**BAB I**

**PENDAHULUAN**

1. **Latar Belakang Masalah**

Matematika sebagai ilmu pengetahuan dasar memegang peranan yang sangat penting dalam perkembangan ilmu pengetahuan lain. Hal-hal yang dipelajari dalam matematika terdiri atas beberapa kelompok ilmu, seperti: aljabar, geometri, analisis, dan matematika terapan. Tugas akhir ini menjelaskan suatu analisis matematika yang diaplikasikan dalam matematika terapan, selanjutnnya dikembangkan suatu metode yang telah ada sebelumnya dibidang kriptografi.

Seiring dengan perkembangan zaman dan kemajuan teknologi yang sangat pesat, semakin banyak komputer yang terhubung dalam dunia maya yang biasa dikenal dengan sebutan internet. Tidak ketinggalan perusahaan, lembaga pemerintahan, lembaga keuangan, dan masih banyak lembaga lainnya yang juga turut berkecimpung dalam dunia maya ini. Banyak hal yang dapat dilakukan melalui internet untuk mempermudah hubungan antar lembaga diantaranya adalah penyampaian data dari satu pihak ke pihak yang lain dan pengiriman data yang dapat diakses oleh publik. Keamanan dalam proses pengiriman data sangat diperlukan. Kriptografi merupakan salah satu jawaban atas tuntutan tersebut. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhunbungan dengan aspek keamanan informasi seperti keabsahan, integritas data, serta autentikasi data. Kriptografi tidak berarti hanya memberikan keamanan informasi saja, namun lebih ke arah metode-metode yang digunakan.

1

Suatu pesan ketika ditransfer dari suatu tempat ke tempat lain, ada kemungkinan bahwa data tersebut dapat diambil atau bahkan dimodifikasi oleh pihak-pihak yang tidak diinginkan. Kriptografi sangat berperan dalam menjadikan pesan-pesan yang dikirim tersebut menjadi pesan yang tidak dapat dimengerti oleh pihak lain. Kriprografi disebut tangguh jika data yang dikirimkan akan tetap aman kendati setiap orang dapat mengaksesnya secara bebas.

Huruf yang sama pada pesan dalam dunia kriptografi mempunyai *image* huruf yang sama pula. Hal ini mempunyai tingkat resiko yang tinggi karena mudah ditebak. Oleh karena itu pesan haruslah disandikan (*encoding*). Tujuan membuat sandi adalah menjaga keamanan pesan dari para pembongkar sandi sehingga hanya penerima saja yang dapat mengetahui isinya.

Salah satu metode persandian dalam kriptografi adalah Sandi Hill. Sandi Hill merupakan algoritma kunci simetris yaitu algoritma yang menggunakan kunci yang sama pada proses enkripsi dan dekripsinya. Sandi Hill merupakan penerapan aritmatika modulo pada kriptografi. Teknik kriptogafi ini menggunakan sebuah matriks persegi sebagai kunci yang digunakan untuk melakukan enkripsi dan dekripsi serta menggunakan *m* buah persamaan linear. Enkripsi adalah proses dalam menyandikan pesan, sedangkan dekripsi adalah proses dalam menterjemahkan pesan yang telah disandikan. Dalam penerapannya, Sandi Hill menggunakan teknik perkalian matriks dan teknik invers terhadap matriks.

Pengembangan Sandi Hill diharapkan tidak terbatas pada matriks persegi sebagai kuncinya. Berdasar pada teori Matriks Invers Tergeneralisasi (MIT), Sandi Hill dapat diperluas agar memungkinkan untuk memakai matriks persegi panjang sebagai kunci.

1. **Rumusan Masalah**

Berdasarkan latar belakang di atas, rumusan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana membentuk invers tergeneralisasi dari sebuah matriks persegi panjang ?
2. Bagaimana konsep Matriks Invers Tergeneralisasi diterapkan pada Sandi Hill (*Cipher Hill*) ?
3. **Batasan Masalah**

Tugas akhir ini, dibatasi pada: penerapan matriks invers tergeneralisasi pada sandi hill berordo m x n dimana m ≠ n dalam *ZP*, m dan n adalah banyak baris dan kolom dalam matriks serta *P* adalah bilangan prima; Algoritma dan proses pembangkitan kunci (*enkripsi* dan *dekripsi*).

1. **Tujuan Penelitian**

Adapun tujuan penelitian ini adalah :

1. Memahami konsep Matriks Invers Tergeneralisasi dalam menemukan invers matriks persegi panjang.
2. Memahami metode Sandi Hill (*Cipher Hill*).
3. Memanfaatkan Matriks Invers Tergeneralisasi yang diterapkan pada metode Sandi Hill agar lebih handal.
4. Mengaplikasikan metode Sandi Hill yang diperluas pada program komputer (Pascal).
5. **Manfaat Penelitian**

Penelitian ini diharapkan dapat member manfaat sebagai berikut :

1. Memperdalam pemahaman mengenai teori Matriks Invers Tergeneralisasi dalam menemukan invers matriks persegi panjang.
2. Memahami ilmu sandi atau kriptografi dengan metode Sandi Hill.
3. Menghasilkan Sandi Hill yang lebih kuat dari serangan *hacker* (kriptoanalisis).
4. Menghasilkan program yang memudahkan pengguna dan penterjemah pesan.
5. **Kontribusi Penelitian**

Sandi hill merupakan salah satu metode dalam menyandikan data yang memanfaatkan metode perkalian matriks persegi yang invertibel dan metode ini hanya terbatas pada bilangan *Z26* yang tidak *field*. Dengan demikian, matriks yang dapat digunakan dalam metode ini lebih terbatas. Penelitian ini bermaksud mengembangkan metode sandi hill agar tidak terbatas pada matriks persegi yang invertibel saja, tetapi dapat pula digunakan pada matriks persegi panjang pada bilangan *ZP* yang *field.* Pengembangan metode sandi hill ini, sangat diharapkan dapat memberikan keamanan pesan yang lebih baik jika dibandingkan metode sandi hill pada umumnya.

1. **Kerangka Penulisan Tugas Akhir**

**Bab 1 Pendahuluan,** dalambab ini dijelaskan mengenai latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, kontribusi penelitian, dan kerangka penulisan tugas akhir.

**Bab 2 Kajian Pustaka,** dalam bab ini dijelaskan mengenai definisi, penjelasan, teorema, lemma, dan bukti yang mendasari pembahasan. Beberapa definisi yang dibahas adalah persamaan linear, matriks, vektor, ruang vektor, teori bilangan, kriptogarfi, dan sandi hill.

**Bab 3 Metodologi Penelitian,** dalam bab ini dijelaskan mengenai lokasi, waktu, fokus kajian, material pendukung, dan prosedur penelitian.

**Bab 4 Pembahasan,** dalambab ini dijelaskan mengenai teori matriks invers tergeneralisasi yang kemudian diterapkan pada sandi hill. Dalam bab ini pula akan dilakukan simulasi menggunakan bahasa pemrograman pascal.

**Bab 5 Penutup,** berisi tentang kesimpulan dan saran.

**BAB II**

**KAJIAN PUSTAKA**

1. **Sistem Persamaan Linear**

Variabel adalah sebuah notasi yang mewakili suatu bilangan dengan nilai yang belum diketahui. Sebagai contoh, sebuah garis dalam bidang *xy* secara aljabar dapat dinyatakan oleh persamaan yang berbentuk *a1x +a2y = b.* persamaan ini dinamakan persamaan linear dalam variabel *x* dan variabel *y*. secara lebih umum, mendefinisikan persamaan linear dalam n peubah *x1, x2, …, xn* sebagai persamaan yang dapat dinyatakan dalam persamaan (2.1):

(2.1)

dimana *a1, a2, …,an* dan b adalah konstanta-konstanta.

