SISTEM PENGKODEAN *VIGENERE* DENGAN KUNCI BARISAN KARAKTER

SKRIPSI

Oleh

Yulan Isa Puspita
NIM 071810101074

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2015

SISTEM PENGKODEAN *VIGENERE* DENGAN KUNCI BARISAN KARAKTER

SKRIPSI

diajukan guna melengkapi dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

Yulan Isa Puspita
NIM 071810101074

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2015

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. kedua orang tua tercinta, Bapak Suyut Ismulat dan Ibu Siti Chotijah, yang telah mendoakan, memberi kasih sayang tanpa batas serta pengorbanan selama ini, semoga Allah selalu mendekap erat dengan kasih sayang-Nya;
2. sahabatku Wika Anggani yang selalu memberi semangat, inspirasi, keceriaan, dukungan, serta selalu setia menemani dalam suka maupun duka;
3. guru-guru sejak taman kanak-kanak hingga perguruan tinggi, yang telah memberikan ilmu serta bimbingan dengan penuh kesabaran;
4. almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTTO

Yakinlah atas segala usaha yang yang telah kamu lakukan,
serahkan hasilnya kepada Allah S.W.T dan jika niatmu baik,
maka Allah S.W.T pasti akan memberi lebih atas semua yang kamu harapkan
(Terjemahan Surat Al Kahfi ayat 45)[*)]

Sesungguhnya sesudah kesulitan itu ada kemudahan,
maka apabila kamu telah selesai dari suatu urusan,
kerjakanlah dengan sungguh-sungguh urusan yang lain
(Terjemahan Surat Al Insyiroh ayat 94)[*)]

[*)] Departemen Agama Republik Indonesia. 1998. *Al Qur'an dan Terjemahannya*. Semarang: PT. Karya Toha Putra.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Yulan Isa Puspita

NIM : 071810101074

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul **"***Sistem Pengkodean Vigenere dengan Kunci Barisan Karakter***"** adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, Januari 2015

Yang menyatakan,

Yulan Isa Puspita

NIM 071810101074

SKRIPSI

SISTEM PENGKODEAN *VIGENERE*
DENGAN KUNCI BARISAN KARAKTER

Oleh

Yulan Isa Puspita
NIM 071810101074

Pembimbing

Dosen Pembimbing Utama : Kiswara Agung Santoso, S.Si., M.Kom.

Dosen Pembimbing Anggota : Kusbudiono, S.Si., M.Si.

PENGESAHAN

Skripsi yang berjudul "*Sistem Pengkodean Vigenere dengan Kunci Barisan Karakter*" telah diuji dan disahkan pada:

hari :

tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji :

Ketua,	Sekretaris,
Kiswara Agung Santoso, S.Si., M.Kom.	Kusbudiono, S.Si., M.Si.
NIP 19720907 199803 1 003	NIP 19770430 200501 1 001
Anggota I,	Anggota II,
Ika Hesti Agustin, S.Si, M.Si.	Ahmad Kamsyakawuni, S.Si., M.Kom.
NIP 19840801 200801 2 006	NIP 19721129 199802 1 001

Mengesahkan,

Dekan,

Prof. Drs. Kusno, DEA., Ph.D.

NIP 19610108 198602 1 001

RINGKASAN

Sistem Pengkodean *Vigenere* dengan Kunci Barisan Karakter; Yulan Isa Puspita; 071810101074; 2015: 38 halaman; Jurusan Matematika Fakultas MIPA Universitas Jember.

Kriptografi merupakan seni dan ilmu menyembunyikan informasi dari pihak yang tidak berhak. Didalam kriptografi terdapat dua proses penting, yaitu enkripsi dan dekripsi. Enkripsi adalah sebuah proses penyandian yang melakukan perubahan pesan yang dapat dimengerti maknanya (plainteks) menjadi sebuah pesan yang tidak dapat dimengerti maknanya (cipherteks). Sedangkan proses dekripsi merupakan kebalikan dari proses enkripsi, yaitu proses perubahan cipherteks menjadi plainteks. Kedua proses tersebut memerlukan suatu mekanisme dan kunci tertentu. Salah satu algoritma kriptografi adalah *Vigenere Cipher*. Algoritma ini termasuk *polyalphabetical subtitution cipher* (cipher abjad majemuk) karena enkripsi terhadap satu huruf yang sama dapat menghasilkan huruf yang berbeda sehingga lebih sulit untuk menemukan pola enkripsinya.

Pada penelitian ini penulis melakukan modifikasi pada pembentukan kunci yang digunakan *Vigenere Cipher*. Pada *Vigenere* umumnya, karakter kunci disubtitusi secara langsung dengan karakter plainteks sehingga menghasilkan cipherteks. Sedangkan pada skripsi ini, sebelum karakter kunci disubtitusikan dengan karakter plainteks yang bersesuaian, karakter kunci tersebut harus melalui beberapa tahapan terlebih dahulu agar didapatkan suatu nilai baru. Nilai-nilai yang dihasilkan tersebut kemudian disubtitusikan dengan karakter plainteks yang bersesuaian sehingga menghasilkan cipherteks. Tujuan yang ingin dicapai dalam penulisan skripsi ini adalah mendapatkan suatu sistem pengkodean dengan menggunakan metode *Vigenere Cipher* dengan kunci berupa barisan karakter. Hasil penelitian ini diharapkan dapat memberikan sebuah alternatif dalam usaha untuk menyembunyikan

suatu informasi penting sehingga dapat menyulitkan pihak-pihak tidak berhak yang ingin mengetahui informasi tersebut.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala limpahan rahmat serta karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "*Sistem Pengkodean Vigenere dengan Kunci Barisan Karakter*". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak, oleh karena itu penulis ingin menyampaikan terima kasih kepada:

1. Prof. Drs. Kusno, DEA., Ph.D., selaku Dekan FMIPA Universitas Jember
2. Bapak Kiswara Agung Santoso, S.Si., M.Kom., selaku Dosen Pembimbing Utama dan Bapak Kusbudiono, S.Si., M.Si., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatiannya dalam penulisan skripsi ini;
3. Ibu Ika Hesti Agustin, S.Si., M.Si., selaku Dosen Penguji I dan Bapak Ahmad Kamsyakawuni, S.Si., M.Kom., selaku Dosen Penguji II yang telah memberikan saran dan kritik demi terselesaikannya penulisan skripsi ini;
4. Ibu Dian Anggraeni, S.Si., M.Si., selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama menjadi mahasiswa;
5. Bapak Suyut Ismulat dan Ibu Siti Chotijah yang telah memberikan doa, dorongan semangat, serta nasihat demi terselesaikannya skripsi ini;
6. Angga Bayu Prasetyo, Wika Anggani, Feiruz, Tita, Mas Fiqi, Vivi, Dani, Mas Arif Fajar, serta teman-teman angkatan 2007 yang telah menemani dan membantu dalam menyelesaikan skripsi ini;
7. Semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menerima kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Januari 2015 Penulis

DAFTAR ISI

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Saat ini ilmu dan teknologi informasi berkembang pesat seiring dengan perkembangan zaman. Salah satu hasil teknologi yang dikembangkan adalah komputer, karena komputer memiliki peran yang sangat penting dalam berbagai bidang. Di era internet seperti sekarang ini, selain digunakan untuk menyimpan berbagai macam informasi, komputer juga digunakan sebagai alat komunikasi, baik itu yang bersifat umum maupun rahasia. Pesan yang dikirimkan dari pengirim ke penerima melalui suatu media transmisi pesan memiliki resiko untuk disadap oleh pihak yang tidak berkepentingan. Untuk itu diperlukan adanya teknologi keamanan informasi untuk meningkatkan sistem keaman informasi tersebut.

Untuk menjaga kerahasiaan pesan terdapat dua macam pendekatan yang umum digunakan, yaitu dengan menggunakan jalur komunikasi yang terjamin keamanannya atau dengan menyandikan pesan dalam bentuk yang hanya dapat dibaca oleh pihak berkepentingan. Pendekatan yang banyak digunakan adalah penyandian pesan, hal ini dikarenakan menjaga keamanan jalur komunikasi membutuhkan biaya yang cukup besar jika dibandingkan dengan menjaga kerahasiaan pesan dengan penyandian. Untuk menjaga kerahasiaan pesan maka digunakan ilmu penyandian, yaitu kriptografi.

Kriptografi adalah ilmu atau seni untuk menjaga keamanan pesan. Didalam kriptografi terdapat dua proses penting, yaitu enkripsi dan dekripsi. Enkripsi adalah sebuah proses penyandian yang melakukan perubahan sebuah pesan yang bisa dimengerti maknanya (plainteks) menjadi sebuah pesan yang tidak bisa dimengerti maknanya (cipherteks). Sedangkan proses dekripsi merupakan kebalikan dari proses enkripsi, yaitu proses perubahan cipherteks menjadi plainteks. Kedua proses tersebut

memerlukan suatu mekanisme dan kunci tertentu. Berdasarkan jumlah kunci yang digunakan, kriptografi terbagi menjadi dua metode yaitu kriptografi klasik dan kriptografi modern. Kunci yang digunakan pada kriptografi klasik terdiri dari satu jenis kunci, sedangkan pada kriptografi modern terdiri dari dua kunci yang berbeda.

