PENGAMANAN *POLYALPHABETIC* DENGAN *AFFINE CIPHER* BERDASARKAN BARISAN FIBONACCI

SKRIPSI

Oleh
Lestari Fidi Astuti
161810101016

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2020

PENGAMANAN *POLYALPHABETIC* DENGAN *AFFINE CIPHER* BERDASARKAN BARISAN FIBONACCI

SKRIPSI

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

Oleh
Lestari Fidi Astuti
161810101016

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2020

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Kedua orang tuaku tersayang, Bapak Halil dan Ibu Asiati yang senantiasa mendoakan dan memberikan dukungan baik secara moral ataupun materi.
2. Kakakku Feri Agung Teguh Prakoso, adik-adikku Moh. Farid Nur Hidayat dan Diayu Puji Astuti serta keluarga besar yang selalu memberikan dukungan dan Doa.
3. Guru-guru dari RA Raisul Anwar, MI Raisul Anwar, MTSN Paiton, MAN Paiton, dan dosen FMIPA UNEJ yang telah memberikan ilmu serta bimbingan dengan penuh kesabaran.
4. Almamater tercinta Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.
5. Keluarga besar Matematika angkatan 2016 (MISDIRECTION'16) serta sahabat-sahabatku yang telah membantu dan memberikan pengalaman hidup yang luar biasa.
6. Semua pihak yang telah membantu dalam menyelesaikan skripsi ini.

MOTO

“Barang siapa yang bersungguh-sungguh, sesungguhnya kesungguhan tersebut untuk kebaikan dirinya sendiri”
(Terjemahan QS. Al – Ankabut: 6)

“Dan janganlah berputus asa dari rahmat Allah. Sesungguhnya tiada berputus dari rahmat Allah melainkan kaum yang kafir”
(Terjemahan QS. Yusuf: 87)

Departemen Agama Republik Indonesia. 2010. Al – Qur’an dan Maknanya.

PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Lestari Fidi Astuti

NIM : 161810101016

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Pengamanan *Polyalphabetic* dengan *Affine Cipher* Berdasarkan Barisan Fibonacci" adalah benar-benar hasil karya sendiri, kecuali kutipan yang telah disebutkan sumbernya, belum pernah diajukan di institusi manapun, dan 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 ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, Januari 2020

Yang menyatakan,

Lestari Fidi Astuti

NIM 161810101016

SKRIPSI

**PENGAMANAN*POLYALPHABETIC*DENGAN*AFFINE CIPHER*
BERDASARKAN BARISAN FIBONACCI**

Oleh

Lestari Fidi Astuti
161810101016

Pembimbing

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

Dosen Pembimbing Anggota : Ahmad Kamsyakawuni, S.Si., M.Kom

PENGESAHAN

Skripsi berjudul "Pengamanan *Polyalphabetic* dengan *Affine Cipher* Berdasarkan Barisan Fibonacci" telah diuji dan disahkan pada:

Hari, tanggal :

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Ketua,	Anggota I
Dr. Kiswara Agung Santoso, S.Si., M.Kom.	Ahmad Kamsyakawuni, S.Si., M.Kom.
NIP. 197209071998031003	NIP. 197211291998021001
Anggota II,	Anggota III
Ikhsanul Halikin, S.Pd., M.Si.	Kusbudiono, S.Si., M.Si.
NIP. 198610142014041001	NIP. 197704302005011001

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

Drs. Achmad Sjaifullah, M.Sc., Ph.D.

NIP. 195910091986021001

RINGKASAN

Pengamanan *Polyalphabetic* dengan *Affine Cipher* Berdasarkan Barisan Fibonacci; Lestari Fidi Astuti, 161810101016; 2019: 67 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Perkembangan teknologi yang semakin maju memudahkan kehidupan manusia salah satunya dalam bidang komunikasi. Dalam komunikasi sangat penting untuk menjaga kerahasiaannya, oleh sebab itu dibutuhkan ilmu kriptografi untuk menjaga pesan tersebut tetap aman dan tidak diketahui oleh pihak yang tidak berhak. Dalam kriptografi terdapat istilah *plaintext* dan *ciphertext*. *Plaintext* merupakan pesan asli, sedangkan *ciphertext*merupakan pesan yang sudah dikodekan. Proses mengubah *plaintext* menjadi *ciphertext* disebut enkripsi, sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* disebut dekripsi.Terdapat dua macam kriptografi, yaitu kriptografi klasik dan kriptografi modern.

Affine cipher merupakan salah satu contoh algoritma kriptografi klasik. *Affine cipher* dalam proses enkripsi dan dekripsi menggunakan dua buah kunci yaitu, kunci *a*berupa bilangan yang relatif prima sebagai pengali dan kunci *b*berupa bilangan bulat sebagai penggeser. Penelitian ini bertujuan untuk menjaga keamanan pengiriman pesan teks menggunakan *affine cipher* dengan modifikasi kunci berdasarkan barisan Fibonacci. Peneliti melakukan modifikasi terhadap kunci *b*, dimana kunci *b* ini dimodifikasi dengan aturan barisan Fibonacci. Pada saat proses enkripsi*plaintext* dan dua buah kunci diinputkan. Dua buah kunci yang diinputkan yaitu, kunci *a*berupa bilangan yang relatif prima dengan 95 dan kunci *b* berupa teks. Kunci *b* ini harus melalui proses pembentukan kunci dengan cara dikonversi ke desimal menggunakan tabel ASCII, selanjutnya kunci *b* tersebut dijumlah dan dimodulo 95. Hasil yang diperoleh akan menjadi nilai pertama dan untuk mendapatkan nilai selanjutnya mengikuti aturan barisan Fibonacci. Jumlah

kunci *b* sama dengan *plaintext* yang akan dienkripsi.Modifikasi kunci ini bertujuan untuk mendapatkan karakter *ciphertext* yang berbeda-beda sehingga untuk memecahkan kunci yang digunakan lebih sulit dari pada *affine cipher* yang asli.

Pada penelitian ini dilakukan perbandingan antara *affine cipher*dan *affine cipher*dengan modifikasi kunci berdasarkan barisan Fibonacci. Perbandingan dilakukan dengan cara mencari nilai koefisien korelasi dari kedua algoritma tersebut. Nilai koefisien korelasi yang lebih kecil menunjukan bahwa algoritma tersebut lebih baik. Hasil yang di peroleh menunjukkan bahwa *affine cipher* dengan modifikasi kunci berdasarkan barisan Fibonacci lebih baik dari *affine cipher*.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat serta karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Pengamanan *Polyalphabetic* dengan *Affine Cipher* Berdasarkan Barisan Fibonacci". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

