Form. S 4a

```
function sljcgotos2(event:MouseEvent):void{
  p1 = 0;
  gotoAndStop(1165);
}
```

NILAI HASIL QUIS

```
stop();
skor = p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 + p10 +
p11 + p12 + p13 + p14 + p15 + p16 + p17 + p19 + p20;
hslnilai_dt.text = skor;
var b:int=parseInt(hslnilai_dt.text);
var c:int= b/5;
totjwbbenar_dt.text = String(c);
var d:int=20-c;
totjwbsalah_dt.text = String(d);
var e:String;
if(c <= 10){
  e = "Kurang";
  }
  else if((c > 10)&&(c < 14)){
       e = "Cukup";
       }
       else if((c > 14)&&(c <= 16)){
            e = "Baik";
            }
            else{
                 e = "Sangat Baik";
                 }
pemahaman_dt.text = e;
nnkuis_bt.addEventListener(MouseEvent.CLICK,
kemenuutama);
function kemenuutama (event:MouseEvent):void{
  sc.stop();
  gotoAndPlay(1122);
}
```

Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Teknik Elektro S-1

LEMBAR PENGAJUAN JUDUL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika/Teknik Komputer & Informatika*)

1.	Nama Mahasiswa: [illegible]			Nim: [illegible]
2.	Waktu Pengajuan	Tanggal:	Bulan:	Tahun:
		31	JULI	2009

Spesifikasi Judul (berilah tanda silang)**)

3.	a. Sistem Tenaga Elektrik	e. Elektronika & Komponen	
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer	
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi	
	d. Sistem Kendali Industri	h. lainnya	

4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen*) Dr. Supeno Djanali, ST, MSc	Ketua Jurusan Ir. F. Yudi Limpraptono,MT NIP. P. 1039500274
5.	Judul yang diajukan mahasiswa:	Pengembangan Algoritma Kriptografi Klasik pada aplikasi Chatting dengan metode C.E - N.J
6.	Perubahan judul yang disetujui Dosen sesuai materi bidang ilmu	.....
	Catatan:	.....
7.	Persetujuan Judul skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui 7/8/ 200 Dosen .....

Perhatian:

1. Formulir pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1
2. Keterangan: *) Coret yang tidak perlu
   **) dilingkari a, b, c, ... atau g sesuai bidang keahlian

Form S-2

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T.Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : RENDRA P.
NIM : 04.12.654
Perbaikan meliputi :

- SESUAIKAN RUMUS² MM DAN PROG.
- SIMULASI ICAP VAR DATA²

Malang,

( )

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T.Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Rendra Pugah W.S
NIM : 04.12.057
Perbaikan meliputi :

- Fimus Vigure
- Keterangan gambar ditambah
- Kesimpulan sesuaikan dengan penyajian

Malang, 18/2 2011

(SONNY PRASETIOSTAMI)

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : **RENDRA PUGUH W.S**
Nim : **04.12.654**
Masa Bimbingan : **2 AGUSTUS 2010 s/d 2 FEBRUARI 2011**
Judul Skripsi : **PENGEMBANGAN ALGORITMA KRYPTOGRAFI KLASIK PADA APLIKASI CHATTING DENGAN METODE CE - VI**

).	Tanggal	Uraian	Paraf Pembimbing
	27-01-2011	BAB I PENDAHULUAN	
	27-01-2011	BAB II DASAR TEORI	
	27-01-2011	BAB III ANALISIS DAN PERANCANGAN SISTEM	
	27-01-2011	BAB IV IMPLEMENTASI	
	27-01-2011	BAB V PENUTUP	

Malang,
Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

Form S-4B

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : **RENDRA PUGUH W.S**
Nim : **04.12.654**
Masa Bimbingan : **2 AGUSTUS 2010 s/d 2 FEBRUARI 2011**
Judul Skripsi : **PENGEMBANGAN ALGORITMA KRYPTOGRAFI KLASIK PADA APLIKASI CHATTING DENGAN METODE CE - VI**

o.	Tanggal	Uraian	Paraf Pembimbing
.	20-01-2011	BAB I PENDAHULUAN	
.	20-01-2011	BAB II DASAR TEORI	
.	20-01-2011	BAB III ANALISIS DAN PERANCANGAN SISTEM	
.	20-01-2011	BAB IV IMPLEMENTASI	
.	20-01-2011	BAB V PENUTUP	
.			
.			
.			
.			
.			

Malang,
Dosen Pembimbing II

Dr. Eng. Aryuanto Soetedjo, MT.
NIP.Y. 1030800417

Form S-4B

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No.2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama	: Rendra Puguh W.S
NIM	: 04.12.654
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Komputer dan Informatika
Masa Bimbingan	: 2 Agustus 2010 s/d 2 Februari 2011
Judul Skripsi	: **PENGEMBANGAN ALGORITMA KRYPTOGRAFI KLASIK PADA APLIKASI CHATTING DENGAN METODE CE-VI**

Tanggal	Uraian	Paraf
Penguji I 18 Februari 2011	Sesuaikan rumus – rumusnya dengan program	
	Simulasikan dengan data - data	
Penguji II 18 Februari 2011	Rumus Vignere	
	Keterangan gambar ditambah	
	Kesimpulan sesuaikan dengan pengujian	

Disetujui :

Penguji I

Joseph Dedy Irawan,ST. MT.
NIP. 19740416 200501 1 002

Penguji II

Sonny Prasetio, ST, MT.
NIP.P 1031000433

Mengetahui :

Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

Dosen Pembimbing II

Dr. Eng. Aryuanto Soetedjo, MT.
NIP.Y. 1030800417

INSTITUT TEKNOLOGI NASIONAL
Jl. Sigura-gura No 2
M A L A N G

Lampiran : 1 (Satu) Berkas
Pembimbing Skripsi

Kepada : Yth. Ir. Yusuf Ismail Nakhoda. MT.
Dosen Institut Teknologi Nasional
M A L A N G

Yang bertanda tangan di bawah ini:

Nama : RENDRA PUGUH W.S
Nim : 04.12.654
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing (Utama / ~~Pendamping~~ *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

"PENGEMBANGAN ALGORITMA KRYPTOGRAFI KLASIK PADA APLIKASI CHATTING DENGAN METODE CE - VI"

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Skripsi Sarjana Teknik.
Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami ucapkan terima kasih.

Malang, 6 Agustus 2009

Hormat kami,

Rendra Puguh W.S

Ketua
Jurusan Teknik Elektro S-1,

Ir.F. Yudi Limpraptono. MT
NIP.Y. 1039500274

*) coret yang tidak perlu

Form S-3a

INSTITUT TEKNOLOGI NASIONAL
Jl. Sigura-gura No 2
M A L A N G

Lampiran : 1 (Satu) Berkas
Pembimbing Skripsi

Kepada : Yth. Dr. Eng. Aryuanto Soetedjo, MT.
Dosen Institut Teknologi Nasional
M A L A N G

Yang bertanda tangan di bawah ini:

Nama : RENDRA PUGUH W.S
Nim : 04.12.654
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing (~~Utama~~ / Pendamping *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

"PENGEMBANGAN ALGORITMA KRYPTOGRAFI KLASIK PADA APLIKASI CHATTING DENGAN METODE CE - VI"

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Skripsi Sarjana Teknik.
Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami ucapkan terima kasih.

Malang, 6 Agustus 2009

Hormat kami,

Ketua
Jurusan Teknik Elektro S-1,

Ir.F. Yudi Limpraptono, MT
NIP.Y. 1039500274

Rendra Puguh W.S

*) coret yang tidak perlu

Form S-3a