Vigenere Cipher adalah salah satu metode dari kriptografi klasik yang merupakan perluasan dari metode *Caesar Cipher*. Konsep dasar yang dipakai pada metode *Caesar Cipher* adalah pergeseran tiap karakternya, sedangkan pada *Vigenere Cipher* membutuhkan sebuah tabel bujur sangkar yang disebut bujur sangkar *Vigenere*. Pada bujur sangkar ini, kolom paling kiri menyatakan huruf-huruf kunci, baris paling atas menyatakan huruf-huruf dari plainteks, sedangkan setiap baris di dalam bujur sangkar menyatakan huruf-huruf cipherteks. *Vigenere Cipher* mempunyai ciri khas yaitu jika panjang kunci lebih pendek dari panjang plainteks maka kunci akan diulang secara periodik sepanjang plainteks. hal ini sekaligus menjadi kelemahan dari *Vigenere Cipher*, karena dengan perulangan kunci tersebut akan mengakibatkan pihak-pihak yang tidak berhak lebih mudah memecahkan sandi. Salah satu cara untuk mengatasi kelemahan tersebut adalah dengan melakukan modifikasi pada pengkodean *Vigenere.*

Sebelumnya, beberapa penelitian telah dilakukan mengenai kriptografi, diantaranya seperti Sistem Pengkodean Gabungan *Affine* dan *Merkle-Hellman* (Guntoro, 2011), Penyandian *Hill* menggunakan Kode *Plaintext* Kembar (Asmara, 2009), Sistem Pengkodean *Playfair-Vigenere* (Rosidah, 2009). Pada penelitian tersebut menghasilkan kesimpulan bahwa semakin banyak langkah-langkah yang dibutuhkan untuk memecahkan cipherteks, maka pengkodean tersebut dapat dikatakan lebih aman.

Berbeda dengan penelitian yang dilakukan sebelumnya, pada penelitian ini penulis melakukan modifikasi pada kunci yang digunakan *Vigenere Cipher*. Pada *Vigenere* umumnya, proses enkripsi dilakukan dengan mensubtitusikan karakter kunci dengan karakter plainteks sehingga menghasilkan cipherteks begitu juga pada proses dekripsinya. Sedangkan pada skripsi ini, sebelum karakter kunci disubtitusikan

dengan karakter plainteks yang bersesuaian, karakter kunci tersebut harus melalui beberapa tahapan terlebih dahulu agar diperoleh suatu karakter kunci yang baru. Kunci baru yang dihasilkan tersebut kemudian disubtitusikan dengan karakter plainteks sehingga menghasilkan cipherteks. Hal tersebut dilakukan dengan harapan agar dapat meningkatkan performa sekaligus menutupi kelemahan dari metode *Vigenere*.

1.2 RumusanMasalah

Permasalahan yang akan dibahas pada skripsi ini adalah bagaimana cara mengubah plainteks menjadi cipherteks dan juga sebaliknya menggunakan algoritma *Vigenere* dengan kunci barisan karakter.

1.3 Batasan Masalah

Dalam skripsi ini karakter yang digunakan berada dalam jangkauan ASCII (*American Standart Code for Information Interchange*), yaitu karakter 32-126. Karakter- karakter tersebut dipilih karena merupakan karakter yang paling umum digunakan serta merupakan karakter yang dapat dicetak (*printable characters*).

1.4 Tujuan

Tujuan yang ingin dicapai dalam penulisan skripsi ini adalah untuk membuat sistem pengkodean *Vigenere* dengan kunci barisan karakter serta mendeskripsikan cara pembentukan kunci untuk proses enkripsi dan dekripsinya.

1.5 Manfaat

Manfaat dari penulisan skripsi ini adalah dapat meningkatkan pemahaman tentang kriptografi bagi penulis maupun pembaca, serta cara menjaga keamanan pesan dengan menggunakan sistem pengkodean *Vigenere* dengan kunci barisan karakter.

BAB 2. TINJAUAN PUSTAKA

Dalam bab ini dibahas mengenai dasar-dasar teori yang digunakan dalam sistem pengkodean *Vigenere* dengan kunci barisan karakter. Dasar-dasar teori tersebut antara lain: bilangan bulat, bilangan prima, dan aritmetika modulo.

2.1 Bilangan Bulat

Bilangan bulat adalah bilangan yang tidak mempunyai pecahan desimal, misalnya 8, 32, 758, 65, 0, dan sebagainya. Berlawanan dengan bilangan bulat adalah bilangan riil yang mempunyai titik desimal, seperti 8,8; 34,5; 0,02, dan sebagainya.

Definisi 2.1 Misalkan a dan b adalah dua buah bilangan bulat dengan syarat $a \neq 0$. Dinyatakan bahwa a habis membagi b jika terdapat bilangan bulat c sedemikian sehingga $b = ac$.

Secara umum, jika hasil pembagian bilangan bulat dinyatakan sebagai bilangan bulat juga, maka sembarang bilangan bulat bila dibagi dengan suatu bilangan bulat positif, maka selalu terdapat hasil bagi dan sisa pembagian. Sifat ini terdapat pada Teorema 2.1.

Teorema 2.1 (Teorema Euclidean) Misalkan m dan n adalah dua buah bilangan bulat dengan syarat $n > 0$. Jika m dibagi dengan n maka terdapat dua buah bilangan bulat unik q (*quotient*) dan r (*remainder*), sedemikian sehingga

$$m = nq + r \tag{2.1}$$

dengan $0 \leq r < n$ (Munir, 2012).

2.2 Bilangan Prima

Bilangan prima adalah bilangan bulat positif yang lebih besar dari 1 yang hanya habis dibagi oleh 1 dan dirinya sendiri.

Definisi 2.2 Bilangan bulat $p > 1$ dikatakan prima jika ia hanya mempunyai pembagi p dan 1.

Dengan kata lain bilangan prima tidak mempunyai pembagi selain dari 1 dan dirinya sendiri. Karena bilangan prima harus lebih besar dari 1, maka barisan bilangan prima dimulai dari 2, yaitu 2, 3, 5, 7, ... Seluruh bilangan prima adalah bilangan ganjil, kecuali 2 yang merupakan bilangan genap. Bilangan selain prima disebut bilangan komposit (*composite*). Bila n komposit maka ia dapat dinyatakan sebagai $n = ab$, dimana $a,b \in Z$, $1 < a < n$, $1 < b < n$.

Definisi 2.3 Jika n adalah suatu bilangan komposit maka n mempunyai suatu faktor prima p sedemikian sehingga $p^2 \leq n$.

Untuk menguji apakah n merupakan bilangan prima atau komposit, cukup dilakukan pembagian n dengan sejumlah bilangan prima mulai dari 2, 3, ... , bilangan prima $\leq \sqrt{n}$. Jika n habis di bagi dengan salah satu dari bilangan prima tersebut, maka n adalah bilangan komposit, tetapi jika n tidak habis dibagi oleh semua bilangan prima tersebut, maka n adalah bilangan prima (Munir, 2012).

2.3 Aritmetika Modulo

Aritmetika modulo memainkan peran yang penting dalam aplikasi kriptografi. Operator yang digunakan adalah mod. Operator mod, jika digunakan pada pembagian bilangan bulat akan memberikan sisa pembagian sebagai kembaliannya.

Definisi 2.4 Misalkan a adalah bilangan bulat dan m adalah bilangan bulat > 0. Operasi *a* **mod** *m* (dibaca ”*a* modulo *m*”) memberikan sisa jika *a* dibagi dengan *m*. Dengan kata lain, *a* **mod** *m* = *r* sedemikian sehingga $a = mq + r$, dengan $0 \le r < m$.

Definisi 2.5 Jika *m* bilangan bulat positif dan *a* serta *b* bilangan bulat sebarang, maka dikatakan bahwa *a* setara *b* modulo *m*, ditulis sebagai:

$$a \equiv b(\text{mod } m)$$

jika $a-b$ adalah bilangan bulat kelipatan *m*.

Untuk modulus *m* sebarang dapat dibuktikan bahwa setiap bilangan bulat *a* adalah setara, modulo *m*, terhadap salah satu bilangan bulat

$$0,1,2,\cdots,m-1$$

bilangan bulat ini disebut sisa (*residue*) dari *a* modulo m, dan dituliskan:

$$Z_m = \{0,1,2,\cdots,m-1\}$$

(Asmara, 2009).

2.4 Kriptografi

2.4.1 Pengertian Kriptografi

Kata kriptografi berasal dari bahasa Yunani yaitu ”*criptos*” dan ”*graphein*”. *Criptos* artinya rahasia, sedangkan *graphein* artinya tulisan. Kriptografi diartikan sebagai ilmu sekaligus seni untuk menjaga kerahasiaan dan keamanan pesan dengan cara menyandikan kedalam bentuk yang tidak dapat dimengerti maknanya. Kemudian seiring perkembangan, kriptografi tidak lagi sebatas menyandikan pesan, tetapi juga memberikan aspek keamanan yang lain seperti serangan dari kriptanalis yaitu orang yang ingin membuka pesan tanpa memiliki kunci. Karena itu pengertian kriptografipun berubah menjadi ilmu sekaligus seni untuk menjaga keamanan pesan (Ekaputri dalam Rosidah (2009)).

Dalam menjaga kerahasiaan data dengan kriptografi, data sederhana yang dikirim (plainteks) diubah kedalam bentuk data sandi (cipherteks), kemudian data sandi tersebut hanya dapat dikembalikan kebentuk data sebenarnya hanya dengan menggunakan kunci (*key*) tertentu yang dimiliki oleh pihak yang sah saja.

Tentunya hal ini menyebabkan pihak lain yang tidak memiliki kunci tersebut tidak akan dapat membaca data yang sebenarnya sehingga dengan kata lain data akan tetap terjaga kerahasiaannya.