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

1. Dr. Kiswara Agung Santoso, S.Si., M.Kom., selaku Dosen Pembimbing Utama dan Ahmad Kasyamkawuni, S.Si., M.Kom., selaku Dosen Pembimbing anggota yang telah memberikan bimbingan dan bantuan dalam penyempurnaan skripsi ini;
2. Ikhsanul Halikin, S.Pd., M.Si., selaku Dosen Penguji I dan Kusbudiono, S.Si., M.Si., selaku Dosen Penguji II yang telah memberikan kritik dan saran yang membangun dalam penyempurnaan skripsi ini;
3. Kedua orang tua, Bapak Halil dan Ibu Asiati serta kakak, adek dan keluarga besar yang telah memberikan dukungan dan Doa;
4. Dosen dan Karyawan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
5. Seluruh teman-teman Angkatan 2016 (MISDIRECTION'16) serta sahabat-sahabatku yang telah memberikan dukungan dan pengalaman hidup yang luar biasa;
6. Semua pihak yang tidak bisa disebutkan satu persatu;

Penulis menerima segala kritik dan saran yang bersifat membangun dari semua pihak demi kesempurnaan penulisan skripsi ini. Akhirnya Penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Januari 2020

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin canggih memberikan perubahan yang besar terhadap kehidupan manusia. Saat ini berbagai macam kegiatan manusia dibantu dengan teknologi salah satunya dalam bidang komunikasi. Komunikasimerupakan proses penyampaian pesan dari seseorang yang ditujukan kepada orang lain. Dalam komunikasi sangat penting menjaga keamanan pengiriman pesan untuk menjaga privasi seseorang. Keamanan pengiriman pesan dapat menerapkan ilmu kriptografi.

Ilmu kriptografi merupakan ilmu ataupun seni yang mempelajari tentang menjaga keamanan pengiriman pesan dari seseorang yang ditujukan ke orang lain. Pesan asli yang akan dikirim disebut dengan *plaintext,* sedangkan pesan hasil dari pengkodean disebut dengan *ciphertext*. Enkripsi adalah proses mengubah *plaintext* menggunakan kode rahasia sehingga menghasilkan *ciphertext*. Dekripsi adalah proses mengubah *cipertext* ke bentuk aslinya dengan menggunakan kode rahasia. Kriptografi menjadi salah satu cabang ilmu matematika yang disebut *cryptology*. Terdapat dua macam kriptografi, yaitu kriptografi klasik dan kriptografi modern (Setyaningsih, 2015).

Kriptografi klasik adalah kriptografi zaman dahulu yang digunakan sebelum ditemukan komputer atau sudah ditemukan tetapi tidak canggih seperti komputer saat ini. Bentuk penyandiannya berupa karakter dengan menggunakan kertas dan pensil, atau mengunakan mesin sandi yang sederhana. Terdapat dua teknik dalam kriptografi, yaitu teknik transposisi dan teknik substitusi. Teknik transposisi merupakan proses enkripsi yang mengubah urutan pada *plaintext*, sedangkan teknik substitusi merupakan proses enkripsi untuk setiap karakter pada *plaintext* diganti dengan karakter lain. Terdapat empat istilah dalam teknik substitusi, yaitu *monoalphabet*, *polyalphabet*, *monograph*, dan *polygraph*. Ariyus (2008) mendefinisikan "*monoalphabe*t: setiap karakter teks-kode menggantikan salah satu karakter teks-asli. *Polyalphabet:* setiap karakter teks-kode dapat menggantikan lebih dari satu macam karakter teks-asli. *Monograph:* satu enkripsi

dilakukan terhadap satu karakter teks-asli. *Polygraph:* satu enkripsi dilakukan terhadap lebih dari satu karakter teks-asli". Contoh dari kriptografi klasik, yaitu *caesar cipher, vigenere cipher, affine cipher*, dan lain-lain (Ariyus, 2008).

Caesar cipher merupakan teknik substitusi pertama yang terjadi pada pemerintahan Julius Ceasar. Proses enkripsi *caesar cipher* dengan cara pergeseran sebanyak *K* ke kiri dalam alfabet. Misalkan pergeseran sebanyak 3 sehingga A menjadi D, B menjadi E, dan seterusnya. *Vigenere cipher* merupakan salah satu teknik substisusi *polyalphabet*. *Vigenere cipher* menggunakan teknik yang sama dengan *caesar cipher*, perbedaannya terletak pada proses enkripsi setiap karakter *plaintext* dapat dikodekan dengan kunci yang berbeda (Arius, 2008). *Affine cipher* adalah algoritma kriptografi klasik hasil pengembangan dari *caesar cipher*. Perbedaan yang mendasar teletak pada proses enkripsi menggunakan perkalian dengan bilangan yg relatif prima. Algoritma ini menggunakan kunci bilangan relatif prima dan bilangan bulat untuk penggeser (Prabhat dkk., 2014). Kekutan algoritma *affine cipher* terletak pada kunci yang digunakan. *Affine cipher* merupakan algoritma yang paling baik dari algoritma substitusi lainnya, karena kunci penggeser dapat menggunakan barisan tertentu. Barisan adalah bilangan yang membentuk suatu pola urutan tertentu. Terdapat beberapa macam barisan salah satunya Barisan Fibonacci. Tung (2008) mendefinisikan Barisan Fibonacci merupakan suatu urutan bilangan dimana nilai suku selanjutnya merupakan jumlah dua suku sebelumnya.

Beberapa artikel penelitian sebelummya yang berhubungan dengan *affine cipher*, yaitu Sriramoju (2017) dengan judul *modification affine cipher algorithm for cryptography password* yang membahas tentang modifikasi pada *plaintext*, dimana sebelum melakukan enkripsi dan dekripsi posisi *plaintext* dibalik, sehingga karakter pertama berada di posisi terakhir. Sharma, dkk (2018) dengan judul *Encryption of Hindi plaintext Using Modifited Affine cipher Technique* yang membahas tentang penerapan algoritma *affine cipher* dengan *plaintext* berupa karakter huruf india. Puspita (2015) dengan judul Sistem *Pengkodeaan Affine Berdasarkan Kunci Barisan Karakter* yang membahas tentang modifikasi pada kunci *b*, dimana kunci yang digunakan merupakan barisan karakter.

Berdasarkan uraian beberapa penelitian di atas, peneliti akan menggunakan algoritma *affine cipher* untuk keamanan pengiriman pesan teks dengan modifikasi kunci pada kunci penggeser. Pengirim pesan memberikan dua jenis kunci kepada penerima yaitu, kunci bilangan yang relatif prima sebagai pengali dan kunci penggeser berupa teks. Penerima pesan melakukan pembentukan pada kunci penggeser dengan aturan barisan Fibonacci untuk melakukan proses dekripsi sehingga menghasilkan pesan asli.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini sebagai berikut.

a. Bagaimana proses enkripsi dan dekripsi menggunakan *affine cipher* dengan modifikasi kunci berdasarkan barisan Fibonacci pada keamanan pengiriman pesan teks ?
b. Bagaimana pembuatan program dengan menggunakan *affine cipher*dengan modifikasi kunci berdasarkan barisan Fibonacci pada keamanan pengiriman pesan teks ?
c. Bagaimana hasil analisis perbandingan antara *affine cipher*dan*affine cipher* dengan modifikasi kunci berdasarkan barisan Fibonacci ?