Solusi persamaan linear adalah urutan n bilangan *s1, s2, …, sn*sehingga suatu persamaan linear dipenuhi bila disubtitusikan . Himpunan dari semua solusi persamaan linear dinamakan himpunan solusi.

6

Menurut Anton, H (1987:4) bahwa sebuah sistem persamaan linear yang terdiri dari *m* persamaan linear dengan *n* bilangan tak diketahui (variabel) dapat ditulis pada persamaan (2.2):

(2.2)

untuk menuliskan sistem ini dalam bentuk persamaan perkalian matriks *Ax = b*, dimana diketahui, x adalah matriks *n* x 1 dari peubah-peubah, dan b suatu matriks *m* x 1 yang mewakili ruas kanan dari sistem. Hal ini ditetapkan sebagaimana pada persamaan (2.3)

(2.3)

dan ditetapkan hasil kali *Ax* sebagaimana pada persamaan (2.4)

(2.4)

yang selanjutnya jika disebut sistem *Ax = b* berarti ekivalen dengan menyebutkan sistem persamaan linear dengan *n* variabel dan *m* persamaan yang bisa direpresentasikan sebagai sistem persamaan perkalian matriks *Ax = b*. Jika semuanya nol maka sistem ini disebut sistem persamaan linear homogen. Jika terdapat maka disebut sistem persamaan linear tak homogen.

Jika sistem linear tidak memiliki penyelesaian, maka dikatakan bahwa sistem tersebut takkonsisten. Jika sistem linear mempunyai paling sedikit satu penyelesaian, maka dikatakan bahwa sistem tersebut konsisten (Leon, 1998).

Definisi 2.1 (Anton,1987)

Matriks *A* disebut dalam bentuk eselon baris tereduksi (*reduced row echelon form*) disingkat *rref* jika memenuhi empat kondisi berikut:

1. Jika ada baris nol (baris yang seluruh entrinya bernilai nol), maka baris nol tersebut terletak paling bawah atau paling akhir.
2. Jika baris tidak seluruhnya terdiri dari nol, maka entri pertama yang bukan nol dalam baris itu adalah 1, disebut satu utama (*leading one*).
3. Dalam sebarang dua baris yang berurutan yang tidak seluruhnya terdiri dari nol, maka satu utama dalam baris yang lebih rendah terdapat lebih jauh ke kanan dari satu utama dalam baris yang lebih tinggi.
4. Setiap kolom yang memuat satu utama mempunyai nol di tempat lain.

Sebuah matriks yang mempunyai sifat-sifat 1, 2, dan 3, dikatakan berada dalam bentuk eselon baris (*row-echelon form*).

Rank dari matriks *A* ditulis rank(*A*) adalah banyaknya baris tak nol setelah *A* dibentuk ke dalam bentuk eselon baris. Suatu matriks *Amxn* dikatakan mempunyai rank kolom penuh jika *rank*(*A*) = *n* dan rank baris penuhjika *rank*(*A*)= *m*.

1. **Matriks**
2. **Pengertian matriks**

Definisi 2.2 (Anton, 1987)

Sebuah matriks adalah susunan segi empat siku-siku dari bilangan-bilangan. Bilangan-bilangan dalam susunan tersebut dinamakan entri dalam matriks.

Elemen yang terletak pada baris *i* dan kolom *j* di dalam matriks dinyatakan sebagai *aij* dalam notasi matriks *A* = [*aij*] untuk *i* = 1, 2, …, m dan *j* = 1, 2, .., n. jadi matriks umumnya *m x n* sebgaimana persamaan (2.5)

(2.5)

Matris *A* dikatakan berukuran *m x n* dan unsur *aij* berada pada baris *i* dan kolom *j*. dua matriks dikatakan sama jika ukurannya sama dan mempunyai unsur yang sama di dalam setiap posisi.

Bilangan-bilangan *a11, a12, …, amn* yang menyusun rangkaian matriks pada definisi di atas disebut entri dalam matriks. Sedangkan ordo atau ukuran suatu matriks ditentukan oleh banyakya baris *m* dan *n* kolom, maka matriks *A* berordo *m x n*.

1. **Aljabar matriks**

Definisi 2.3 (Anton, 1987)

Jika *A* dan *B* adalah sebarang matriks yang ukurannya sama, maka jumlah *A + B* adalah matriks yang diperoleh dengan menambahkan bersama-sama entri yang bersesuaian dalam kedua matriks tersebut. Matriks yang ukurannya berbeda tidak dapat ditambahkan.

Contoh 2.1

Maka

Sedangkan *A + C* dan *B + C* tidak didefinisikan.▲

Definisi 2.4 (Anton, 1987)

Jika *A* adalah suatu matriks dan *c* adalah suatu skalar, maka hasil kali (*product*) *cA* adalah matriks yang diperoleh dengan mengalikan masing-masing entri dari *A* oleh *c*.

Contoh 2.2

*c* = 2

Maka

Definisi 2.5 (Leon, 1998)

Jika adalah matriks *m* x *n* dan adalah matriks *n* x *r*, maka hasil kali adalah matriks *m* x *r* yang entri-entrinya didefinisikan pada persamaan (2.6):

(2.6)

Contoh 2.3

Maka hasil kali *AB* diberikan berikut:

*AB11* = 2.1 + 1.2 = 4

*AB12* = 2.2 + 1.(-1) = 3

*AB13* = 2.0 + 1.1 = 1

*AB21* = 3.1 + 0.2 = 3

*AB22* = 3.2 + 0.(-1) = 6

*AB23* = 3.0 + 0.1 = 0

Sifat 2.1 (Anton, 1987)

Secara umum perkalian matriks tidak bersifat komutatif, yaitu tidak selalu berlaku *AB=BA.*

Contoh 2.4

Jika kalikannya akan dihasilkan

Jadi *AB ≠ BA ▲*

Teorema 2.1 (Leon, 1998)

Apabila persaman (2.7) sampai (2.15) adalah sahih untuk setiap skalar α dan β dan untuk setiap matriks *A, B,* dan *C* maka operasi-opereasi yang bersangkutan terdefinisi.

1. A+ B = B + A (2.7)
2. (A + B) + C = A + (B + C) (2.8)
3. (AB)C = A(BC) (2.9)
4. A(B + C) = AB + AC (2.10)
5. (A + B)C = AC + BC (2.11)
6. (αβ)A = α(βA) (2.12)
7. α(AB) = (αA)B = A(αB) (2.13)
8. (α + β)A = αA + βA (2.14)
9. α(A + B) = αA + αB (2.15)

Bukti dari persamaan (2.9)

Misalkan *A* matriks *m* x *n*, *B* matriks *n* x *r*, dan C matriks *r* x *s*. misal *D = AB* dan *E = BC*. Harus ditunjukkan bahwa *DC = AE*. Berdasarkan definisi 2.5, dituliskan persamaan (2.16)

(2.16)

Entri ke-*ij* dari *DC* adalah sebagaimana pada persamaan (2.17)

(2.17)

dan entri ke-*ij* dari *AE* adalah seperti ditunjukkan pada persamaan (2.18)

(2.18)

Karena berdasarkan persamaan (2.19)

(2.19)

Maka dihasilkan persamaan (2.20)

▲ (2.20)