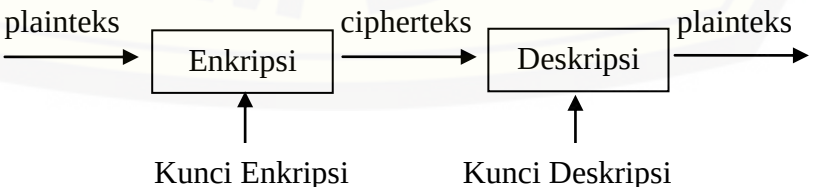
2.4.2 Konsep Kriptografi

Tujuan dari kriptografi:

a. kerahasiaan (*confidentiality*), yaitu menjaga supaya pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak
b. integritas data (*data integrity*), yaitu memberikan jaminan bahwa untuk tiap bagian pesan tidak akan mengalami perubahan dari saat data dikirim oleh pengirim sampai dengan saat data tersebut dibuka oleh penerima data
c. otentikasi (*authentication*), yaitu berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi maupun mengidentifikasi kebenaran sumber pesan
d. anti-penyangkalan (*non repudiation*), yaitu memberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang tertentu sehingga apabila ada seseorang yang mencoba mengakui memiliki dokumen tersebut, dapat dibuktikan kebenarannya dari pengakuan orang tersebut (Menezes, 1996).

Algoritma kriptografi terdiri dari tiga komponen dasar, yaitu:

a. enkripsi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiaannya. Pesan asli (plainteks) diubah menjadi kode-kode yang tidak dimengerti atau disebut cipherteks.
b. dekripsi, merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi (cipherteks) dikembalikan ke bentuk semula (plainteks).
c. Kunci (*key*), yang dimaksud adalah kunci yang digunakan untuk melakukan enkripsi dan dekripsi. Secara umum fungsi tersebut dapat digambarkan :



Gambar 2.1 Proses Enkripsi dan Dekripsi

2.4.3 Prinsip kerja kriptografi

Pembakuan penulisan pada kriptografi dapat ditulis dalam notasi matematis. Fungsi-fungsi yang mendasar dalam kriptografi adalah fungsi enkripsi dan fungsi dekripsi. Enkripsi adalah proses mengubah suatu pesan asli (plainteks) menjadi suatu pesan dalam pesan sandi (cipherteks). Jika cipherteks dilambangkan dengan *C* dan plainteks dilambangkan dengan *P*, maka fungsi enkripsi *E* memetakan *P* ke *C*,

$$E\,(P) = C \qquad (2.2)$$

sedangkan dekripsi adalah proses mengubah pesan dalam suatu bahasa sandi menjadi pesan asli kembali. Dengan kata lain, proses dekripsi merupakan kebalikan dari proses enkripsi. Fungsi dekripsi *D* memetakan *C* ke *P*,

$$D\,(C) = P \qquad (2.3)$$

Umumnya, selain menggunakan fungsi tertentu dalam melakukan enkripsi dan dekripsi, seringkali fungsi itu diberi parametertambahan yang disebut dengan istilah kunci (Munir, 2012).

2.4.4 Jenis-Jenis Serangan

Selain terdapat pihak yang ingin menjaga agar pesan tetap aman, terdapat juga pihak-pihak yang ingin mengetahui pesan rahasia tersebut secara tidak sah. Bahkan ada pihak-pihak yang ingin agar dapat mengubah isi pesan tersebut. Menurut Nugraha (2009), ilmu untuk mendapatkan pesan yang asli dari pesan yang telah disandikan tanpa memiliki kunci untuk membuka pesan rahasia tersebut disebut kriptanalisis. Sedangkan usaha untuk membongkar suatu pesan sandi tanpa memiliki kunci dengan cara yang sah dikenal dengan istilah (*attack*).

Penyerangan terhadap pesan yang sudah dienkripsi terbagi menjadi tiga, yaitu (Nugraha, 2009).

1. *Ciphertext only attack*, artinya adalah penyerang hanya mendapatkan pesan yang sudah tersandikan saja.
2. *Choosen plaintext attack*, dimana penyerang mendapatkan cipherteks dan memiliki penggalan plainteks.

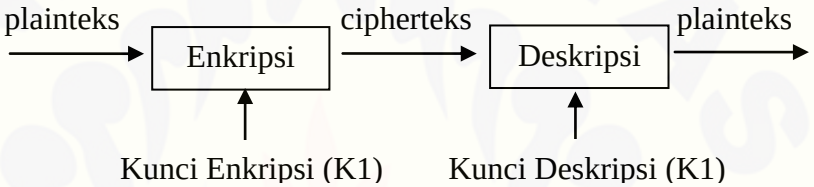
3. *Known plaintext attack*, dimana penyerang selalin mendapatkan sandi juga mendapatkan pesan asli.

2.4.5 Algoritma Simetri dan Asimetri

Terdapat dua jenis algoritma dalam kriptografi.

a. Algoritma Simetri

Algoritma simetri (*symmetric algorithm*) adalah suatu algoritma dimana kunci enkripsi yang digunakan sama dengan kunci dekripsi sehingga algoritma ini disebut juga sebagai *single-key* algorithm.



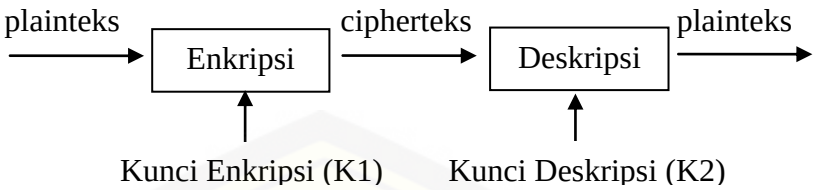
Gambar 2.2 Proses Enkripsi dan Dekripsi Algoritma Simetris

Sebelum melakukan pengiriman pesan, pengirim dan penerima harus memilih suatu kunci tertentu yang sama untuk dipakai bersama, dan kunci ini haruslah rahasia bagi pihak yang tidak berkepentingan sehingga algoritma ini disebut juga algoritma kunci rahasia (*secret-key algorithm*).

b. Algoritma Asimetri

Algoritma asimetri (*asymmetric algorithm*) adalah suatu algoritma dimana kunci enkripsi yang digunakan tidak sama dengan kunci dekripsi. Pada algoritma ini kunci yang digunakan ada dua, yakni kunci publik (*public key*) dan kunci privat (*private key*). Kunci publik disebarkan secara umum, sedangkan kunci privat disimpan secara rahasian oleh si pengguna. Walaupun kunci publik telah diketahui namun akan sangat sukar mengetahui kunci privat yang digunakan.

Pada umumnya kunci publik digunakan sebagai kunci enkripsi sementara kunci privat digunakan sebagai kunci dekripsi.



Gambar 2.3 Diagram Proses Enkripsi dan Dekripsi Algoritma Asimetris

2.5 *Vigenere Cipher*

Pada abad ke-16, Blaise de Vigenere mengembangkan sebuah kode substitusi baru yang merupakan perluasan dari penggunaan *Caesar Chiper* berdasarkan kunci yang dinamakan *Vigenere Cipher*. Pada metode *Caesar Cipher*, setiap huruf alfabet pada plainteks akan disubtitusi sepanjang 3 huruf sesudah huruf tersebut. Sebagai contoh, huruf A akan diganti dengan huruf D, B akan diganti dengan huruf E, Y akan diganti dengan huruf B dengan metode *Caesar Cipher*.

Vigenere Cipher termasuk *polyalphabetical subtitution cipher* (cipher abjad majemuk) karena enkripsi terhadap satu huruf yang sama bisa menghasilkan huruf yang berbeda. Tujuan utama dari *Vigenere Cipher* ini adalah menyembunyikan keterhubungan antara plainteks dan cipherteks dengan menggunakan kata kunci sebagai penentu pergeseran karakternya. *Vigenere Cipher* menggunakan sebuah tabel bujur sangkar *Vigenere* seperti pada tabel 2.1 untuk memperoleh cipherteks dengan menggunakan kunci yang sudah ditentukan. Pada bujursangkar ini, kolom paling atas menyatakan huruf-huruf plainteks, kolom paling kiri bujursangkar menyatakan kunci-kunci, dan setiap baris di dalam bujursangkar menyatakan huruf-huruf cipherteks.

Tabel 2.1 Bujursangkar *Vigenere*

	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

Algoritma ini bersifat periodik, yang artinya proses enkripsi terhadap keseluruhan plainteks mengikuti periode tertentu, yang besarnya sama dengan panjang kunci yang digunakan. Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci akan diulang penggunaannya. Apabila panjang kunci yang digunakan adalah *m*, maka periodenya dikatakan *m*. Proses enkripsi pada algoritma *Vigenere* pada dasarnya menggunakan prinsip *Caesar cipher*, yaitu melakukan enkripsi karakter pada plainteks menjadi karakter lain pada cipherteks.

Prosedur enkripsi pada *Vigenere Cipher* dapat dijelaskan melalui contoh sebagai berikut:

Plainteks : CONTOH PROSES ENKRIPSI VIGENERE

Kunci : COBA

Langkah 1 : Hilangkan spasi dari plainteks

CONTOHPROSESENKRIPSIVIGENERE

Langkah 2 : Tuliskan kunci secara periodik

CONTOHPROSESENKRIPSIVIGENERE

COBACOBACOBACOBACOBACOBACOBA

Langkah 3 : Enkripsikan setiap karakter plainteks dengan karakter kunci yang bersesuaian, menggunakan Bujursangkar *Vigenere*.