1.3 Tujuan

Tujuan dalam penelitian ini sebagai berikut.

a. Menerapkan algoritma *affine cipher* dengan modifikasi kunci berdasarkan barisan Fibonacci pada keamanan pengiriman pesan teks.
b. Membuat program dengan menggunakan *affine cipher*dengan modifikasi kunci berdasarkan barisan Fibonacci pada keamanan pengiriman pesan teks.
c. Mengetahui algoritma yang lebih baik antara *affine cipher* dan *affine cipher* modifikasi kunci berdasarkan barisan Fibonacci.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah kunci pengali harus relatif prima dan kunci penggeser berupa teks.

1.5 Manfaat

Manfaat yang diperoleh dalam penelitian ini sebagai berikut.

a. Menjaga keamanan pengiriman pesan menggunakan algoritma *affine cipher* dengan modifikasi kunci.
b. Menambah pengetahuan baru tentang algoritma *affine cipher* dengan modifikasi kunci menggunakan barisan Fibonacci.

BAB 2. TINJAUAN PUSTAKA

2.1 Kriptografi

Teknologi informasi yang semakin canggih terutama dalam bidang komunikasi memudahkan proses dalam pengiriman pesan. Proses ini menjadi aspek yang sangat penting untuk menjaga keamanannya. Salah satu cara untuk menjamin proses pengiriman pesan yaitu dengan menerapkan ilmu kriptogari.

2.1.1 Pengertian Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa yunani, yaitu dari kata *cypto* yang berarti rahasia dan *graphia* yang berarti tulisan. Kriptografi merupakan ilmu yang mempelajari tentang keamanan pengiriman pesan dari seseorang yang ditujukan ke orang lain (Ariyus, 2008). Kriptografi adalah salah satu cabang ilmu matematika yang disebut dengan kriptologi (*cryptolgy*). Kriptografi bertujuan menjaga kerahasian informasi yang terkandung dalam sebuah data sehingga tidak diketahui oleh pihak yang tidak sah. Kriptografer adalah perancang dari algoritma kriptografi.

Kriptanalisis (*cryptanalysis*) merupakan ilmu dan seni mengubah *ciphertext* menjadi *plaintext* tanpa memahami kunci yang digunakan. Orang yang melakukan kriptanalisis disebut kriptanalis (*cryptanalys*). Kriptanalis merupakan kebalikan dari kriptografer. Kriptografi merupakan studi terhadap teknik matematis yang berhubungan dengan aspek keamanan sistem informasi seperti kerahasiaan, integritas data, autentikasi, dan ketiadaan penyangkalan. Aspek tersebut merupakan tujuan dari sistem kriptografi.

a. Kerahasiaan (*confidentiality*)

Kerahasiaan adalah layanan yang digunakan untuk menjaga informasi dari pihak yang tidak berhak mengaksesnya. Informasi ini hanya dapat diakses oleh orang-orang yang berhak. Contohnya adalah sniffing. Pengamanannya dilakukan dengan metode enkripsi.

b. Integritas Data (*data integrity*)

Integritas data adalah layanan yang berfungsi untuk mencegah terjadinya pengubahan informasi oleh pihak yang tidak berhak. Integritas data harus

menjamin agar sistem informasi dapat mendeteksi terjadinya pemalsuan data. Pemalsuan data yang dimaksud meliputi penyisipan, penghapusan, atau penggantian data. Contohnya adalah *spoofing*, virus, *trojan horse*, atau *man in the middle attack*. Pengamanan dilakukan dengan *signature*, *certificate*, dan *hash*.

c. Autentikasi (*authentication*)

Autentikasi adalah layanan yang berhubungan dengan indentifikasi terhadap pengguna yang ingin mengakses sistem informasi (*authentication*) atau keaslian data dari sistem informasi (*data origin authentication*). Contohnya adalah password palsu, situs web palsu, dan lain-lain. Pengamanan dilakukan dengan *certificates.*

d. Ketiadaan Penyangkalan (*nonrepudiation*)

Ketiadaan penyangkalan adalah layanan yang bertujuan untuk mencegah terjadinya penyangkalan, seperti pengirim yang menyangkal tidak mengirim pesan atau penerima yang menyangkal tidak menerima pesan(Setyaningsih, 2015).

2.1.2 Mekanisme Kriptografi

Kriptografi pada umumnya digunakan dalam sistem keamanan informasi. Cara kerja dari kriptografi yaitu dengan menyandikan sebuah pesan menggunakan kode rahasia sehingga pesan tersebut hanya dipahami oleh pihak tertentu. Cara kerja seperti ini sudah digunakan sejak romawi kuno. Pada teknologi informasi saat ini cara kerja yang sama masih digunakan, namun penerapannya berbeda. Terdapat beberapa istilah yang umum digunakan dalam kritografi, yaitu sebagai berikut.

a. *Plaintext*

Plaintext merupakan pesan asli yang akan dikirim kepada seseorang dan harus dijaga keamanannya.

b. *Ciphertext*

Ciphertext merupakan pesan yang sudah dikodekan dengan kode rahasia.

c. *Cipher*

Cipher merupakan algoritma yang digunakan untuk mengodekan *plaintext* menjadi *ciphertext.*

d. Enkripsi

Enkripsi merupakan proses mengodekan *plaintext* menjadi *ciphertext*.

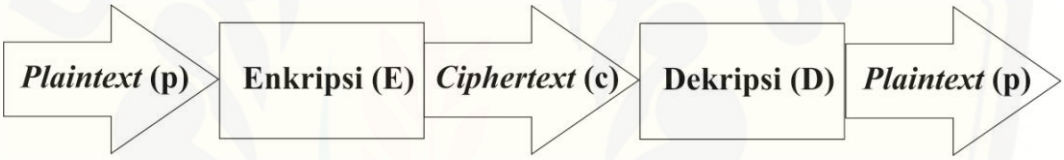
e. Dekripsi

Dekripsi merupakan proses untuk mendapatkan kembali pesan yang asli sebelum dikodekan.

f. Sistem Kriptografi

Sistem kriptografi merupakan sistem yang diciptakan untuk mengamankan sistem informasi dengan menggunakan kriptografi.

Proses kriptografi pada dasarnya sangat sederhana. *Plaintext* (p) akan diubah melalui proses enkripsi (E) sehingga menghasilkan *ciphertext* (c). Selanjutnnya, untuk mendapatkan kembali *plaintext* (p), *ciphertext* (c) melalui proses dekripsi (D) akan menghasilkan kembali *plaintext* (p).



Gambar 2.1 Mekanisme kriptografi

Secara matematis proses kriptografi sebagai berikut.