Contoh 2.5

Sebagai gambaran teorema 2.1 untuk persamaan (2.9) , dimisalkan

Kemudian

Sehingga

Sebaliknya

Maka

Jadi *A(BC) = (AB)C,* seperti yang ditunjukan oleh teorema 2.1 untuk persamaan (2.9). ▲

1. **Jenis-jenis matriks**
2. **Matriks persegi panjang**

Matriks persegi panjang adalah sebuah matriks yang berbentuk persegi panjang dengan elemennya terletak pada baris *i* dan kolom *j* untuk *i* = 1, 2, …, m dan *j* = 1, 2, …, n dengan notasi matriks *A* = [*aij*]. jadi matriks umumnya *m x n* sebagaimana pada persamaan (2.21)

(2.21)

1. **Matriks persegi**

Matriks persegi adalah suatu matriks yang banyaknya baris sama dengan banyaknya kolom, yang dinyatakan dengan *Amxn*, dimana *m = n*, dapat ditulis *Anxn*= *(aij)nxn*. Matriks berordo *n x n* disebut juga matriks persegi ordo *n* sebagaimana pada persamaan (2.22)

(2.22)

Elemen-elemen matriks persegi disebut element diagonal utama, sedangkan disebut element diagonal kedua. Dalam hal ini hanya matriks persegi yang mempunyai elemen diagonal utama dan diagonal kedua.

1. **Matriks nol**

Sebuah matriks yang semua entrinya sama dengan nol dinamakan matriks nol (*zero matrix*). Matriks nol akan dinyatakan oleh 0; jika ukurannya penting untuk ditekankan, maka kita akan menuliskan 0*mxn* untuk matriks nol *m x n*. (Anton, 1987)

Menurut Anton (1987:32) bahwa beberapa aturan ilmu hitung untuk bilangan riil tidak dapat digunakan terhadap ilmu hitung matriks, maka akan merupakan anggapan yang gegabah bahwa semua sifat bilangan riil nol akan dapat digunakan terhadap matriks nol. Misalnya, tinjaulah kedua hasil berikut dalam ilmu hitung bilangan riil.

1. Jika *ab=ac* dan *a ≠* 0, maka b = c. (ini dinamakan *hukum peniadaan*)
2. Jika *ad =* 0, maka setidak-tidaknya satu dari faktor disebelah kiri sama dengan 0.

Seperti yang diperlihat, ternyata palsu dalam ilmu hitung matriks.

1. **Matriks diagonal**

Matriks diagonal merupakan matriks persegi dengan semua elemen-elemen yang bukan elemen diagonal utama adalah nol. Dengan kata lain suatu matriks *A* berordo *n x n* disebut matrks diagonal jika *aij* = 0 untuk *i ≠ j* sebagaimana pada persamaan (2.23).

(2.23)

1. **Matriks identitas**

Matriks identitas (*identity matrix*) adalah matriks skalar dengan elemen-elemen diagonal utama semuanya sama dengan satu.

Definisi (Leon, 1998)

Matriks identitas adalah matriks *I* = (δ*ij*) berorde *n x n*, dimana ditunjukkan pada persamaan (2.24)

(2.24)

1. **Matriks taksingular**

Matriks taksingular adalah matriks yang determinannya tidak sama dengan nol.

Contoh 2.6

Maka determinannya adalah

Jadi *A* merupakan matriks non-singular karena determinannya tidak sama dengan nol.

1. **Matriks singular**

Matriks singular adalah matriks yang determinannya sama dengan nol. Jika determinan dari matriks persegi lenyap, yaitu baris (atau kolom) adalah bergantung linear, maka matriks dikatakan singular.

Contoh 2.7

Maka

1. **Matriks skalar**

Matriks skalar adalah matriks diagonal dengan semua elemen-elemen diagonal utamanya sama dengan *k*, dimana *k* ≠ 0. Bentuk umum matriks skalar adalah untuk *i* = *j* dan untuk *i* ≠ *j* sebagaimana pada persamaan (2.25).

(2.25)

1. **Matriks segitiga**

Suatu matriks *A* berorde *n* x *n* disebut matriks segitiga atas (*upper triangular*) jika *aij =* 0 untuk *i > j* dan matriks segitiga bawah (*lower triangular*) jika *aij =* 0 untuk *i < j*. *A* disebut juga matriks segitiga (*triangular*) jika *A* matriks segitiga atas atau matriks segitiga bawah. Sebagai contoh, matriks-matriks 3 x 3 :

(a) (b)

Gambar 2.1: Matriks segitiga 3 x 3 (a) matriks segitiga atas (b) Matriks segitiga bawah.

1. **Transpos, transpos konjugat, dan hermitian**

Definisi 2.6 (Leon, 1998)

Transpos dari suatu matriks *A* berorde *m* x *n* adalah matriks *B* berorde *n*  x *m* yang didefinisikan oleh untuk j dan . transpos dari *A* dinyatakan oleh *AT*.

Contoh 2.7

Jika , maka

Konjugat dari *A,* ditulis , merupakan matriks yang dibentuk dengan menegasikan bagian imajiner setiap entri *A,* jadi . Transpos konjugat dari *A* didefinisikan oleh *AH* = *T.* Matriks *A* dikatakan hermitian jika *AH* = *A.*

Contoh 2.8

maka,

Jika *A* riil maka *A* dikatakan matriks simetri, yaitu *AT = A*. Dapat dilihat bahwa setiap matriks hermitian haruslah persegi. Teorema berikut mengenai sifat-sifat transpos.

Teorema 2.2 (Anton, 1987) :

Jika ukuran matriks seperti operasi yang diberikan dapat dilakukan, maka

1. (2.26)
2. (2.27)
3. , dimana α adalah sebarang skalar. (2.28)
4. (2.29)

Menurut Wijna (2009) suatu matriks persegi dikatakan ortogonal jika dan hanya jika dan suatu matriks persegi dikatakan uniter jika dan hanya jika .

1. **Invers matriks**

Definisi 2.7 (Anton, 1987)

Jika *A* adalah matriks persegi, dan jika kita dapat mencari matriks *B* sehingga *AB = BA = I*, maka *A* dikatakan dapat dibalik (*invertibel*) dan *B* dinamakan invers (inverse) dari *A,* ditulis *B = A-1*.

Jika *A* adalah sebarang matriks persegi *n* x *n* dan *cij* adalah kofaktor *aij*, maka matriks dituliskan pada persamaan (2.30)

(2.30)

dinamakan matriks kofaktor *A*. Transpos matriks ini dinamakan *adjoin* *A* dan dinotasikan dengan Adj(*A*). Jika *A* adalah matriks yang dapat dibalik, maka dituliskan persamaan (2.31)

(2.31)

Dari persamaan diatas, dapat disimpulkan jika suatu matriks mampunyai nilai determinan nol maka matriks tersebut tidak punya invers. Sebab jika det(*A*) = 0 maka terjadi pembagian dengan nol.

Sebuah matriks persegi *U* disebut matriks uniter jika berlaku sifat *UHU = UUH = I* . jika *U* riil maka dikatakan *U* ortogonal.

1. **Ruang Vektor**

Definisi 2.8 (Leon, 1998):

Misalkan *V* adalah himpunan dimana didefinisikan operasi-operasi penjumlahan dan perkalian dengan skalar. Dengan ini kita mengartikannya bahwa untuk setiap pasang elemen-elemen x dan y di dalam *V*, kita dapat mengasosiasikannya dengan elemen x + y yang tunggal yang juga berada di *V*, dan dengan setiap elemen x di *V* dan setiap skalar α, kita dapat mengasosiasikannya dengan elemen αx yang tuggal di *V*.