Proses enkripsi dengan menggunakan Bujursangkar *Vigenere* (lihat Tabel 2.1) dilakukan dengan cara mencari perpotongan antara baris dari karakter kunci dengan kolom dari karakter plainteks. Jadi, untuk karakter ketiga misalnya, cari perpotongan antara baris B dan kolom N. Karakter yang merupakan perpotongan antara baris dan kolom menjadi subtitusi dari karakter plainteks yang bersangkutan yaitu O. Dengan mengulangi langkah ini untuk setiap karakter dari plainteks, maka akan didapatkan cipherteks

ECOTQVQRQGFSGBLRKDTIXWHEPSSE

sedangkan untuk dekripsi adalah kebalikannya. Misalkan untuk karakter pertama cipherteks E. Dekripsi dilakukan dengan cara mencari karakter pertama kata kunci pada baris E, yaitu karakter C. Kemudian dapat dilihat pada baris C, karakter E terdapat pada kolom C, hal ini menunjukkan bahwa karakter A merupakan karakter pertama plainteks. Dengan mengulangi langkah ini untuk setiap karakter cipherteks, maka akan didapatkan plainteks semula.

Selain itu, algoritma *Vigenere* dapat disajikan dalam bentuk aljabar. Jika karakter huruf diberi nomor 0 sampai *m*-1dan kemudian dilakukan operasi modulo *m*, maka proses enkripsi dapat ditulis dalam persamaan berikut (Ariesanda, 2009):

$$C_i \equiv (P_i + K_i) \bmod m \qquad (2.4)$$

dimana:

C_i = representasi numerik dari karakter cipherteks ke-*i;*

P_i = representasi numerik dari karakter plainteks ke-*i*;

K_i = representasi numerik dari karakter kunci (yang berulang) ke-*i*;

i = [1 ... jumlah karakter plainteks]

m = bilangan bulat positif

sedangkan proses dekripsinya menggunakan persamaan berikut :

$$P_i \equiv (C_i - K_i) \bmod m \qquad (2.5)$$

Jika $C_i < K_i$ maka $C_i = C_i + m$.

2.6 Modifikasi Kunci *Vigenere Cipher*

Pada skripsi ini, penulis memodifikasi kunci yang digunakan untuk proses enkripsi dan dekripsi algoritma *Vigenere*. Jika pada *Vigenere* umumnya karakter kunci disubtitusikan secara langsung dengan karakter plainteks sehingga menghasilkan cipherteks, maka pada algoritma *Vigenere* ini, kunci tersebut harus melalui beberapa tahapan tertentu terlebih dahulu sehingga menghasilkan suatu kunci baru yang kemudian digunakan untuk proses enkripsi dan dekripsi. Kunci awal penulis definisikan sebagai Kunci *Q*, sedangkan kunci akhir didefinisikan sebagai Kunci *K*. Tahap-tahap untuk mengubah Kunci *Q* menjadi Kunci *K* adalah:

a. mengubah karakter Kunci *Q* kebentuk numerik
b. nilai numerik dari Kunci *Q* digunakan untuk mencari nilai Kunci *K* melalui persamaan:

$$K_i = \begin{cases} 2q_i - 1, \text{ untuk } i = \text{prima} \\ 3q_i + 5, \text{ untuk } i = \text{ganjil tidak prima} \\ 2q_i + 2, \text{ untuk } i = \text{genap tidak prima} \end{cases} \qquad (2.6)$$

dimana:

q_i = representasi numerik untuk Kunci *Q* (yang berulang) ke-*i*
K_i = nilai Kunci *K* ke-*i*
i = [1 ... jumlah karakter plainteks].

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

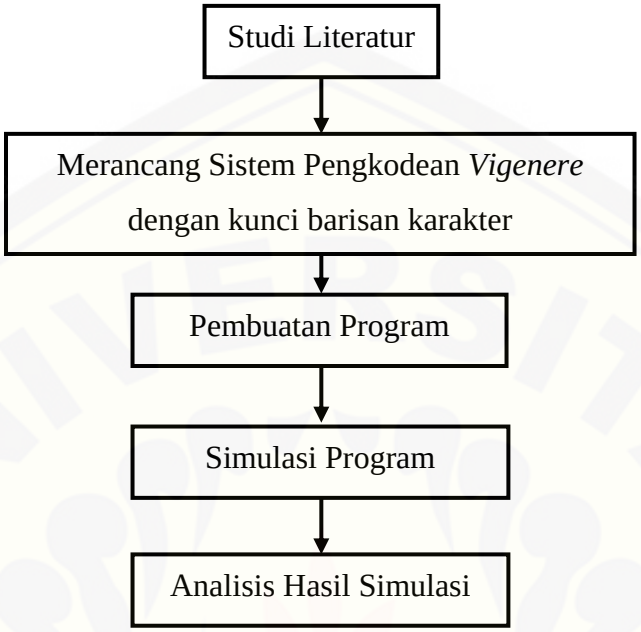
Pada penelitian ini, data yang digunakan sebagai plainteks, kunci, dan cipherteks dapat dilihat pada tabel 3.1. Data tersebut bersumber pada Tabel ASCII. Tabel ASCII secara lengkap dapat dilihat pada lampiran A (halaman 39).

Tabel 3.1 Nilai numerik dari karakter

space 32	**!** 33	**“** 34	**#** 35	**$** 36	**%** 37	**&** 38	**‘** 39	**(** 40	**)** 41
*** 42	+ 43	**,** 44	**-** 45	**.** 46	/ 47	**0** 48	**1** 49	**2** 50	**3** 51
4 52	**5** 53	**6** 54	**7** 55	**8** 56	**9** 57	**:** 58	**;** 59	< 60	= 61
> 62	**?** 63	**@** 64	**A** 65	**B** 66	**C** 67	**D** 68	**E** 69	**F** 70	**G** 71
H 72	**I** 73	**J** 74	**K** 75	**L** 76	**M** 77	**N** 78	**O** 79	**P** 80	**Q** 81
R 82	**S** 83	**T** 84	**U** 85	**V** 86	**W** 87	**X** 88	**Y** 89	**Z** 90	**[** 91
\ 92	**]** 93	^ 94	**_** 95	**`** 96	**a** 97	**b** 98	**c** 99	**d** 100	**e** 101
f 102	**g** 103	**h** 104	**i** 105	**j** 106	**k** 107	**l** 108	**m** 109	**n** 110	**o** 111
p 112	**q** 113	**r** 114	**s** 115	**t** 116	**u** 117	**v** 118	**w** 119	**x** 120	**y** 121
z 122	**{** 123	**|** 124	**}** 125	**~** 126					

Sumber: Tabel ASCII

3.2 Langkah Penelitian



Gambar 3.1 Skema Langkah-langkah Penelitian

Penjelasan langkah-langkah penelitian pada skema diatas adalah sebagai berikut.

a. Studi Literatur

Langkah awal yang dilakukan adalah mengumpulkan dan mempelajari berbagai literatur yang berkaitan dengan kriptografi, khususnya pengkodean *Vigenere*.

b. Merancang sistem pengkodean *Vigenere* dengan kunci barisan karakter. Terdapat tiga tahap dalam langkah ini, antara lain:

1. Pembentukan Kunci (*key*)

Pada proses ini, Kunci *Q* diubah menjadi Kunci *K* dengan cara:

a) mengubah Kunci *Q* kebentuk numerik sesuai tabel 3.1
b) mencari Kunci *K* dengan menggunakan persamaan 2.6.

Apabila panjang Kunci *Q* lebih pendek dari panjang plainteks maka Kunci *Q* diulang secara periodik sepanjang plainteks sehingga letak posisi dari

perulangan Kunci Q bersesuaian dengan posisi plainteks. Sedangkan jika panjang Kunci Q lebih panjang dari panjang plainteks, maka Kunci Q yang diambil hanya sepanjang plainteks.

2. Enkripsi

   Pada proses enkripsi, masing-masing karakter dari plainteks diubah kebentuk numerik berdasarkan tabel 3.1. Setelah itu proses enkripsi dilakukan menggunakan persamaan (2.4) dengan kunci yang didapatkan dari tahap (1) yakni kunci K untuk menghasilkan cipherteks dengan $32 \leq C \leq 126$ (sesuai dengan tabel 3.1). Cipherteks hasil enkripsi tersebut kemudian diubah kembali kedalam bentuk karakter berdasarkan tabel 3.1.

3. Dekripsi

   Pada tahap yang ketiga, cipherteks yang telah didapatkan dari proses enkripsi pada tahap sebelumnya, diubah kebentuk numerik berdasarkan tabel 3.1. Kemudian dilakukan proses dekripsi menggunakan kunci yang sama dengan proses enkripsi yaitu kunci K dan persamaan (2.5). Sama halnya dengan proses enkripsi, nilai plainteks yang didapatkan harus dalam jangkauan $32 \leq P \leq 126$ sesuai dengan tabel 3.1. Hal ini bertujuan agar plainteks dapat kembali seperti semula.

c. Pembuatan Program

   Pada langkah yang ketiga, terlebih dahulu dibuat algoritma pembangkitan kunci, enkripsi, dan dekripsi dengan metode *Vigenere* dengan kunci barisan karakter. Setelah itu akan dibuat program dengan menggunakan *software* MATLAB 2009a berdasarkan algoritma yang telah dibuat sebelumnya.

d. Simulasi Program

   Setelah program selesai dibuat, langkah selanjutnya adalah simulasi program tersebut. Program diuji dengan menggunakan teks pesan dengan berbagai macam karakter.

e. Analisis Hasil Simulasi

Langkah terakhir adalah menganalisis hasil dari simulasi program. Analisis dilakukan dengan membandingkan karakter sebelum dan sesudah proses enkripsi.

BAB 4. HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai algoritma, contoh perhitungan manual pengkodean *Vigenere* dengan kunci barisan karakter serta hasil dari program menggunakan *software* MATLAB 2009a.