Proses enkripsi

$$c = E(p)$$

Proses dekripsi

$$p = D(c)$$

2.1.3 Sejarah kriptografi

Kriptografi memiliki sejarah yang sangat panjang. Sejak 4000 tahun yang lalu kriptografi sudah diperkenalkan oleh orang-orang mesir melalui *hieroglyph. Hieroglyph* berasal dari bahasa yunani *hieroglyphica* yang artinya ukiran rahasia. Pada zaman romawi kuno Julius Caesar ingin mengirim pesan kepada jendralnya. Pesan tersebut dikirim melalui seorang pelayan. Julius Caesar tidak ingin pesan tersebut diketahui oleh pihak lain karna bersifat rahasia. Julius Caesar memutuskan untuk mengacak pesan tersebut dengan mengganti semua susunan huruf abjad, sehingga hanya dapat dipahami oleh jendralnya. Proses pengacakan

pesan yang dilakukan Julius Caesar disebut dengan enkripsi, sedangkan proses mengembalikan pesan yang dilakukan sang jendral disebut dengan dekripsi (Ariyus, 2008).

2.1.4 Sistem Kriptografi

Sistem kriptografi merupakan sistem yang diciptakan untuk mengamankan informasi dengan menggunakan kriptografi. Karakteristik yang harus dimiliki oleh sistem kriptografi, yaitu sebagai berikut.

a. Aspek yang paling penting dalam keamanan sistem terletak pada kerahasiaan kunci yang digunakan bukan terletak pada algoritmanya.
b. Sistem kriptografi yang baik memiliki kunci yang panjang.
c. *Ciphertext* yang dihasilkan harus terlihat acak
d. Sistem kriptografi harus dapat menahan serangan dari kriptanalis

Secara umum, berdasarkan jenis kunci yang digunakan dalam proses enkripsi dan dekripsi sistem kriptografi dibedakan menjadi dua jenis, yaitu:

1) Kriptografi kunci simetri

Kriptografi kunci simetri (*symmetric cryptosystem*) menggunakan kunci yang sama pada proses enkripsi dan dekripsi. Kriptografi ini sangat menjaga kerahasiaan dari kunci yang digunakan untuk proses enkripsi dan dekripsi. Kriptografi kunci simetri disebut juga dengan kriptografi kunci rahasia (*secret-key cryptography*) yang merupakan bentuk kriptografi sederhana dengan menggunakan kunci tunggal pada proses enkripsi dan dekripsi pesan. Pengirim dan penerima memiliki kunci yang sama. Secara matematis kriptografi kunci simetri dapat dinyatakan sebagai berikut.

$$E_k(p) = c$$
$$D_k(c) = p$$

2) Kriptografi kunci asimetri

Kriptografi kunci asimetri (*assymmetric cryptosystem*) menggunakan dua buah kunci, yaitu kunci publik (*public key*) yang digunakan pada proses enkripsi dan kunci pribadi (*private key*) yang digunakan pada proses dekripsi. Nama lain dari kunci asimetri adalah *cryptosystem* kunci publik (*public key cryptosystem*). Salah satu penerapan dari kunci asimetri yaitu pada PGP dimana proses enkripsi *e-mail*

menggunakan kunci publik, sedangkan proses dekripsi menggunakan kunci pribadi (Setyaningsih, 2015).

2.2 Kriptografi Klasik

Kriptografi klasik merupakan kriptografi zaman dahulu sebelum komputer ditemukan atau sudah ada tetapi tidak canggih seperti saat ini. Bentuk penyandian dari kriptografi klasik berupa karakter dengan menggunakan kertas dan pensil, atau dengan mesin sandi yang sederhana. Kriptografi klasik termasuk dalam kriptografi kunci simetri karena sudah digunakan jauh sebelum ditemukan kriptografi kunci asimetri. Terdapat dua macam teknik dalam kriptografi klasik, yaitu:

a. Teknik substitusi

Teknik substitusi adalah proses enkripsi yang mengganti setiap karakter pada *plaintext* dengan karakter lain. Teknik substitusi terdiri empat jenis, yaitu abjad-tunggal (*monoalphabetic*), abjad-majemuk (*polyalphabetic*), *monograph*, dan *polygraph*.

1) Abjad-tunggal (*monoalphabetic*)

Abjad-tunggal merupakan setiap karakter pada *ciphertext* hanya dapat menggantikansatu karakter pada *plaintext*. contoh dari abjad tunggal sebagai berikut.

a) *Caesar cipher*

Caesar cipher merupakan algoritma kriptografi yang sudah ada sejak zaman Julius Caesar. Algoritma ini digunakan oleh julius untuk mengirim pesan yang ditujukan kepada jendralnya. Proses enkripsinya menggunakan teknik pergeseran karakter pada alfabet 3 langkah ke kiri. Pergeseran kunci yang dilakukan tergantung keinginan pengirim pesan. Jadi, untuk setiap pergeseran karakter ke-*i* pada alfabet menjadi karakter ke-*i+k* pada urutan alfabet.

Secara matematis proses enkripsi dan dekripsi dari *caesar cipher* sebagai berikut.

Proses enkripsi

$$C=E(P) = (P+K) \bmod (26) \qquad (2.1)$$

Proses dekripsi

$$P = D(C) = (C-K) \bmod (26) \quad (2.2)$$

Keterangan :

P : plaintext

C : ciphertext

K : kunci yang digunakan

E : proses enkripsi

D : proses dekripsi

Berikut ini merupakan contoh dari caesar cipher dengan pergeseran 3 yang digunakan oleh Julius Caesar dilihat pada Tabel 2.1

Tabel 2.1 : susunan alfabet setelah digeser 3 huruf

P =	**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 =	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

b) *Affine cipher*

Affine cipher merupakan algoritma kriptografi klasik hasil pengembangan dari *caesar cipher*. Perbedaan yang mendasar dari algoritma ini teletak pada proses enkripsi menggunakan perkalian dengan bilangan yang relatif prima. Algoritma ini menggunakan kunci bilangan relatif prima dan bilangan bulat untuk penggeser. *Affine cipher* termasuk kriptogarfi kunci simetri karena proses enkripsi dan dekripsi menggunakan kunci yang sama. Proses enkripsi dan dekripsi *affine cipher* menggunakan dua jenis kunci untuk mendapatkan *ciphertext* yaitu, kunci *a* sebagai pengali yang merupakan bilangan relatif prima terhadap modulo yang digunakan dan kunci *b* bilangan bulat sebagai penggeser. Kunci *a* harus memiliki invers perkalian sehingga harus memenuhi FPB(*a, m*)=1. Secara matematis proses enkripsi dan dekripsi sebagai berikut.