Definisi 2.9 (Anton, 1987):

Jika adalah himpunan vektor, maka persamaan vektor mempunyai paling sedikit satu pemecahan, yakni

Jika ini adalah satu-satunya pemecahan, maka *S* kita namakan himpunan bebas linear(*linearly independent*). Jika ada pemecahan lain, maka *S* kita namakan himpunan tak-bebas linear(*linearly dependent*).

Definisi 2.10 (Anton, 1987)

Jika *V* adalah sebarang ruang vektor dan merupakan himpunan berhingga dari vektor-vektor pada *V,* maka *S* kita namakan basis untuk *V* jika *S* bebas linear dan *S* merentang *V*.

Berikut ini akan diberikan definisi mengenai basis ortonormal. Sebelumnya, akan didefinisikan dulu mengenai vektor ortogonal dan vektor normal yang diberikan pada vektor-vektor di . Karena definisi tersebut juga berlaku pada karena adanya hubungan .

Menurut Wijna (2009:8) vektor dikatakan saling ortogonal jika dan hanya jika dan vektor dikatakan vektor normal jika dan hanya jika .

Definsi 2.11 (Wijna, 2009)

Diketahui *V* ruang vektor atas *C,* dan merupakan basis untuk *V*. Basis *S* disebut basis ortonormal jika dan hanya jika:

1. Setiap merupakan vektor normal, dan
2. untuk setiap *i, j* dengan *i ≠ j*.
3. **Ruang Hasil Kali Dalam**

Menurut Leon (1998) dua vektor x dan y di dalam *Rn*dapat dipandang sebagai matriks *n* x 1. Kemudian kita dapat membentuk hasil kali matriks *xTy.* hasil kali ini adalah matriks 1 x 1 yang dapat dipandang sebagai sebuah vektor di dalam *R1* atau lebih sederhananya sebagai sebuah bilangan riil. Hasil kali *xTy* disebut *hasil kali skalar* dari x dan y. Secara khusus, jika dan , maka dituliskan persamaan (2.32)

(2.32)

Definisi 2.12 (Anton, 1987):

Sebuah hasil kali dalam (*inner product*) pada ruang vektor riil *V* adalah fungsi yang mengasosiasikan bilangan riil (u,v) dengan masing-masing pasangan vektor u dan *v* pada *V* sedemikian rupa sehingga aksioma-aksioma berikut dipenuhi untuk semua vektor *u,v*, dan *w* di *V* dan juga untuk semua skalar *k*.

1. (*u,v*) = (*v,u*) (aksioma simetri) (2.33)
2. (*u + v, w*) = (*u,w*) + (*v,w*) (aksioma penambahan) (2.34)
3. (*ku,v*) = *k*(*u,v*) (aksioma kehomogenan) (2.35)
4. (*v,v*) ≥ 0; dan (*v,v*) = 0 (aksioma kepositifan) (2.36) jika dan hanya jika *v* = 0

Sebuah ruang vektor riil dengan sebuah hasil kali dalam dinamakan ***ruang hasil kali dalam riil* (*riil product space*)**.

1. **Nilai Eigen dan Vektor Eigen**

Definisi 2.13 (Anton, 1987)

Jika *A* adalah matriks persegi *n* x *n,* maka vektor taknol *x* di dalam *Rn* dinamakan vektor eigen dari *A* jika *Ax* adalah kelipatan skalar dari *x,* yaitu untuk suatu skalar λ. Skalar λ dinamakan nilai eigen dari *A* dan *x* dikatakan vektor eigen yang bersesuaian dengan λ.

Teorema 2.3 (Leon, 1998)

Misalkan *A* adalah matriks *n* x *n* dan λ adalah suatu skalar. Pernyataan-pernyataan berikut ini adalah ekivalen.

1. λ adalah nilai eigen dari *A.*
2. (*A –* λ*I*)x = 0 mempunyai penyelesaian taktrivial.
3. *N*(*A –* λ*I*)x ≠ {0}.
4. *A –* λ*I*  adalah singular.
5. *det* (*A –* λ*I*) = 0.

Contoh 2.7

Tentukan nilai eigen dan vektor eigen dari matriks berikut ini

Penyelesaian:

Untuk mencari nilai eigen, digunakan persamaan karakteristik dari A yaitu teorema 2.3.5.

Jadi nilai eigennya adalah Penyelesaian dari yaitu dan .

adalah vektor eigen dari matriks A yang terkait dengan λ, jika dan hanya jika λ adalah sebuah solusi tak trivial dari (*A – λI*)*x = 0*, yaitu:

Jika , maka diperoleh :

Penyelesaian vektor eigen pertama untuk adalah

Jika , maka diperoleh:

Penyelesaian vektor eigen kedua untuk adalah

1. **Dekomposisi Nilai Sigular**

Dekomposisi nilai singular atau *Singular Value Decomposition* yang selanjutnya ditulis dengan SVD adalah suatu teknik yang digunakan secara luas untuk mendekomposisikan suatu matriks ke dalam beberapa komponen matriks yang berkaitan erat dengan nilai singular dari matriksnya. Proses dekomposisi ini juga disebut dengan faktorisasi.

Dalam dekomposisi nilai singular, suatu matriks difaktorkan menjadi tiga bagian, dimana salah satu dari ketiga matriks tersebut entrinya merupakan nilai singular dari matriksnya.

Definisi 2.14 (Wijna, 2009)

Diketahui matriks dengan *rank A = r.* Diketahui pula nilai eigen dari matriks *AHA* adalah sebagaimana pada persamaan (2.37)

(2.37)

Bilangan , untuk setiap 1 ≤ *i* ≤ *n* disebut nilai singular dari matriks *A*.

Teorema 2.4 (Wijna, 2009)

Diketahui matriks dengan *rank A = r.* maka terdapat tepat sejumlah *r* nilai singular yang tak nol.

Bukti:

Misalkan nilai eigen dari matriks *AHA* adalah dengan . Berarti terdapat sejumlah *n* vektor eigen *xi* , *i =* 1, 2, 3, …, *n* yang bersesuaian dengan nilai-nilai eigen tersebut. Misalkan merupakan himpunan *n* vektor eigen yang dimaksud. Diperlihatkan bahwa himpunan tersebut membentuk basis ortogonal untuk *Cn* . Jika basis ortogonal tersebut dinormalisasi akan diperoleh basis ortonormal untuk *Cn* , dengan ditunjukkan pada persamaan (2.38)

untuk setiap 1 ≤ *i* ≤ *n* . (2.38)

Karena himpunan merupakan basis ortonormal untuk *Cn* , maka berlaku untuk setiap 1 ≤ *i* ≤ *n,* akibatnya dituliskan sebagaimana pada persamaan (2.39)

(2.39)

Menurut definisi 2.14, berlaku , untuk setiap 1 ≤ *i* ≤ *n.* Karena diketahui rank *A* = *r*, maka berlaku , dan . Jadi diperoleh , untuk *i* = 1,2,3,…,*r*.