4.1 Algoritma Pengkodean *Vigenere* dengan Kunci Barisan Karakter

4.1.1 Proses Pembentukan Kunci

Langkah-langkah pembentukan kunci K apabila diketahui plainteks P dan kunci Q adalah sebagai berikut:

1. bangkitkan kunci Q, panjang Q adalah n ($n \geq 1$)
2. menghitung panjang plainteks P, dengan panjang P adalah p
3. jika $n < p$, maka ada perulangan Q sepanjang P
4. jika $n > p$, maka karakter dari Q yang digunakan hanya sepanjang P
5. konversikan karakter kunci Q kebentuk numerik sesuai dengan tabel 3.1, sehingga $Q_i=q_i$ dimana q_i adalah representasi numerik dari Kunci Q ke- i
6. cari nilai kunci K, dengan ketentuan:

$$K_i = \begin{cases} 2q_i - 1, & \text{untuk } i = \text{prima} \\ 3q_i + 5, & \text{untuk } i = \text{ganjil tidak prima} \\ 2q_i + 2, & \text{untuk } i = \text{genap tidak prima} \end{cases}$$

K_i adalah kunci K ke-i.

4.1.2 Proses Enkripsi

Pada proses ini, plainteks yang telah ditentukan diubah menjadi cipherteks dengan menggunakan kunci. Algoritma dari proses enkripsi adalah sebagai berikut:

Input : Pesan yang akan dienkripsi (Plainteks)

Output : Pesan yang dikodekan (Cipherteks)

Langkah-langkah:

1. konversi karakter plainteks *P* kebentuk numerik sesuai tabel 3.1
2. menentukan kunci yang bersesuaian dengan plainteks sesuai algoritma 4.1.1
3. proses enkripsi dengan menggunakan persamaan $C_i = (P_i + K_i)$ **mod** 95, dimana P_i adalah representasi numerik dari plainteks ke-*i* dan $32 \leq C \leq 126$
4. konversi C_i kebentuk karakter, dimana C_i merupakan karakter dari cipherteks.
5. diperoleh cipherteks.

Tabel 4.1 Enkripsi

P:	P_1	P_2	P_3	...	P_p
K:	K_1	K_2	K_3	...	K_p
C:	(P_1+K_1) **mod** 95	(P_2+K_2) **mod** 95	(P_3+K_3) **mod** 95	...	(P_p+K_p) **mod** 95

4.1.3 Proses Dekripsi

Pada proses ini, cipherteks yang didapatkan dari proses enkripsi diubah menjadi plainteks menggunakan kunci yang sama dengan kunci pada proses enkripsi. Algoritma dari proses dekripsi adalah sebagai berikut:

Input : Cipherteks

Output : Plainteks

Langkah-langkah:

1. konversi karakter cipherteks C_i kebentuk numerik sesuai tabel 3.1
2. menentukan kunci yang bersesuaian dengan chiperteks sesuai algoritma 4.1.1
3. proses dekripsi dengan menggunakan persamaan $P_i = (C_i - K_i)$ **mod** 95, jika $C_i < K_i$ maka $C_i = C_i + 95$ dimana $32 \leq P \leq 126$
4. konversi P_i kebentuk karakter
5. diperoleh plainteks.

Tabel 4.2 Dekripsi

C:	C_1	C_2	C_3	...	C_p
K:	K_1	K_2	K_3	...	K_p
P:	(C_1-K_1) **mod** 95	(C_2-K_2) **mod** 95	(C_3-K_3) **mod** 95	...	(C_p-K_p) **mod**95

4.2. Pengkodean *Vigenere* dengan Kunci Barisan Karakter

Sebelum penyusunan program terlebih dahulu didemonstrasikan secara manual penyelesaian pengkodean *Vigenere* dengan kunci barisan karakter melalui 3 buah contoh. Contoh pertama menggunakan Kunci *Q* yang panjangnya kurang dari panjang plainteks, contoh kedua menggunakan Kunci *Q* yang panjangnya sama dengan panjang plainteks, dan contoh ketiga menggunakan Kunci *Q* yang panjangnya lebih dari panjang plainteks.

4.2.1 Pengkodean *Vigenere* dengan Panjang Kunci *Q* Kurang dari Panjang Plainteks

Contoh

Plainteks : AAAAAAAAAAA

Kunci *Q* : YULAN

a. Proses pembentukan kunci

Berdasarkan plainteks dan Kunci *Q* yang telah ditentukan diatas, maka:

p = 11

n = 5

Karena $n < p$, maka Kunci *Q* diulang secara periodik sepanjang plainteks *P*. Perulangan tersebut dapat dilihat pada tabel 4.3.

Tabel 4.3 Perulangan Kunci *Q* terhadap Plainteks dengan $n < p$

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}
A	A	A	A	A	A	A	A	A	A	A
Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_{11}
Y	U	L	A	N	Y	U	L	A	N	Y

Untuk mendapatkan Kunci *K*, masing-masing karakter dari Kunci *Q* pada tabel 4.3 diubah kebentuk numerik berdasarkan tabel 3.1 Hasilnya dapat dilihat pada tabel 4.4.

Tabel 4.4 Konversi Karakter Kunci *Q* kebentuk numerik dengan $n < p$

Karakter:	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_{11}
	Y	U	L	A	N	Y	U	L	A	N	Y
Numerik:	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}
	89	85	76	65	78	89	85	76	65	78	89

Dengan melihat indeks numerik pada variabel *q* dan menggunakan persamaan:

$$K_i = \begin{cases} 2q_i - 1, \text{ untuk } i = \text{prima} \\ 3q_i + 5, \text{ untuk } i = \text{ganjil tidak prima} \\ 2q_i + 2, \text{ untuk } i = \text{genap tidak prima} \end{cases}$$

maka kunci *K* diperoleh dengan cara:

- Untuk K_1, $i = 1$ dan i = ganjil

  $K_1 = 3q_1 + 5$

  $K_1 = 3(89) + 5$

  $K_1 = 272$

- Untuk K_2, $i = 2$ dan i = prima

  $K_2 = 2q_2 - 1$

  $K_2 = 2(85) - 1$

  $K_2 = 169$

  .
  .
  .

- Untuk K_{11}, $i = 11$ dan i = prima

  $K_{11} = 2q_{11} - 1$

  $K_{11} = 2(89) - 1$

  $K_{11} = 177$

Keseluruhan nilai kunci *K* yang dihasilkan dari perhitungan tersebut dapat dilihat pada tabel 4.5.

Tabel 4.5 Kunci *K* dengan $n < p$

K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}
272	169	151	132	155	180	169	154	200	158	177

Kunci K tersebut digunakan untuk mengubah plainteks menjadi cipherteks melalui persamaan (2.4), dan juga mengubah cipherteks kembali menjadi plainteks melalui persamaan (2.5).

b. Proses Enkripsi

Pada proses enkripsi, plainteks yang telah ditentukan sebelumnya yaitu AAAAAAAAAAA, diubah menjadi cipherteks menggunakan kunci K yang ada pada tabel 4.5. Terlebih dahulu dikonversikan tiap karakter dari plainteks pada tabel 4.6 kebentuk numerik sesuai dengan tabel 3.1.

Tabel 4.6 Karakter Plainteks dengan $n < p$

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}
A	A	A	A	A	A	A	A	A	A	A

Dengan menggunakan persamaan (2.4) dan kunci K pada tabel 4.5, maka cipherteks yang dihasilkan dapat dilihat pada tabel 4.7.

Tabel 4.7 Hasil Enkripsi dengan $n < p$

P :	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}
	65	65	65	65	65	65	65	65	65	65	65
Kunci K:	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}
	272	169	151	132	155	180	169	154	200	158	177
C :	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
	52	44	121	102	125	55	44	124	75	33	52

Perhatikan tabel 4.7 diatas, misalkan pada P_1, untuk mendapatkan nilai C_1 adalah dengan cara menjumlahkan nilai P_1 yaitu 65 dengan nilai kunci K_1 kemudian hasilnya dimodulo 95 sehingga menghasilkan nilai $C_1 = 52$. Cara tersebut berlaku pula untuk nilai-nilai cipherteks yang lain. Seluruh nilai C yang didapat dari tabel 4.7 selanjutnya diubah kembali kebentuk karakter berdasarkan tabel 3.1. Hasil keseluruhan cipherteks dapat dilihat pada tabel 4.8.

Tabel 4.8 Karakter Cipherteks dengan $n < p$

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
4	,	y	f	}	7	,	|	K	!	4

c. Proses Dekripsi

Pada proses ini, karakter cipherteks pada tabel 4.8 yang didapatkan dari proses enkripsi diubah kembali ke plainteks menggunakan kunci K yang ada pada tabel 4.5 agar pesan dapat kembali dimengerti. Caranya adalah mengubah karakter cipherteks kebentuk numerik berdasarkan tabel 3.1. Kemudian dengan menggunakan persamaan (2.5) dan kunci K yang terdapat pada tabel 4.5 akan menghasilkan plainteks yang dapat dilihat pada tabel 4.9.

Tabel 4.9 Hasil Dekripsi dengan $n < p$

C :	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
	52	44	121	102	125	55	44	124	75	33	52
Kunci K:	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}
	272	169	151	132	155	180	169	154	200	158	177
P :	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}
	65	65	65	65	65	65	65	65	65	65	65

Pada tabel 4.9, tinjau nilai P_1 , nilai tersebut didapatkan dengan cara nilai dari C_1 dikurangi nilai K_1 yaitu 272 dan hasilnya dimodulo 95. Karena $52 < 272$, maka 52 ditambah dengan 95 hingga nilai $C_1 > K_1$ sehingga nilai P_1 yang didapatkan adalah 65. Cara tersebut berlaku untuk mencari nilai P yang lain. Bila diperhatikan, nilai Plainteks yang dihasilkan pada tabel 4.9 adalah sama. Apabila nilai plainteks tersebut diubah kebentuk karakter akan menghasilkan plainteks AAAAAAAAAAA. Hal ini menunjukkan bahwa cipherteks dapat kembali menjadi plainteks semula.