Proses enkripsi

$$C_i = (aP_i + b) mod\ m \quad (2.3)$$

Proses dekripsi

$$P_i = a^{-1}(C_i - b) mod m \quad (2.4)$$

Keterangan :

P_i : *plaintext*

C_i: *ciphertext*

a : kunci bilangan yang relatif prima

b : kunci bilangan intereger

a^{-1} : invers dari a

m : modulo yang digunakan

Contoh proses enkripsi dan dekripsi pada *affine cipher*

Plaintext = FIDI; a = 7 dan b= 3; m=26: a^{-1}=15

Penyelesain:

Plaintext diubah ke dalam bentuk desimal menggunakan Tabel 2.2

Tabel 2.2 Konversi Desimal

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
0	1	2	3	4	5	6	7	8	9	10	11	12

N	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
13	14	15	16	17	18	19	20	21	22	23	24	25

Didapatkan :

F	I	D	I
5	8	3	8

Proses enkripsi menggunakan persamaan (2.3)

Plaintext	F	I	D	I
P_i	5	8	3	8
$7P_i + 3$	38	59	24	59
$7P_i + 3 \ mod \ 26$	12	7	24	7

Ciphertext	M	H	Y	H

Proses dekripsi menggunakan persamaan (2.4)

Ciphertext	M	H	Y	H
C_i	12	7	24	59
$15(C_i - 3)$	135	60	315	60
$15(C_i - 3) mod\ 26$	5	8	3	8
Plaintext	F	I	D	I

2) Cipher Abjad-majemuk (*polyalphabetic*)

Cipher abjad-majemuk merupakan setiap karakter *ciphertext* dapat menggantikan lebih dari satu jenis karakter pada *plaintext.* Cipher abjad-majemuk dibuat dari sejumlah cipher abjad tunggal dimana masing-masing menggunakan kunci yang berbeda. Metode ini mulai dikembangkan pada tahun 1568 oleh Leon Battista Alberti. Alberti menggunakan suatu *disk cipher* yang didefinisikan sebagai substitusi berganda(Setyaningsih, 2015).

Cipher abjad-majemuk membuat *ciphertext* sulit untuk dipecahkan karna karakter *plaintext* yang sama tidak dienkripsi dengan *ciphertext* yang sama. Penerapan cipher abjad-majemuk pada umumnya adalah mengulang kunci sesuai dengan panjang *plaintext* dengan kata lain panjang kunci sama dengan panjang *plaintext.* Seperti contoh sebagai berikut: Misal, diketahui kunci = KUNCI, sehingga kunci diperluas menjadi KUNCIKUNCI... sampai ukurannya sama dengan *plaintext.* Jika setiap huruf diberi bobot A = 0, B = 1, ..., Z = 25.

Plaintext : KRIPTOGRAFI

Kunci : KUNCIKUNCIK

ciphertetx : ULVRBYAECNS

(K+K) mod 26 = (10+10) mod 26=20=U

(R+U) mod 26 = (17+20) mod 26=11=L

(I+N) mod 26 = (8+13) mod 26=21=V

(P+C) mod 26 = (15+2) mod 26=17=R

(T+I) mod 26 = (19+8) mod 26=1=B

(O+K) mod 26 = (14+10) mod 26=24=Y

(G+U) mod 26 = (6+20) mod 26=0=A

(R+N) mod 26 = (17+13) mod 26=4=E

(A+C) mod 26 = (0+2) mod 26=2=C

(F+I) mod 26 = (5+8) mod 26=13=N

(I+K) mod 26 = (8+10) mod 26=18=S

Metode abjad-majemuk juga menghasilkan pola enkripsi yang lebih acak karena setiap karakter yang sama menghasilkan karakter *ciphertext* yang berbeda. *Plaintext* diatas memiliki dua huruf I yang menghasilkan *ciphertext* yang berbeda yaitu huruf V dan S. Contoh dari metode Abjad-majemuk, yaitu *vigenere cipher* dan *beaufort cipher* (Permanasari, 2018).

b. Teknik Transposisi

Teknik transposisi merupakan proses enkripsi yang mengubah urutan pada *plaintext.* Nama lain dari algoritma ini adalah permutasi karena transposisi setiap karakter dalam teks sama dengan mempermutasikan karakter-karakter tersebut dengan menggunakan kunci penggambaran tambahan dan metode untuk menuliskan dan meletakkannya ke dalam urutan tertentu dengan cara zig-zag perbaris. Algoritma ini memiliki banyak cara dalam pengubahan urutan pada karakternya, diantaranya transposisi grup, transposisi serial, dan transposisi kolom atau baris(Setyaningsih, 2015).

2.3 Invers modulo

Affine cipher memliki dua buah kunci yaitu, kunci pengali dan kuncipenggeser. Kedua kunci tersebut digunakan dalam proses enkripsi dan dekripsi. Kunci pengali harus relatif prima dengan modulo yang digunakan, sehingga kunci tersebut memiliki invers modulo terhadap pekalian. Dua buah bilangan bulat *a*dan *m* dikatakan relatif prima jika FPB(*a*,*m*) = 1. Invers dari kunci pengali akan digunakan dalam proses dekripsi.

Jika *a* dan *m* relatif prima dan *m* > 1, maka kita dapat menemukan invers dari *a* modulo *m*. Invers dari *a* modulo *m* adalah bilangan bulat *x* sedemikian sehingga

$$x\ a = 1\ (mod\ m)$$

(Munir, 2004)

Apabila mencari invers modulo, kita dapat menggunakan perluasan dari algoritma *euclidien.* Algoritma *euclidien* merupakan algoritma untuk mencari FPB dari dua buah bilangan bulat. Algoritma *euclidien* dihitung dengan algoritma pembagian sebagai berikut.

$$\left.\begin{aligned} r_0 &= q_1 r_1 + r_2\ ,\ 0 < r_2 < r_1 \\ r_1 &= q_2 r_2 + r_3\ ,\ 0 < r_3 < r_2 \\ &\vdots \\ r_{n-2} &= q_{n-1} r_{n-1} + r_n\ ,\ 0 < r_n < r_{n-1} \\ r_n &= q_n r_n \end{aligned}\right\} \quad (2.5)$$

(Buchman, 2000).

Contoh mencari invers dari *7 mod 26*menggunakan perluasan dari algoritma *euclidean*.

$$\left.\begin{aligned} 26 &= 7.3 + 5 \quad \Longrightarrow \quad 5 = 26 - 7.3 \\ 7 &= 5.1 + 2 \quad \Longrightarrow \quad 2 = 7 - 5.1 \\ 5 &= 2.2 + 1 \quad \Longrightarrow \quad 1 = 5 - 2.2 \\ 2 &= 1.2 + 0 \end{aligned}\right\} \quad (2.6)$$

Substistusikan persamaan (2.6)

$$\begin{aligned} 1 &= 5 - 2.2 \\ 1 &= 5 - (7 - 5.1)\ .\ 2 \\ 1 &= 5 - 7.2 + 5.2 \\ 1 &= 5.3 - 7.2 \\ 1 &= (26 - 7.3)\ .\ 3 - 7.2 \\ 1 &= 26.3 - 7.9 - 7.2 \\ 1 &= 26.3 + 7\ (-11) \end{aligned}$$

Dari persamaan terakhir diatas -11 *mod* 26 = 15. Sehingga diperoleh 15 merupakan invers dari 7 *mod* 26.

2.4 Bilangan Fibonacci

Bilangan Fibonacci ditemukan oleh Leonardo da Pisa dalam bukunya Liberabaci tahun 1202, namun Leonardo da Pisa sesudah itu dikenal dengan Fibonacci. Dalam matematika bilangan Fibonacci adalah deretan yang didefinisikan secara rekursif sebagai berikut:

$$F(n) = \begin{cases} 0 & \text{Jika } n = 0 \\ 1 & \text{Jika } n = 1 \\ F(n-1) + F(n-2) & \text{Jika tidak} \end{cases} \quad (2.7)$$

Berdasarkan definisi dari $F(n)$. $f_3 = f_1 + f_2$; $f_4 = f_2 + f_3$;$f_5 = f_3 + f_4$ dan seterusnya. Deret ini berawal dari 0 dan 1, kemudiannilai suku selanjutnya merupakan jumlah dua suku sebelumnya.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,...