Teorema 2.5 (Yanai, 2011)

(2.40)

Dimana,

(2.41)

Perhitungan SVD meliputi pencarian nilai eigen dan vektor eigen dari *AHA.* Kolom-kolom matriks singular *V* berisikan vektor eigen dari *AHA*, kolom-kolom matriks singular *U* diperoleh dari persamaan (2.42)

(2.42)

dan berdasar definisi 2.14, matriks singular *S* diperoleh dari persamaan (2.43)

(2.43)

1. **Teori Bilangan**

Teori bilangan (*number theory*) adalah teori dasar dalam memahami kriptografi. Bilangan yang digunakan disini adalah bilangan bulat porsitif (*integer*). Bilangan bulat positif adalah himpunan bilangan asli yang dinotasikan dengan “*A*” yaitu *A* = {1, 2, 3, …}. Himpunan semua bilangan bulat yang dinotasikan dengan *Z* adalah himpunan {…,-3, -2, -1, 0, 1, 2, 3, …}. Himpunan ini berperan sangat penting karena banyak algoritma kriptografi yang menggunakan sifat-sifat himpunan semua bilangan bulat dalam melakukan prosesnya. Pada himpunan bilangan bulat berlaku sifat asosiatif, komutatif dan distributif terhadap operasi penjumlahan dan pergandaan biasa. (Murtiyasa, 2001)

Dalam hitungan modulus, diberikan bilangan bulat positif *m* yang disebut modulus dan dua bilangan bulat sebarang yang selisihnya adalah kelipatan bulat. Modulus itu dipandang sebagai “sama” atau “setara” terhadap modulus. (Tiro, 2008)

Definisi 2.15 (Anton, 1987):

Jika *m* bilangan bulat positif dan *a* dan *b* bilangan bulat sebarang, maka dikatakan bahwa *a* setara *b* modulo *m*. ditulis sebagai *a = b* (mod *m*) jika *a-b*  adalah bilangan bulat kelipatan *m*.

Contoh 2.8

7 = 2 (mod 5)

19 = 3 (mod 2)

-1 = 25 (mod 26)

12 = 0 (mod 4)

Jika *a* bilangan bulat tak negatif, maka modulo sisanya hanyalah sisa yang dihasilkan bilamana *a* dibagi *m*.

Teorema 2.6 (Anton, 1987):

Untuk bilangan bulat *a* sebarang dan modulo *m*, seandainya

Maka sisa r dari modulo *m* diberikan oleh

Definisi 2.16 (Tiro, 2008)

Jika dan , maka *a* disebut bilangan yang dibagi (*dividend*), *b* disebut bilangan pembagi (*divisor*), q disebut bilangan hasil bagi (*quationt*) dan r sisebut bilangan sisa pembagian (*remainder*).

Diberikan , m > 0 membagi (*b* – *a*), maka *a* disebut kongruen dengan *b* modulo *m,* ditulis *a*  *b* (mod *m*). bilangan bulat *m* disebut modulus. Misalkan *a* dan *b* dibagi dengan *m*, didapatkan hasil bagi bilangan bulat dan sisa, dimana sisa bernilai antara 0 dan *m* – 1. Yaitu dan , dimana dan . Maka jelas bahwa *a*  *b* (mod *m*) jika dan hanya jika .

Definisi 2.17 (Anton, 1987)

Jika *a* adalah suatu bilangan dalam *Zm*, maka bilangan *a*-1 dalam *Zm* disebut balikan atau invers perkalian dari *a* modulo *m* jika *aa*-1 = *a*-1*a* = 1 (mod *a*).

Dalam *Zp* dengan *p* prima, suatu bilangan *a* mempunyai invers terhadap pergandaaan jika dan hanya jika ppt(*a,p*)=1. Bilangan tersebut adalah semua elemen dalam *Zp* kecuali 0. Matriks *A* atas *Zp* mempunyai invers modulo *p* jika dan hanya jika ppt(*det A,p*)=1.

1. **Algoritma Euclid**

Algoritma Euclid dirumuskan sebagai berikut : Misalkan akan dicari pembagi persekutuan terbesar dari bilangan bulat a dan b. karena ppt ( | *r0*|,| *r1*| ) = ppt (*r0* , *r1*) dan misalkan *r0* ≥ *r1* > 0. Langkah pertama menerapkan algoritma pembagian terhadap a dan b (definisi 2.16) diperoleh

r0 = q1 r1 + r2 0 ≤ r2 < r1

Jika terjadi r2 = 0, maka b|ap dan ppt (r0,r1) = r1. jika r2 ≠ 0, bagilah r1 oleh r2 dan diperoleh q2 dan r3 yang memenuhi

r1 = q2r2 + r3 0 ≤ r3 < r2

Jika r3 = 0, maka berhenti, sebaliknya jika r3 ≠ 0 dengan cara yang sama diperoleh

r2 = q3r3 + r4 0 ≤ r4 < r3

Proses pembagian ini dilanjutkan sampai sisa pembagian nol, katakanlah pada langkah ke (n+1) yang mana rn-1 dibagi rn  dengan r1 > r2 > r3 >…≥ 0.

Proses diatas menghasilkan sistem persamaan (2.44) berikut:

r0 = q1 r1 + r2 0 ≤ r2 < r1

r1 = q2r2 + r3 0 ≤ r3 < r2

r2 = q3r3 + r4 0 ≤ r4 < r3

.

. (2.44)

.

rn-2 = qn-1 rn-1 + rn 0 ≤ rn < rn-1

rn-1 = qn rn + 0

Sisa pembagian yang terakhir yang bukan nol rn = ppt (r0, r1).

Lemma 2.1 (Mulyana, 2002)

Jika a = qb + r, maka ppt (a,b) = ppt (b,r)

Bukti:

Jika d = ppt (a,b) maka d|a dan d|b mengakibatkan d|(a-bq) atau d|r. jadi d pembagi persekutuan dari b dan r. Dilain pihak jika c sebarang pembagi persekutuan dari b dan r, maka c|(qb+r) atau c|a. ini mengakibatkan c merupakan pembagi persekutuan dari a dan b dengan c ≤ d. Berdasarkan definisi pembagi persekutuan terbesar d = ppt (a,b) = ppt (b,r).

Berdasarkan lemma ini, dari system persamaan diatas diperoleh ppt (a,b) = ppt (b,r1) = ppt (r1,r2) = …= ppt (rn-1,rn) = ppt (rn,0) = rn

Pada teorema sebelumnya menyatakan bahwa ppt (a,b) dapat dinyatakan sebagai ax + by. Untuk menentukan x dan y yang memenuhi ppt (a,b) = ax + by adalah sama dengan subtitusi balik algoritma Euclid ini.

rn = rn-2 – qn-1rn-1

rn = rn-2 – qn-1 (rn-3 – qn-2rn-2)

= (1 + qn-1 qn-2) rn-2 + (-qn-1 )rn-3 (2.44)

Representasi m sebagai kombinasi linear dari rn-1 dan rn-3. Dengan meneruskan subtitusi balik dari sistem persamaan tersebut, kita akan berhasil mengeliminasi rn-1, rn-2, …,r2, r1 sehingga rn = ppt (a,b) dinyatakan sebagai kombinasi linear dari a dan b.

1. **Algoritma Euclid Yang Diperluas**

Dari algoritma Euclide dapat diketahui apakah suatu bilangan mempunyai invers atas *Zp*atau tidak, namun belum dapat menghitung nilai inversnya (jika ada). Dengan algoritma Euclide yang diperluas, dapat dihitung nilai invers dari suatu bilangan.

Misalkan didefinisikan suatu barisan bilangan dengan ketentuan sebagai berikut :

(2.46)

(2.47)

(mod ) *j* ≥ 2 (2.48)

maka, untuk 0 ≤ *j* ≤ *m*, (mod ). Dimana didapatkan dari algoritma Euclid.