4.2.2 Pengkodean *Vigenere* dengan Panjang Kunci *Q* Sama dengan Panjang Plainteks

Contoh

Plainteks : AAAAAAAAA

Kunci *Q* : YULAN ISA

a. Proses pembentukan kunci

Berdasarkan plainteks dan kunci *Q* yang telah ditentukan tersebut, maka:

$p = 9$

$n = 9$

karena $n = p$, maka kunci *Q* tidak perlu diulang secara periodik sepanjang plainteks *P*. Peletakan kunci *Q* terhadap plainteks dapat dilihat pada tabel dibawah ini.

Tabel 4.10 Peletakan kunci *Q* terhadap Plainteks dengan $n = p$

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
A	A	A	A	A	A	A	A	A
Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
Y	U	L	A	N	space	I	S	A

Hasil konversi karakter Q_i pada tabel 4.10 kebentuk numerik dapat dilihat pada tabel 4.11.

Tabel 4.11 Konversi karakter kunci *Q* kebentuk numerik dengan $n = p$

Karakter:	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
	Y	U	L	A	N	space	I	S	A
Numerik:	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9
	89	85	76	65	78	32	73	83	65

Dengan melihat indeks numerik pada variabel q dan menggunakan persamaan:

$$K_i = \begin{cases} 2q_i - 1, \text{ untuk } i = \text{prima} \\ 3q_i + 5, \text{ untuk } i = \text{ganjil tidak prima} \\ 2q_i + 2, \text{ untuk } i = \text{genap tidak prima} \end{cases}$$

maka kunci K dapat diperoleh dengan cara:

- Untuk K_1, $i = 1$ dan i = ganjil

  $K_1 = 3q_1 + 5$

  $K_1 = 3(89) + 5$

  $K_1 = 272$

- Untuk K_2, $i = 2$ dan i = prima

  $K_2 = 2q_2 - 1$

  $K_2 = 2(85) - 1$

  $K_2 = 169$

  .
  .
  .

- Untuk K_9, $i = 9$ dan i = ganjil

  $K_9 = 3q_1 + 5$

  $K_9 = 3(65) + 5$

  $K_9 = 200$

Keseluruhan nilai kunci K yang dihasilkan dari perhitungan tersebut dapat dilihat pada tabel 4.12.

Tabel 4.12 Kunci K dengan $n = p$

K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9
272	169	151	132	155	66	145	168	200

Kunci K pada tabel 4.12 tersebut digunakan untuk mengubah plainteks menjadi cipherteks melalui persamaan (2.4), dan juga mengubah cipherteks kembali menjadi plainteks melalui persamaan (2.5).

b. Proses Enkripsi

Pada proses enkripsi, plainteks AAAAAAAAA diubah menjadi cipherteks menggunakan kunci *K* yang dapat dilihat pada tabel 4.12. Terlebih dahulu dikonversikan tiap karakter dari plainteks pada tabel 4.13 kebentuk numerik sesuai tabel 3.1.

Tabel 4.13 Karakter Plainteks dengan $n = p$

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
A	A	A	A	A	A	A	A	A

Dengan menggunakan persamaan (2.4) dan kunci *K* maka cipherteks yang dihasilkan dapat dilihat pada tabel 4.14.

Tabel 4.14 Hasil Enkripsi dengan $n = p$

P :	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
	65	65	65	65	65	65	65	65	65
Kunci *K*:	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9
	272	169	151	132	155	66	145	168	200
C :	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
	52	44	121	102	125	36	115	43	75

Nilai *C* yang didapat dari tabel 4.14 selanjutnya diubah kembali kebentuk karakter. Hasil konversi keseluruhan bilangan dapat dilihat pada tabel 4.15.

Tabel 4.15 Karakter Cipherteks dengan $n = p$

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
4	,	y	f	}	$	s	+	K

c. Proses Dekripsi

Pada proses ini, cipherteks pada tabel 4.15 yang sudah didapatkan dari proses enkripsi diubah kembali ke plainteks menggunakan kunci *K* agar pesan

dapat kembali dimengerti. Karakter cipherteks diubah kebentuk numerik sesuai tabel 3.1. Dengan menggunakan persamaan (2.5) dan kunci *K* maka plainteks yang dihasilkan dapat dilihat pada tabel 4.16 dibawah ini.

Tabel 4.16 Hasil Dekripsi dengan $n = p$

***C* :**	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
	52	44	121	102	125	36	115	43	75
Kunci *K*:	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9
	272	169	151	132	155	66	145	168	200
***P* :**	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
	65	65	65	65	65	65	65	65	65

Pada tabel 4.16 dapat dilihat bahwa nilai-nlai plainteks tersebut apabila dikonversikan kedalam bentuk karakter berdasarkan tabel 3.1 akan menjadi AAAAAAAAA sama seperti plainteks semula.

4.2.3 Pengkodean *Vigenere* dengan Panjang Kunci *Q* Lebih dari Panjang Plainteks

Misalkan:

Plainteks : AAAAA

Kunci *Q* : YULAN ISA

a. Proses pembentukan kunci

Berdasarkan plainteks dan Kunci *Q* yang telah ditentukan diatas, maka:

$p = 5$

$n = 9$

Karena $n > p$, maka Kunci *Q* yang digunakan hanya sepanjang plainteks *P*. Peletakan kunci *Q* terhadap plainteks dapat dilihat pada tabel 4.17.

Tabel 4.17 Peletakan Kunci *Q* terhadap Plainteks dengan $n > p$

P_1	P_2	P_3	P_4	P_5
A	A	A	A	A
Q_1	Q_2	Q_3	Q_4	Q_5
Y	U	L	A	N

Untuk mendapatkan kunci *K* yang digunakan pada proses enkripsi dan dekripsi, masing-masing karakter *Q* pada tabel 4.17 diubah kebentuk numerik berdasarkan tabel 3.1 Hasil perubahan tersebut dapat dilihat pada tabel 4.18.

Tabel 4.18 Konversi Karakter Kunci *Q* ke Bilangan dengan $n > p$

Karakter:	Q_1	Q_2	Q_3	Q_4	Q_5
	Y	U	L	A	N
Numerik:	q_1	q_2	q_3	q_4	q_5
	89	85	76	65	78

Dengan melihat indeks numerik pada variabel *q* dan menggunakan persamaan:

$$K_i = \begin{cases} 2q_i - 1, \text{ untuk } i = \text{prima} \\ 3q_i + 5, \text{ untuk } i = \text{ganjil tidak prima} \\ 2q_i + 2, \text{ untuk } i = \text{genap tidak prima} \end{cases}$$

maka kunci *K* diperoleh dengan cara:

- Untuk K_1, $i = 1$ dan i = ganjil

  $K_1 = 3q_1 + 5$

  $K_1 = 3(89) + 5$

  $K_1 = 272$
- Untuk K_2, $i = 2$ dan i = prima

  $K_2 = 2q_2 - 1$

  $K_2 = 2(85) - 1$

  $K_2 = 169$

  .

.
.

- Untuk K_5, $i = 5$ dan i = prima

  $K_5 = 2q_5 - 1$

  $K_5 = 2(78) - 1$

  $K_5 = 155$

Keseluruhan nilai kunci K yang dihasilkan dari perhitungan tersebut dapat dilihat pada tabel 4.19.

Tabel 4.19 Kunci K dengan $n > p$

K_1	K_2	K_3	K_4	K_5
272	169	151	132	155

Kunci K pada tabel diatas digunakan sebagai kunci pada proses mengubah plainteks menjadi cipherteks, dan juga sebaliknya.

b. Proses Enkripsi

Plainteks yang telah ditentukan di awal yaitu AAAAA diubah menjadi cipherteks menggunakan kunci K (lihat tabel 4.19). Terlebih dahulu dikonversikan tiap karakter plainteks pada tabel 4.20 kebentuk numerik sesuai dengan tabel 3.1.

Tabel 4.20 Karakter Plainteks dengan $n > p$

P_1	P_2	P_3	P_4	P_5
A	A	A	A	A

Dengan menggunakan persamaan (2.4) dan kunci K maka cipherteks yang dihasilkan dapat dilihat pada tabel 4.21.

Tabel 4.21 Hasil Enkripsi dengan $n > p$

$\boldsymbol{P}$ **:**	$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$	$\boldsymbol{P_4}$	$\boldsymbol{P_5}$
	65	65	65	65	65
Kunci $\boldsymbol{K}$**:**	$\boldsymbol{K_1}$	$\boldsymbol{K_2}$	$\boldsymbol{K_3}$	$\boldsymbol{K_4}$	$\boldsymbol{K_5}$
	272	169	151	132	155
$\boldsymbol{C}$ **:**	$\boldsymbol{C_1}$	$\boldsymbol{C_2}$	$\boldsymbol{C_3}$	$\boldsymbol{C_4}$	$\boldsymbol{C_5}$
	52	44	121	102	125

Nilai C pada tabel 4.21 selanjutnya diubah kebentuk karakter berdasarkan tabel 3.1 dan hasilnya dapat dilihat pada tabel 4.22.