(1+1=2), (1+2=3), (2+3=5), dan seterusnya (Tung, 2008).

2.5 Kode ASCII

ASCII *(American Standard Code for Information Interchange)* merupakan kode huruf dan symbol yang berstandard internasional. Kode ASCII selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 bit, namun ASCII disimpan sebagai sandi 8 bit dengan menambahkan satu angka 0 sebagai bit significant paling tinggi. Jumlah kode ASCII sebanyak 255 kode. Kode ASCII 0...127 merupakan kode ASCII untuk manipulasi teks, sedangkan kode ASCII 128..255 merupakan kode ASCII untuk manipulasi grafik. Kode ASCII sendiri dapat dikelompokkan lagi kedalam beberapa bagian, yaitu:

1. Kode yang tidak terlihat simbolnya seperti kode 10 (*Line Feed*), 13 (*Carriage Return*),, 8 (Tab), 32 (*Space*)
2. Kode yang terlihat simbolnya seperti abjad (A...Z) numerik (0...9), karakter khusus (~!}@#$%^&*()_+?:{})

3. Kode yang tidak ada di keyboard namun dapat ditampilkan. Kode ini umumnya untuk kode-kode grafik

(Irawan, 2015).

2.6 Analisis Koefisien Korelasi

Perhitungan koefisien korelasi *plaintext* dan *cipertext* bertujuan untuk mengetahui hubungan linier antara *plaintext* dan *ciphertext*tersebut. Jika *plaintext* dan *ciphertext* cenderung mengikuti garis lurus dengan kemiringan yang sama, maka menghasilkan nilai korelasi positif yang tinggi antara keduanya. Sedangkan untuk *plaintext* dan *ciphertext* yang mengikuti garis lurus dengan kemiringan berbeda, maka menghasilkan nilai korelasi yang negatif. Jika nilai korelasi yang dihasilkan 1 atau -1 maka memiliki hubungan linier yang kuat antara *plaintext* dan *ciphertext* . Dalam kriptografi hal ini merupakan enkripsi yang tidak baik. Jika korelasi bernilai 0 maka*plaintext* dan *ciphertext* tidak memiliki hubungan linear sehingga algoritma kriptografi memiliki proses enkripsi yang baik (Cahyono, 2016)

Perhitungan nilai korelasi dilakukan dengan menggunakan persamaan berikut.

$$KK(x,y) = \frac{\sum_{i=1}^{n}\big(x_i - \mu(x)\big)\big(y_i - \mu(y)\big)}{\sigma(x)\sigma(y)} \tag{2.8}$$

Dengan:

$$\mu(x) = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{dan} \quad \mu(y) = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\sigma(x) = \sqrt{\sum_{i=1}^{n}(x_i - \mu(x))^2} \quad \text{dan} \quad \sigma(y) = \sqrt{\sum_{i=1}^{n}(y_i - \mu(y))^2}$$

Keterangan ;

KK	=	Nilai Koefisien Korelasi
x	:	*Plaintext*
y	:	*Ciphertext*
$\mu(x)$	:	Rata-rata *plaintext*
$\mu(y)$	:	Rata-rata *ciphertext*
$\sigma(x)$	:	Standar deviasi *plaintext*
$\sigma(y)$	:	Standar deviasi *ciphertext*

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Penelitian ini menggunakan data berupa teks. Teks tersebut dapat berupa karakter alfabet, angka, simbol, dan tanda spasi yang terdapat dalam *printable* ASCII. Data yang digunakan akan dikonversi ke bentuk desimal menggunakan tabel ASCII sebelum proses enkripsi atau dekripsi.

3.2 Langkah-langkah Penelitian

Langkah-langkah yang akan dilakukan dalam penelitian, yaitu:

a. Studi Literatur

   Tahap ini dilakukan bertujuan untuk memahami beberapa teori yang berhubungan dengan penelitian. Teori yang harus dipelajari adalah algoritma *affine cipher* dan barisan Fibonanci yang digunakan untuk pembentukan kunci penggeser.

b. Perhitungan Manual

   Tahap ini dilakukan perhitungan manual contoh dari proses enkripsi dan dekripsi *affine cipher* denganmodifikasi kunci berdasarkan barisan Fibonacci.

c. Perancangan Program

   Perancangan Program dalam tahap ini menggunakan *flowchart*. Perancangan desain GUI (Guide User Interface) dengan menggunakan *software* MATLAB. Proses perancangan desain seperti tata letak tombol push button dan edit text.

d. Pembuatan Program

   Pembuatan program dilakukan menggunakan *software* MATLAB berdasarkan algoritma yang digunakan pada proses enkripsi dan dekripsi. Proses enkripsi dan dekripsi menggunakan modifikasi kunci dari algoritma *affine cipher*. Kunci penggeser yang digunakan merupakan kunci hasil pembentukan dari aturan Barisan Fibonacci. Langkah-langkah yang dilakukan pada proses enkripsi dan dekripsi sebagai berikut.

   1) Enkripsi

Terdapat beberapa tahapan dalam proses enkripsi, yaitu:

a) Konversi *Plaintext*

Pada langkah ini *plaintext* yang akan melalui proses enkripsi dikonversi terlebih dahulu kedalam bentuk desimal menggunakan tabel ASCII.

b) Pembentukan Kunci

Kunci terdiri dari dua jenis yaitu, kunci bilangan relatif prima sebagai pengali dan kunci pengeser berupa teks. Kunci penggeser yang akan digunakan harus dikonversi terlebih dahulu ke bentuk desimal. Pembentukan kunci penggeser dilakukan dengan menjumlahkan semua karakter dari kunci penggeser. Selanjutnya, pembentukan kunci dilakukan menggunakan aturan barisan Fibonacci dengan memanfaatkan Persamaan (2.7) dan jumlah karakter yang digunakan adalah 95, secara sistematis didapat Persamaan (3.1).

$$F(n) = F(n-1) + F(n-2) \ mod \ 95 \tag{3.1}$$

c) Proses Enkripsi

Poses enkripsi pesan dilakukan menggunakan algoritma *affine cipher*dengan modifikasi kunci berdasarkan barisan Fibonacci. Algoritma ini memiliki dua kunci yaitu, kunci *a* merupakan bilangan yang relatif prima dan kunci *b* sebagai kunci penggeser berupa teks. Hasil pembentukan kunci yang telah dilakukan merupakan kunci penggeser dari algoritma ini. Setiap karakter *plaintext* memiliki kunci penggeser yang berbeda-beda. Proses enkripsi dilakukan dengan memanfaatkan Persaaman (2.3) dan jumlah karakter yang digunakan adalah 95 dimulai dari kerakter yang ke 32 sampai 127 dalam tabel ASCII, sehingga *plaintext* terlebih dahulu dikurangi 32 dan untuk mendapatkan*ciphertext* ditambah 32. Secara sistematis didapat Persamaan (3.2).