Pembuktian dilakukan dengan induksi matematika. Pernyataan benar untuk *j* = 0 dan *j* = 1. Asumsi pernyataan benar untuk *j = i –* 1 dan *j = i –* 2, untuk *i* ≥ 2, akan dibuktikan pernyataan benar untuk *j = i*. Dengan induksi dihasilkan persamaan (2.49):

(2.49)

Selanjutnya dihitung :

(2.50)

Dapat disimpulkan pernyataan terbukti untuk semua *j*.

Misalkan ppt() = 1, maka mod , dapat dibuktikan jika ppt() = 1 maka . Sehingga dari teorema di atas, (mod ), dengan melihat bentuk : , berarti mod .

1. **Struktur Aljabar**

Definisi 2.18 (Tahmir, 2004)

Suatu aljabar (G,\*) dari himpunan tidak kosong G dengan operasi biner \*, dikatakan grup jika memenuhi sifat berikut.

1. berlaku *a\**(*b\*c*) = (*a\*b*)*\*c*. (sifat asosiatif)
2. Ada e G sehingga berlaku *a\**e = e*\*a* = *a.* (adanya unsure identitas)
3. , sehingga *a\*a-1* = a*-1\*a* = e. (adanya unsur invers setiap anggota di G)

Selanjutnya jika (G,\*) merupakan **Grup** dan memenuhi sifat komutatif yaitu *a\*b* = *b\*a*  maka G disebut **grup komutatif**.

Definisi 2.19 (Tahmir, 2005)

Suatu struktur aljabar dengan dua operasi biner (R, +, **.**) dinamakan **Ring** jika memenuhi:

1. (R, +) merupakan grup komutatif
2. (R, **.**) merupakan semigrup
3. berlaku :
4. *a****.***(*b + c*) *= a****.****b + a****.****c* (2.51)
5. (*a + b*) ***.****c = a****.****c + b****.****c* (2.52)

Definisi 2.20 (Tahmir, 2005)

Suatu ring (R, +, **.**) disebut ring komutatif, jika (R, **.**) komutatif, dan disebut ring unsur kesatuan jika ada *a* ϵ R sehingga *a****.****b = b****.****a = b* .

Definisi 2.21 (Tahmir, 2005)

Suatu ring *R* dinamakan ring pembagian (*Division Ring*) jika memenuhi :

1. |*R*| ≥ 2 (|*R*| = banyaknya anggota di *R*)
2. *R* memiliki unsur kesatuan (sebut 1)
3. untuk setiap dengan *x* ≠ 0, ada sehingga xy = yx = 1

suatu gelanggang yang bersifat komutatif dimana setiap elemen tidak nol mempunyai invers perkalian disebut dengan lapangan (*field*). Yang dimaksud dengan invers perkalian adalah untuk setiap a ≠ 0 yang termasuk dalam *F*, terdapat sedemikian hingga .

Definisi 2.22 (Tahmir, 2005)

Ring pembagian yang komutatif disebut lapangan (*Field*).

1. **Matriks Invers Tergeneralisasi**

Ide awal dari invers matriks tergeneralisasi (*Generalized Inverses of Matrix*) adalah untuk menggeneralisasi pengertian invers matriks. Selanjutnya konsep dari invers matriks tergeneralisasi diberikan dalam definisi berikut ini.

Definisi 2.23 (Yanai, 2011)

Diberikan matriks A berukuran *n* x *m* dan misalkan bahwa . Jika solusi persamaan linear dapat ditunjukkan sebagai , matriks berukuran *m* x *n* disebut matriks invers tergeneralisasi (*g-invers*) dari matriks A.

Definisi 2.24 (Yanai, 2011)

Diberikan A matriks *m* × *n* atas *field*, suatu matriks *B* yang memenuhi sifat:

1. *BAB = B* (2.53)

2. *ABA = A* (2.54)

3. (*BA)H = BA* (2.55)

4. (*AB)H = AB* (2.56)

matriks *B* disebut *pseudo-invers* atau *p-invers* atau invers matriks tergeneralisasi (*Generalized Inverses of Matrix*) dari *A*, dinotasikan dengan .

Pada kriptografi klasik data-data yang berupa huruf dikonversikan ke dalam sistem *Z26*. Sedangkan pada tulisan ini dikembangkan khusus pada *Zp*, dengan *p* bilangan prima. Penulis memilih *Zp* sebab *Zp* adalah *field*, sehingga teori-teori yang telah dibahas sebelumnya berlaku. Untuk selanjutnya pembahasan invers matriks tergeneralisasi adalah untuk matriks yang entri-entrinya atas himpunan *Zp*.

Teorema 2.6 (Yanai, 2011)

Jika ,

(2.57)

dan jika

(2.58)

Dalam penghitungan invers tergeneralisasi dibutuhkan *A* mempunyai rank kolom penuh, maka . Dapat dilihat bahwa *AH A invertible*, sehingga (*AH A*)-1 *AH*.

Hal diatas, akan menjadi landasan dari aplikasi invers matriks tergeneralisasi pada algoritma Hill Cipher.

1. **Kriptografi**

Kriptografi berasal dari Bahasa Yunani: “cryptós” artinya rahasia, sedangkan “gráphein” artinya tulisan. Jadi, secara morfologi kriptografi berarti tulisan rahasia. Secara umum kriptografi dapat diartikan sebagai ilmu dan seni penyandian yang bertujuan untuk menjaga keamanan dan kerahasiaan suatu pesan. Dengan adanya tulisan yang tersembunyi ini, orang-orang yang tidak mengetahui bagaimana tulisan tersebut disembunyikan tidak akan mengetahui bagaimana cara membaca maupun menterjemahkan tulisan tersebut. Kriptografi pada dasarnya sudah dikenal sejak lama. Menurut catatan sejarah, kriptografi sudah digunakan oleh bangsa Mesir sejak 4000 tahun yang lalu oleh raja-raja Mesir pada saat perang untuk mengirimkan pesan rahasia kepada panglima perangnya melalui kurir-kurinya. Orang yang melakukan penyandian ini disebut *kriptografer*, sedangkan orang yang mendalami ilmu dan seni dalam membuka atau memecahkan suatu algoritma kriptografi tanpa harus mengetahui kuncinya disebut *kriptoanalis*.Pada perkembangan selanjutnya, kriptografi berkembang menjadi sebuah disiplin ilmu sendiri karena teknikteknik kriptografi dapat diformulasikan secara matematik sehingga menjadi sebuah metode yang formal.

Menurut Hans Delfs dan Helmut Knebl (2006), Menyediakan kerahasiaan adalah bukan hanya tujuan kriptografi. Kriptografi juga digunakan untuk menyediakan solusi untuk masalah lain:

1. *Kerahasiaan*. Pesan (*plainteks*) hanya dapat dibaca oleh pihak yang memliki kewenangan.
2. *Integritas data*. Penerima pesan harus dapat memeriksa apakah pesan itu dimodifikasi selama transmisi, baik sengaja atau tidak disengaja. Tidak seorang pun harus mampu menggantikan pesan asli dengan pesan palsu atau sebagian dari pesan itu.
3. *Autentikasi*. Penerima pesan harus dapat memverifikasi asal-usulnya. Tidak seorang pun harus mampu mengirim pesan ke Bob dan berpura-pura menjadi Alice (data otentikasi asal). Saat melakukan komunikasi, Alice dan Bob harus dapat mengidentifikasi satu sama lain (otentikasi kesatuan).
4. *Non-repudiation*. Pengirim seharusnya tidak dapat menyangkal kemudian bahwa dia mengirimkan pesan.