Tabel 4.22 Karakter cipherteks dengan $n > p$

$\boldsymbol{C_1}$	$\boldsymbol{C_2}$	$\boldsymbol{C_3}$	$\boldsymbol{C_4}$	$\boldsymbol{C_5}$
4	,	y	f	}

c. Proses Dekripsi

Pada proses ini, cipherteks pada tabel 4.22 diubah kembali ke plainteks menggunakan kunci K yang ada pada tabel 4.19 agar pesan dapat kembali dimengerti. Karakter cipherteks yang sebelumnya sudah diperoleh yakni 4,yf} diubah kebentuk numerik sesuai tabel 3.1. Dengan menggunakan persamaan (2.5) dan kunci K maka plainteks yang dihasilkan dapat dilihat pada tabel 4.23.

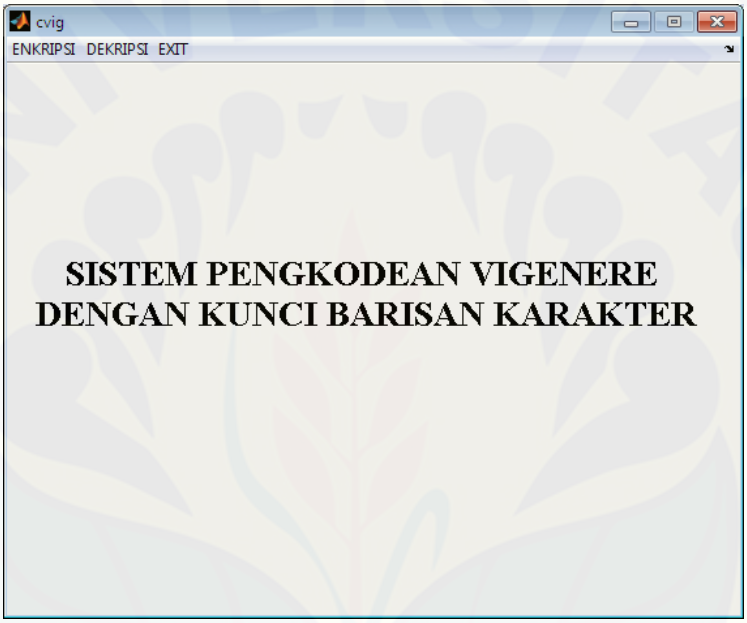
Tabel 4.23 Hasil Dekripsi dengan $n > p$

$\boldsymbol{C}$ **:**	$\boldsymbol{C_1}$	$\boldsymbol{C_2}$	$\boldsymbol{C_3}$	$\boldsymbol{C_4}$	$\boldsymbol{C_5}$
	52	44	121	102	125
Kunci $\boldsymbol{K}$**:**	$\boldsymbol{K_1}$	$\boldsymbol{K_2}$	$\boldsymbol{K_3}$	$\boldsymbol{K_4}$	$\boldsymbol{K_5}$
	272	169	151	132	155
$\boldsymbol{P}$ **:**	$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$	$\boldsymbol{P_4}$	$\boldsymbol{P_5}$
	65	65	65	65	65

Pada tabel 4.23 dapat dilihat bahwa nilai-nlai plainteks yang dihasilkan sama seperti nilai-nilai dari plainteks semula.

4.3 Implementasi program

Berikut ini merupakan paparan implementasi algoritma *Vigenere* dengan kunci barisan karakter dalam program yang telah dibangun menggunakan *software* MATLAB 2009a. Tampilan awal program dapat dilihat pada gambar 4.1 berikut.

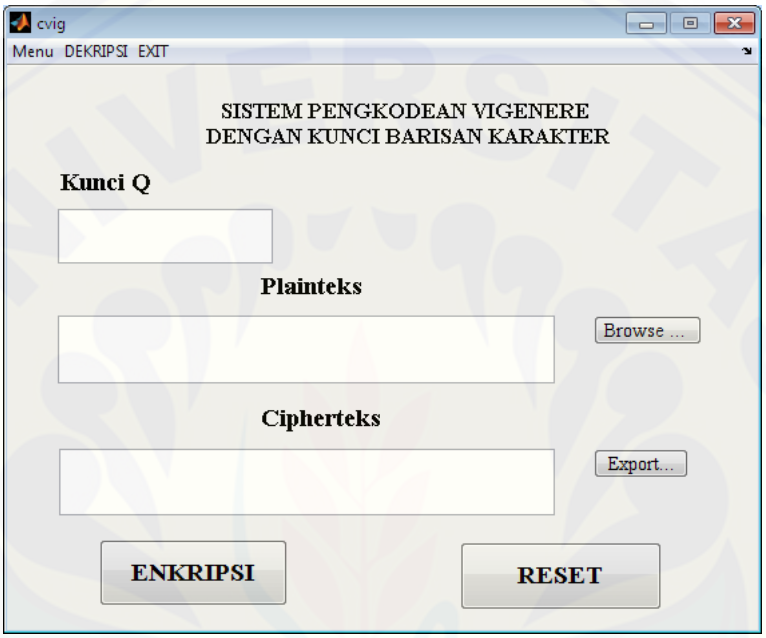


Gambar 4.1 Tampilan awal program

Pada tampilan awal program diatas terdapat tiga menu pilihan yaitu menu ENKRIPSI, DEKRIPSI, dan EXIT. Menu ENKRIPSI digunakan untuk melakukan proses enkripsi, menu DEKRIPSI digunakan untuk melakukan proses dekripsi, sedangkan menu EXIT digunakan untuk keluar dari program.

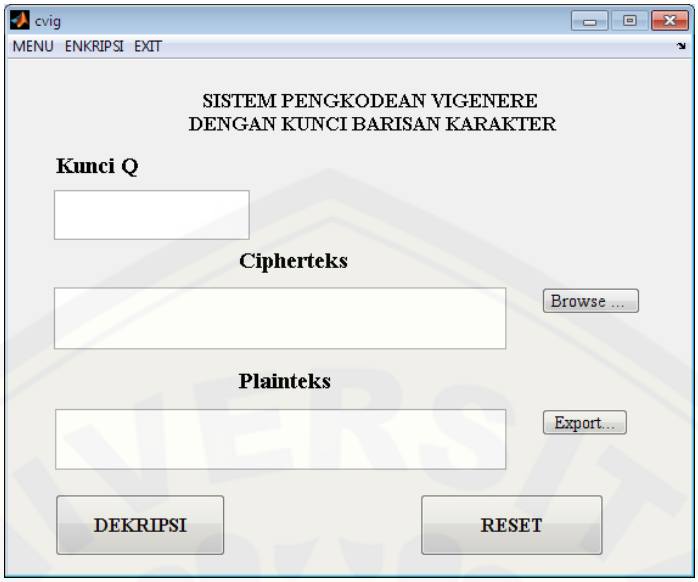
Untuk tampilan dari menu ENKRIPSI dapat dilihat pada gambar 4.2. Pada menu ENKRIPSI terdiri dari kolom kunci, plainteks, dan cipherteks. Selain itu juga terdapat 4 tombol yakni *Browse*, *Export*, Enkripsi, dan Reset. Pada kolom kunci dan plainteks, keduanya merupakan kolom *input*, sedangkan pada kolom cipherteks merupakan kolom *output* hasil proses enkripsi.

Tombol *Browse* berfungsi sebagai perintah untuk menginput file yang akan dienkripsi, sedangkan tombol *Export* berfungsi sebagai perintah untuk menyimpan *output* file hasil dari proses enkripsi. Tombol reset digunakan untuk menghapus isi dari kolom yang ada. Sedangkan tombol Enkripsi sendiri berfungsi sebagai perintah untuk mengaktifkan proses enkripsi.



Gambar 4.2 Tampilan Menu Enkripsi Sistem Pengkodean *Vigenere* dengan Kunci Barisan Karakter

Untuk tampilan menu DEKRIPSI dapat dilihat pada gambar 4.3. Pada gambar tersebut terdapat kolom-kolom dan juga tombol-tombol yang tidak berbeda jauh dengan tampilan menu Enkripsi. Yang membedakan hanya pada letak antara kolom plainteks dan cipherteks yang saling bertukar. Jika pada menu Enkripsi, kolom plainteks merupakan *input*, pada menu Dekripsi kolom plainteks merupakan kolom *output*. Selain itu, pada tampilan menu dekripsi ini terdapat tombol Dekripsi yang digunakan sebagai perintah untuk melakukan proses dekripsi.



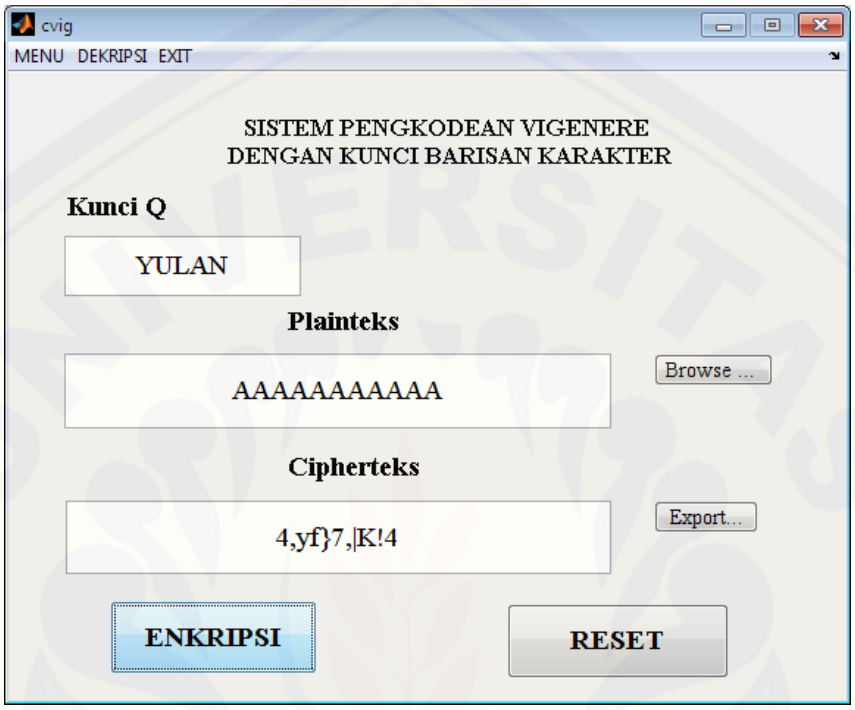
Gambar 4.3 Tampilan menu Dekripsi sistem pengkodean *Vigenere* dengan kunci barisan karakter

Langkah-langkah penggunaan program secara umum adalah sebagai berikut:

1. Pilih menu ENKRIPSI untuk melakukan proses enkripsi
2. Masukkan karakter kunci yang diinginkan pada kolom kunci.
3. Masukkan plainteks pada kolom plainteks lalu pilih tombol ENKRIPSI sehingga akan muncul hasil enkripsi pada kolom cipherteks.
4. Untuk mendekripsi cipherteks, pilih menu DEKRIPSI lalu masukkan kunci yang sama dengan langkah 2. Selanjutnya masukkan cipherteks hasil dari langkah 3 lalu pilih tombol DEKRIPSI untuk menghasilkan plainteks pada kolom plainteks.