$$C_i = \left((a(P_i - 32) + \ b_i) mod 95\right) + 32 \tag{3.2}$$

2) Dekripsi

Terdapat beberapa tahapan dalam proses dekripsi, yaitu:

a) Konversi*Ciphertext*

Ciphertext yang akan didekripsi dikonversi terlebih dahulu kedalam bentuk desimal

b) Pembentukan Kunci

Pada langkah ini pembentukan kunci yang akan dilakukan sama dengan pembentukan kunci pada proses enkripsi. Sehingga kunci yang digunakan pada proses enkripsi dan dekripsi adalah kunci yang sama (kunci tunggal). Selain itu, pada tahap ini dilakukan pencarian terhadap invers dari kunci bilangan yang relatif prima (a^{-1}) dengan Persamaan (2.5).

c) Proses Dekripsi

Pada tahap ini dilakukan untuk mengubah *ciphertext* menjadi *plaintext.*Proses dekripsi dilakukan dengan dengan memanfaatkan persaaman (2.4) dan jumlah karakter yang digunakan adalah 95, dimulai dari kerakter yang ke 32 sampai 127 dalam tabel ASCII, sehingga *ciphertxt* terlebih dahulu dikurangi 32 dan untuk mendapatkan*plaintext* maka harusditambah 32. secara sistematis didapat Persamaan (3.3).

$$P_i = (a^{-1}((C_i - 32) - b_i) mod 95) + 32 \quad (3.3)$$

e. Analisis Hasil

Menganalisa dan menguji program yang telah dibuat sesuai dengan metode yang digunakan dan membandingkan *affine cipher* sebelum dan sesudah modifikasi kunci menggunakan analisis koefisienkorelasi.

f. Kesimpulan

Mengambil kesimpulan dari penelitian yang telah dilakukan dengan menganalisis sebelum dan sesudah modifikasi kunci *affine cipher*.

Gambar 3.1 Diagram Alir Penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan sebagai berikut.

a. Proses enkripsi menggunakan *affine cipher*dengan modifikasi kunci berdasarkan barisan Fibonacci dapat menghilangkan kelemahan pada *affine cipher*yang hanya menggunakan kunci *b* sebanyak 1 karakter. *Ciphertext* yang dihasilkan terlihat acak dan tidak memiliki pola.
b. Pembuatan program menggunakan *affine cipher* dengan modifikasi kunci berdasarkan barisan Fibonacci menggunakan *software* Matlab 2015a. Pembuatan program yang telah dilakukan memudahkan peneliti dalam poses enkripsi dan dekripsi.
c. Hasil analisis pada percobaan yang telah dilakukan membuktikan bahwa *affine cipher* dengan modifikasi kunci berdasarkan barisan Fibonacci lebih baik dari pada *affine cipher*.

5.2 Saran

Saran yang dapat penulis berikan untuk mengembangkan algoritma dalam penelitian ini sebagai berikut.

a. Algoritma dalam penelitian ini dapat diterapkan pada gambar atau media lainnya.
b. Algoritma dalam penelitian ini juga dapat diterapkan pada steganografi dimana hasil dari *ciphertext* dalam penelitian ini disembunyikan pada gambar.

DAFTAR PUSTAKA

Ariyus, D. 2008. *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasinya.* Yogyakarta: CV. Andi Offset.

Bawono, H. R. C. 2015. *Kriptanalisis Pada Algoritma Vigenere*. Skripsi. Yogyakarta: Universitas Sanata Dharma.

Buchmann, J.A. 2000. *Introduction to Cryptography.*New York: Springer-Verlag.

Bruen, A.A., M.A. Forcinito. 2005. *Cryptography, Information Theory, and Error correction: Ahandbook for the 21st Century.* A John Wiley & Sons Inc. New Jersey.

Cahyono, J. 2016. Konstruksi Suatu Algoritma Kriptografi Menggunakan Transformasi Max Plus Wavelet. *Tesis.* Surabaya: Program Magister Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Teknologi Sepuluh Nopember.

Irawan, R., Ilhamsyah, dan Y. Brianorman. 2015. Aplikasi Enkripsi dan Dekripsi Pesan Singkat Menggunakan Algoritma Knapsack Berbasis Android. *Jurnal Coding Sistem Komputer Untan.* 3(3):57-56

Munir, R. 2004. *Diktat Kuliah IF3058 Teori Bilangan.* Bandung: Departemen Teknik Informatika Institut Teknologi Bandung

Munir, R. 2010. *Diktat Kuliah IF3058 Kriptografi.* Bandung: Departemen Teknik Informatika Institut Teknologi Bandung.

Permanasari, Y., dan E. Harahap. 2018. Kriptografi Polyalphabetic. *Jurnal Matematika.* 17(1):31-34

Prabhat, S.S & Verma, K. 2014. Implementation of Affine Substitution Cipher with Keyed Transposition Cipher for Enhancing Data Security.

International Journal of Advanced Research in ComputerScience and Software Engineering 4(1): 237-238.

Puspita, Y. I. 2015. Sistem Penggodean Affine Berdasarkan Kunci Barisan. *Skripsi*. Jember: Program Sarjana Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Setyaningsih, E. 2015. *Kriptografi dan Implementasinya Menggunakan Matlab*. Yogyakarta: CV. Andi Offset.

Shah, A, A. Z. Saidin, I. F. Taha, A. M. Zeki, dan Z. Bhatti. 2013. Similarities and Dissimilarities between Character Frequencies of Written Text of Melayu, English, and Indonesian Languages, *2nd International Conference on Advanced Computer Scienci Application and Technologie*. Kuching, Malaysia. 22-24 Desember: 192-194.

Sharma, P., P. Bhatpahri, dan R. Shrivastava. 2018. Encryption of Hindi Plaintext Using Modified Affine Cipher Technique. *International Journal of education and Information Technology* 3(4): 107-111.

Sriramoju, A. B. 2017. Modification Affine Ciphers Algorithm for Cryptography Password. *International Journal of Research in Science & Engineering* 4(1): 2394-8299.

Tung, K. Y. 2008. *Memahami Teori Bilangan dengan Mudah dan Menarik*. Jakarta: PT. Grasindo.