Tujuan utama dari kriptografi adalah untuk menjaga kerahasiaan *plainteks* dari pembaca rahasia untuk mencoba mendapatkan beberapa informasi tentang *plainteks*. Seperti dibahas sebelumnya, musuh juga dapat aktif dan mencoba untuk memodifikasi pesan. Kemudian, kriptografi diharapkan untuk menjamin integritas pesan. Musuh yang diasumsikan memiliki akses lengkap untuk saluran komunikasi.

Kriptografi pada dasarnya terdiri dari dua proses, yaitu proses enkripsi dan proses dekripsi. Proses enkripsi adalah proses penyandian pesan terbuka (*plainteks*) menjadi pesan rahasia (*cipherteks*). *Cipherteks* inilah yang nantinya akan dikirimkan melalui saluran komunikasi terbuka. Pada saat *cipherteks* diterima oleh penerima pesan, maka pesan rahasia tersebut diubah lagi menjadi pesan terbuka melalui proses dekripsi sehingga pesan tadi dapat dibaca kembali oleh penerima pesan. Secara umum, proses enkripsi dan dekripsi dapat dilihat pada gambar 2.2:

Enkripsi

Dekripsi

Plainteks

Plainteks

Cipherteks

Kunci

Kunci

Gambar 2.2 : proses enkripsi dan dekripsi

Definisi 2.25 (Stinson, 2006)

*Cryptosystem* (sistem kriptografi) adalah suatu 5-tuple , dimana memenuhi kondisi sebagai berikut :

1. adalah himpunan berhingga *plainteks*
2. adalah himpunan berhingsga *cipherteks*
3. adalah himpunan berhingga kunci
4. Untuk setiap terdapat dan . Setiap dan merupakan fungsi sedemikian sehingga untuk setiap *plainteks* .

Dasar matematis yang mendasari proses enkripsi dan deskripsi adalah relasi antara dua himpunan yaitu yang berisi elemen *plainteks* dan yang berisi elemen cipertext. Enkripsi dan dekripsi merupakan fungsi transformasi antara himpunan-himpunan tersebut. Apabila elemen-elemen *plainteks* dinotasikan dengan P, elemen-elemen *cipherteks* dinotasikan dengan C, sedang untuk proses enkripsi dinotasikan dengan E, dekripsi dengan notasi D, maka secara matematis proses kriptografi dapat dinyatakan sebagai berikut :

Enkripsi : *E(P) = C*

Dekripsi : *D(C) = P* atau *D(E(P)) = P*

Pada skema enkripsi konvensional atau *symmetric-key* digunakan sebuah kunci untuk melakukan proses enkripsi dan dekripsinya. Kunci tersebut dinotasikan dengan K, sehingga proses kriptografinya adalah :

Enkripsi : *EK(P) = C*

Dekripsi : *DK(C) = P* atau *DK(EK(P)) = P*

Sedangkan pada sistem *asymmetric-key* digunakan kunci umum (public key) untuk enkripsi dan kunci pribadi (private key) untuk proses dekripsinya sehingga kedua proses tersebut dapat dinyatakan sebagai berikut :

Enkripsi : *EPK(P) = C*

Dekripsi : *DSK(C) = P* atau *DSK(EPK(P)) = P*

Pada tugas akhir ini, dibahas dan dikembangkan salah satu metode dari *symmetric-key* yaitu Sandi Hill dan memandang kepada sisi matematikanya.

1. **Sandi Hill**

Pada tahun 1929 *block-cipher* yang disebut dengan Sandi Hill diperkenalkan oleh Lester S. Hill. Teknik kriptografi ini diciptakan dengan maksud untuk dapat menciptakan cipher (kode) yang tidak dapat dipecahkan menggunakan teknik analisis frekuensi. Sandi Hill tidak mengganti setiap abjad yang sama pada *plainteks* dengan abjad lainnya yang sama pada cipherteks karena menggunakan perkalian matriks pada dasar enkripsi dan dekripsinya.

Sandi Hill dikategorikan sebagai block cipher karena teks yang akan diproses dibagi menjadi blok-blok dengan ukuran tertentu. Setiap karakter dalam satu blok akan saling mempengaruhi karakter lainnya dalam proses enkripsi dan dekripsinya, sehingga karakter yang sama tidak dipetakan menjadi karakter yang sama pula. Sandi Hill termasuk algoritma kriptografi klasik yang sangat sulit dipecahkan oleh kriptanalis apabila dilakukan hanya dengan mengetahui berkas cipherteks saja. Namun, teknik ini dapat dipecahkan dengan cukup mudah apabila kriptanalis memiliki berkas *cipherteks* dan potongan berkas *plainteks*. Teknik kriptanalisis ini disebut known-plaintext attack

Sandi Hill merupakan penerapan aritmatika modulo pada kriptografi. Teknik kriptogafi ini menggunakan sebuah matriks persegi sebagai kunci yang digunakan untuk melakukan enkripsi dan dekripsi serta menggunakan *m* buah persamaan linear. Enkripsi adalah proses dalam menyandikan pesan, sedangkan dekripsi adalah proses dalam menterjemahkan pesan yang telah disandikan. Dalam penerapannya, Sandi Hill menggunakan teknik perkalian matriks dan teknik invers terhadap matriks, sehingga matriks yang digunakan haruslah matriks yang *invertible*.

Ide dari algoritma Sandi Hill adalah untuk membuat *m* kombinasi linear dari *m* karakter alfabetik di dalam suatu elemen *plainteks*, sehingga dihasilkan *m* karakter alfabetik sebagai elemen dari *cipherteks* (Stinson, 2006).

Jika *A* adalah matriks *m x m* atas Z*26* dan sehingga dihitung sebagaimana ditunjukkan pada persamaan (2.59)

(2.59)

sehingga dapat ditulis *y = Ax*.

Fungsi dekripsinya diturunkan dari persamaan (2.49), karena *y = Ax* jika *A****-1*** ada maka *x = A****-1*** *y*. Hal ini akan menjadi landasan dari aplikasi invers matriks tergeneralisasi pada algoritma Sandi Hill.

Sandi Hill mengambil matriks *A* atas *Z26* sebagai kunci. Pasangan matriks digunakan untuk proses enkripsi dan dekripsi. Sebelum membagi teks-teks menjadi deretan blok-blok, terlebih dahulu *plainteks* dikorespondensikan antara huruf dan bilangan, dengan ketentuan bilangan = *Z26* = {0, 1, 2, …, 26} dan huruf = {A, B, C, …, Z}.

Contoh 2.9

Jika terdapat *plainteks* P:

P = MATEMATIKA

Maka *plainteks* tersebut dikonversi menjadi:

P = 13 0 19 4 13 0 19 8 10 0

BAB III

**METODOLOGI PENELITIAN**

1. **Jenis Penelitian**

Jenis penelitian ini merupakan penelitian terapan/aplikasi yang bertujuan menerapkan konsep Matriks Invers Tergeneralisasi dalam bidang kriptografi, khususnya pada Sandi Hill.

1. **Lokus dan Waktu Penelitian**

Penelitian ini dilakukan dengan target penyelesaian 6 bulan terhitung pada Februari 2012 sampai Juli 2012. Sedangkan tempat penelitian dilakukan secara fleksibel.

1. **Fokus Kajian**

Penelitian ini menggunakan metode kajian pustaka yakni dengan mengkaji beberapa literatur yang berkaitan dengan aljabar linear dalam menyelesaikan masalah Matriks Invers Tergeneralisasi (MIT). Secara umum, dikaji proses enkripsi dan dekripsi pesan yang kemudian diaplikasikan dalam Sandi Hill yang diperluas pada Z29.