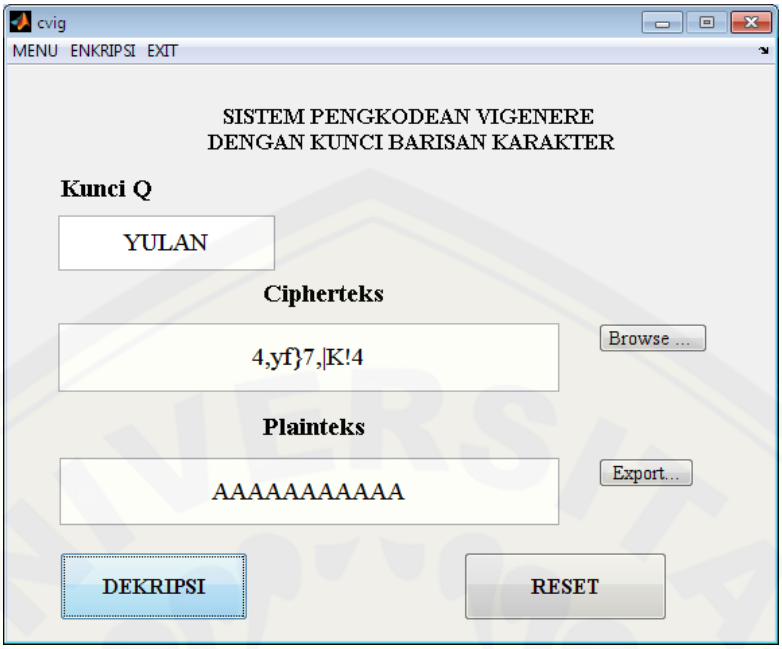
Berdasarkan salah satu contoh yang telah dibuat pada subbab 4.2, dapat dilihat bahwa pada proses enkripsi pengkodean *Vigenere* dengan kunci barisan karakter terdapat sebuah plainteks AAAAAAAAAAA, kunci *Q* YULAN untuk membangkitkan kunci *K* 272 169 151 132 155 180 169 154 200 158 177 sehingga menghasilkan cipherteks 4,yf}7,|K!4. Untuk mencari tahu keberhasilan dari program yang telah dibuat dengan menggunakan *software* MATLAB 2009a, penulis mencoba melakukan pengujian dengan melakukan proses enkripsi

menggunakan data yang sama persis dengan data pada contoh perhitungan manual tersebut. Berikut adalah tampilan dari proses enkripsi dengan menggunakan plainteks AAAAAAAAAAA dan kunci *Q* YULAN.



Gambar 4.4 Tampilan proses enkripsi

Proses enkripsi pada gambar 4.4 menunjukkan bahwa antara hasil enkripsi perhitungan manual dan hasil enkripsi program menghasilkan cipherteks yang sama. Sedangkan untuk tampilan proses dekripsi pesan dapat dilihat pada gambar 4.5.



Gambar 4.5 Tampilan proses dekripsi

Pada gambar 4.5 dapat dilihat bahwa cipherteks yang didekripsi dapat dikembalikan menjadi plainteks yang sama seperti semula. Dengan kata lain, pesan yang telah dikodekan dapat dikembalikan menjadi pesan yang dapat dimengerti. Hal ini membuktikan bahwa program pengkodean *Vigenere* dengan kunci barisan karakter dapat berjalan dengan baik.

4.4 Analisis Hasil Sistem Pengkodean *Vigenere* dengan Kunci Barisan Karakter

Secara umum, pengkodean *Vigenere* adalah sebuah pengkodean pesan yang memiliki tujuan untuk menyembunyikan keterhubungan antara plainteks dan cipherteks dengan menggunakan kunci sebagai penentu pergeseran karakternya. Namun kelemahan dari metode ini adalah jika panjang kunci yang digunakan lebih pendek dari plainteks akan mengakibatkan kunci diulang secara periodik sepanjang plainteks. Hal tersebut memungkinkan cipherteks yang dihasilkan membentuk sebuah pola tertentu sehingga mengakibatkan cipherteks mudah dipecahkan oleh pihak yang tidak berhak dengan cara menganalisis frekuensi dari pola cipherteksnya. Dengan melihat contoh cipherteks yang dihasilkan dari proses

enkripsi menggunakan pengkodean *Vigenere* dengan kunci barisan karakter pada gambar 4.4, dapat diperhatikan bahwa karakter cipherteks yang dihasilkan terlihat acak dan tidak membentuk suatu pola tertentu. Perbedaan antara cipherteks yang dihasilkan dengan plainteks pada sistem pengkodean *Vigenere* dengan kunci barisan karakter berdasarkan contoh perhitungan manual dapat dilihat lebih jelas pada tabel 4.24 dibawah ini.

Tabel 4.24 Perbandingan karakter plainteks dan cipherteks

Panjang Kunci *Q*	Kunci *Q*	Plainteks	Cipherteks
$n < p$	YULAN	AAAAAAAAAAA	4,yf}7,|K!4
$n = p$	YULAN ISA	AAAAAAAAA	4,yf}$s+K
$n > p$	YULAN ISA PUSPITA	AAAAA	4,yf}

Dari tabel diatas terlihat jelas bahwa hasil enkripsi dari metode *Vigenere* ini tidak menghasilkan cipherteks yang membentuk pola berulang meskipun pada plainteks terdiri dari karakter yang sama yaitu karakter A. Hal ini menunjukkan bahwa sistem pengkodean *Vigenere* berhasil meminimalisir kemungkinan terbentuknya suatu pola pada cipherteksnya sehingga mengakibatkan kriptanalis lebih kesulitan dalam memecahkan kunci. Hal ini disebabkan karena analisis frekuensi menjadi lebih sulit dilakukan pada cipherteks yang dihasilkan oleh metode ini.

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan mengenai sistem pengkodean *Vigenere* dengan kunci barisan karakter dapat disimpulkan bahwa :

1. Enkripsi menggunakan pengkodean *Vigenere* dengan kunci barisan karakter dan plainteks AAAAAAAAAAA menghasilkan cipherteks 4,yf}7,|K!4. Ketika plainteks diinputkan kedalam program juga menghasilkan cipherteks yang sama.
2. Enkripsi menggunakan pengkodean *Vigenere* dengan kunci barisan karakter dapat mengamankan pesan atau informasi karena metode ini berhasil menyembunyikan keterhubungan antara plainteks dengan cipherteks melalui sebuah kunci yang digunakan sehingga dapat lebih menyulitkan pihak yang tidak berhak atas informasi dalam memecahkan kunci.

5.2 Saran

Penelitian mengenai kriptografi masih terbuka bagi peneliti lain karena pada kriptografi terdapat bermacam-macam algoritma yang dapat digunakan selain metode *Vigenere*. Disamping itu, untuk lebih menyulitkan kriptoanalis dalam menebak kunci maka peneliti lain dapat menggunakan modifikasi kunci lain yang lebih sulit pada metode *Vigenere*.

DAFTAR PUSTAKA

Ariesanda, B. 2009. Rancangan dan Analisis Cipher Berbasis Algoritma Transposisi dengan Periodisasi Kunci. http://www.informatika.org/~rinaldi/Kriptografi/2006-2007/Makalah1/Makalah1-043.pdf [26 Februari 2013]

Asmara, F.D. 2009. *Penyandian Hill menggunakan Kode Plainteks Kembar*. Skripsi. Tidak diterbitkan. Jember: Jurusan Matematika. Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Jember.

Kromodimoeljo, S. 2010. *Teori dan Aplikasi Kriptografi*. SPK IT *Consulting*.

Menezes, Alfred Paul Van Oorschot and Vaston Sean. 1996. *Handbook of Applied Cryptography*. USA. CRC Press Inc.

Munir R.2006.*Kriptografi*. Bandung:Institut Teknologi Bandung.

Munir R.2012.*Matematika Diskrit*. Bandung:Informatika

Retnawati, H. 2008. *Kreatif Menggunakan Matematika*. Pusat Perbukuan, Kementrian Pendidikan Nasional.

Rosidah. 2009. *Sistem Pengkodean Playfair-Vigenere*. Skripsi. Tidak diterbitkan. Jember: Jurusan Matematika. Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Jember.

Schneier, Bruce. 1996. *Aplied Cryptography 2nd*. New York: John Wiley & Sons.