LAMPIRAN

LAMPIRAN A. Tabel ASCII *Printable Characters*

Desimal	Biner	Karakter	Desimal	Biner	Karakter
032	0100000	Space	055	0110111	7
033	0100001	!	056	0111000	8
034	0100010	“	057	0111001	9
035	0100011	#	058	0111010	:
036	0100100	$	059	0111011	;
037	0100101	%	060	0111100	<
038	0100110	&	061	0111101	=
039	0100111	‘	062	0111110	>
040	0101000	(	063	0111111	?
041	0101001	)	064	1000000	@
042	0101010	*	065	1000001	A
043	0101011	+	066	1000010	B
044	0101100	,	067	1000011	C
045	0101101	-	068	1000100	D
046	0101110	.	069	1000101	E
047	0101111	/	070	1000110	F
048	0110000	0	071	1000111	G
049	0110001	1	072	1001000	H
050	0110010	2	073	1001001	I
051	0110011	3	074	1001010	J
052	0110100	4	075	1001011	K
053	0110101	5	076	1001100	L
054	0110110	6	077	1001101	M

Desimal	Biner	Karakter	Desimal	Biner	Karakter
078	1001110	N	103	1100111	g
079	1001111	O	104	1101000	h
080	1010000	P	105	1101001	i
081	1010001	Q	106	1101010	j
082	1010010	R	107	1101011	k
083	1010011	S	108	1101100	l
084	1010100	T	109	1101101	m
085	1010101	U	110	1101110	n
086	1010110	V	111	1101111	o
087	1010111	W	112	1110000	p
088	1011000	X	113	1110001	q
089	1011001	Y	114	1110010	r
090	1011010	Z	115	1110011	s
091	1011011	[	116	1110100	t
092	1011100	\	117	1110101	u
093	1011101	]	118	1110110	v
094	1011110	^	119	1110111	w
095	1011111	_	120	1111000	x
096	1100000	`	121	1111001	y
097	1100001	a	122	1111010	z
098	1100010	b	123	1111011	{
099	1100011	c	124	1111100	|
100	1100100	d	125	1111101	}
101	1100101	e	126	1111110	~
102	1100110	f	127	1111111	*delete*

LAMPIRAN B. Skrip Program Enkripsi dan Dekripsi *Affine Cipher* pada Matlab

a. Skrip Program Enkripsi pada *Affine Cipher*

```
function Ciphertext = AffineEN(Plaintext, KeyAff, KeyAff2, L)
Ciphertext = '';
if nargin == 3
    L = 95;
end

% Convert Char to Dec
Plaindec = double(Plaintext)-32;
Key2dec = repmat(sum(double(KeyAff2)),1,length(Plaindec));

% Affine Key
Key1dec = RPrima(KeyAff, L);
if Key1dec == 0
    errordlg('Affine Key must be relatively prime with 95');
else
    %Affine Cipher
    Cipherdec = mod(KeyAff*Plaindec + Key2dec, L) + 32;

    %Output
    Ciphertext = char(Cipherdec);
end
```

b. Skrip Program Dekripsi pada *Affine Cipher*

```
function Plaintext = AffineDE(Ciphertext, KeyAff, KeyAff2, L)
Plaintext = '';
if nargin == 3
    L = 95;
end

% Convert Char to Dec
Cipherdec = double(Ciphertext)-32;
Key2dec = repmat(sum(double(KeyAff2)),1,length(Cipherdec));

% Affine Key
Key1dec = RPrima(KeyAff, L);
if Key1dec == 0
    errordlg('Affine Key must be relatively prime with 95');
else
    %Invers Affine Key
    Key3dec = InvPrime(KeyAff, L);

    %Affine Cipher
    Plaindec = mod(Key3dec*(Cipherdec - Key2dec), L) + 32;
```

```
    %Output
    Plaintext = char(Plaindec);
end
```

LAMPIRAN C. Skrip Program Enkripsi dan Dekripsi *Affine Cipher* dengan Modifikasi Kunci Berdasarkan barisan Fibonacci pada Matlab

a. Skrip Program Enkripsi pada *Affine Cipher* Modifikasi Kunci Berdasarkan barisan Fibonacci

```
function Ciphertext = AffineFiboEN(Plaintext, KeyAff, KeyFibo,
L)
Ciphertext = '';
if nargin == 3
    L = 95;
end

% Convert Char to Dec
Plaindec = double(Plaintext)-32;
KeyFibodec = repmat(sum(double(KeyFibo)),1,2);
N = length(Plaindec);

% Affine Key
Key1dec = RPrima(KeyAff, L);
if Key1dec == 0
    errordlg('Affine Key must be relatively prime with 95');
else
    % Fibonacci Key
    Key2dec = Fibokey(KeyFibodec, N, L);

    %Affine Cipher
    Cipherdec = mod(KeyAff*Plaindec + Key2dec, L) + 32;

    %Output
    Ciphertext = char(Cipherdec);
End
```

b. Skrip Program Dekripsi pada *Affine Cipher* Modifikasi Kunci Berdasarkan barisan Fibonacci

```
function Plaintext = AffineFiboDE(Ciphertext, KeyAff, KeyFibo,
L)
Plaintext = '';
if nargin == 3
    L = 95;
end

% Convert Char to Dec
Cipherdec = double(Ciphertext)-32;
```

```
KeyFibodec = repmat(sum(double(KeyFibo)),1,2);
N = length(Cipherdec);

% Affine Key
Key1dec = RPrima(KeyAff, L);
if Key1dec == 0
    errordlg('Affine Key must be relatively prime with 95');
else
    %Invers Affine Key
    Key3dec = InvPrime(KeyAff, L);

    % Fibonacci Key
    Key2dec = Fibokey(KeyFibodec, N, L);

    %Affine Cipher
    Plaindec = mod(Key3dec*(Cipherdec - Key2dec), L) + 32;

    %Output
    Plaintext = char(Plaindec);
end
```

LAMPIRAN D. Skrip Program Kunci Fibonacci, Relatif Prima, Invers Modulo pada Matlab

b. Skrip Program Kunci Fibonacci

```
function Output = Fibokey(Key, N, L)
% Key = Key Input (Decimal form)
% N = Length output
% L = Maximum number
if nargin == 2
    L = 95;
end
Output = Key;
while length(Output) < N
    Output(end+1) = mod(Output(end) + Output(end-1), L);
end
```

c. Skrip Program Relatif Prima

```
function Output = RPrima(Number, L)
% Number = Input number
% L = Maximum number
if nargin == 1
    L = 95;
end
m = L;
n = mod(Number, L);
o = mod(m,n);
while abs (o)> 1
    m = n;
    n = o;
    o = mod(m,n);
end
Output = o;
```

d. Skrip Program Invers Modulo

```
function Output = InvPrime(Number, L)
% Number = Input number
% L = Maximum number
if nargin == 1
    L = 95;
end
for i = 1:L-1
    if mod(i*Number, L) == 1
        Output = i;
        break
    end
end
```

LAMPIRAN E. Skrip Program Koefisien Korelasi pada Matlab

```
function Hasil = AnalisKor(Ciphertext, Plaintext)
nrow = size(Ciphertext,1);
Cipherdec = [];
Plaindec = [];
for i = 1:nrow
    Cipherdec = [Cipherdec double(Ciphertext{i})];
    Plaindec = [Plaindec double(Plaintext{i})];
end
%mu
muc = sum(Cipherdec)/length(Cipherdec);
mup = sum(Plaindec)/length(Plaindec);
%sig
sigc = sqrt(sum((Cipherdec-muc).^2));
sigp = sqrt(sum((Plaindec-mup).^2));
%Korelasi]
Hasil = sum((Cipherdec-muc).*(Plaindec-mup))/(sigc*sigp);
```