1. **Material Pendukung**

Material yang digunakan dalam penelitian ini adalah berupa buku teks, jurnal, dan prosiding. Selain itu, untuk membantu dalam simulasi hasil penelitian ini digunakan notebook jenis Acer dengan spesifikasi intel atom, processor N550, memori 1 Gb DDR3 dengan software bahasa pemrograman Free Pascal.

44

1. **Prosedur Kerja**

MENYUSUN ALGORITMA PEMBUATAN PROGRAM

SIMULASI

PENGAKAJIAN INVERS MATRIKS, DEKOMPOSISI NILAI SINGULAR, DAN MATRIKS INVERS TERGENERALISASI

PENGKAJIAN SYARAT DALAM PENERAPAN MATRIKS INVERS TERGENERALISASI PADA SANDI HILL

MENEMUKAN INVERS MATRIKS BERDASAR PADA SYARAT YANG TELAH DIKETAHUI

INVERS MATRIKS YANG TELAH DITEMUKAN, DIGUNAKAN DALAM PROSES ENKRIPSI DAN DEKRIPSI

Gambar 3.1: Peta Alur Penelitian

ENKRIPSI PESAN

MEMILIH MATRIKS PERSEGI PANJANG SEBAGAI KUNCI ENKRIPSI

MENGKONVERSI KARAKTER PLAINTEKS DALAM BENTUK *ZP*

MEMBAGI PLAINTEKS DALAM BENTUK MATRIKS SESUAI UKURAN KUNCI ENKRIPSI

DILAKUKAN PERHITUNGAN ENKRIPSI SEHINGGA DIPEROLEH CIPHERTEKS

MENGKONVERSI DERETAN ANGKA CIPHERTEKS DALAM BENTUK ALFABETIK

Gambar 3.2 : Alur Proses Enkripsi

DEKRIPSI PESAN

INVERS DARI MATRIKS KUNCI ENKRIPSI DIGUNAKAN SEBAGAI KUNCI DEKRIPSI

MENGKONVERSI KARAKTER CIPHERTEKS DALAM BENTUK *ZP*

MEMBAGI CIPHERTEKS DALAM BENTUK MATRIKS SESUAI UKURAN KUNCI DEKRIPSI

DILAKUKAN PERHITUNGAN DEKRIPSI SEHINGGA DIPEROLEH CIPHERTEKS

MENGKONVERSI DERETAN ANGKA PLAINTEKS DALAM BENTUK ALFABETIK

Gambar 3.3 : Alur Proses Dekripsi

Adapun prosedur kerja yang dilakukan dalam penelitian ini adalah sebagai berikut ini.

1. Mengumpulkan referensi berupa buku-buku tentang matriks, invers matriks, nilai eigen dan nilai singular suatu matriks, skripsi, jurnal maupun tulisan-tulisan yang dimuat disitus web.
2. Pengkajian invers matriks, dekomposisi nilai singular (SVD), matriks invers tergeneralisasi, Sandi Hill dan hubungannya satu sama lain yang diperoleh dari literatur-literatur yang telah dikumpulkan sebelumnya. Hasil dari pengkajian tersebut, kemudian dianalisis dan disimpulkan untuk digunakan ketahap selanjutnya.
3. Mencari invers matriks tergeneralisasi dari sebuah matriks pada *ZP.* Adapun langkah-langkah dalam menemukan invers tergeneralisasi dimana semua operasi atas *ZP* sebagai berikut:
4. Menentukan matriks persegi panjang dimana *rank(A) = n ≤ m*
5. Mencari hasil dari perkalian matriks *ATA*
6. Misalkan *B = ATA*, maka *B-1* dapat dicari dengan menggunakan operasi baris elementer
7. Invers tergeneralisasi dapat diketahui dengan mengalikan *B-1AT*.
8. Matriks yang telah diuji keabsahannya digunakan sebagai kunci untuk melakukan enkripsi dan inversnya digunakan dalam proses dekripsi. adapun langkah-langkahnya sebagai berikut:

Enkripsi Pesan

1. Dipilih sebarang matriks persegi panjang, misal *A* yang sebelumnya telah ditemukan invers tergeneralisasinya yaitu *A*-.
2. Mengkonversi karakter-karakter pada *plainteks* tersebut ke dalam bentuk *ZP*.
3. Membagi *plainteks* yang telah dikonversi kedalam bentuk matriks sesuai ukuran kunci enkripsi. Misal ukuran kunci enkripsi adalah *m x n*, maka *plainteks* haruslah matriks berukuran *n x q*, dimana *q* adalah penyesuaian dari banyaknya karakter pesan.
4. Melakukan perhitungan enkripsi dengan menggunakan kunci *A*. Misal

*y = Ax*

dimana *A* adalah kunci enkripsi, *x* adalah *plainteks*, dan *y* adalah *cipherteks*.

1. Setelah *cipherteks* diperoleh, deretan angka tersebut kemudian dikonversi kembali dalam bentuk alfabetik. Dengan demikian kita telah memiliki sebuah pesan yang telah disandikan.

Dekripsi pesan

1. Matriks persegi panjang, misal *A* yang sebelumnya telah ditemukan invers tergeneralisasinya yaitu *A*- digunakan sebagai kunci dekripsi.
2. Mengkonversi karakter-karakter pada *cipherteks* tersebut ke dalam bentuk *ZP*.
3. Membagi *cipherteks* yang telah dikonversi kedalam bentuk matriks sesuai ukuran kunci dekripsi. Misal ukuran kunci dekripsi adalah *n x m*, maka *cipherteks* haruslah matriks berukuran *m x q*, dimana *q* adalah penyesuaian dari banyaknya karakter pada *cipherteks*.
4. Melakukan perhitungan dekripsi dengan menggunakan kunci *A-*. Misal

*x = A- y*

dimana *A-* adalah kunci dekripsi, y adalah *cipherteks*, dan x adalah *plainteks*.

1. Setelah *plainteks* diperoleh, deretan angka tersebut kemudian dikonversi kembali dalam bentuk alfabetik. Dengan demikian kita telah menterjemahkan kembali sebuah pesan yang telah disandikan sebelumnya.
2. Menyusun algoritma pembuatan program berdasarkan langkah-langkah yang telah dibuat kemudian menuliskannya dalam bahasa pemrograman Pascal.
3. Pengujian atau simulasi proses enkripsi dan deksripsi dengan bantuan program yang telah dibuat dalam bahasa pemrograman Pascal.

BAB IV

**PEMBAHASAN**

Ide awal dari invers matriks tergeneralisasi (*Generalized Inverses of Matrix*) adalah untuk menggeneralisasi pengertian invers matriks. Seperti telah dibahas pada definisi 2.7, jika *A* matriks persegi yang *invertible*, maka terdapat matriks *B* sehingga *AB = BA = I* dengan *I*  adalah matriks identitas. Matriks *B* disebut invers dari matriks *A*, dinotasikan dengan *A-1*. Kemudian konsep Matriks Invers Tergeneralisasi diberikan dalam definisi 2.24.

1. **Invers Tergeneralisasi Suatu Matriks dengan Menggunakan Dekomposisi Nilai Singular**

Sebelumnya dibuktikan eksistensi matriks invers tergeneralisasi untuk setiap matriks. Dari teorema 2.5, untuk setiap matriks A berukuran *m x n* terdapat matriks uniter *U* dan *V*, dan sebuah matriks nilai singular S sedemikian sehingga *A = USVH*. Dalam teorema 2.5 dan berdasarkan persamaan (2.42) dimana S berukuran m x n dapat didefinisikan persamaan (4.1)

(4.1)

Didapatkan bersama *S* memenuhi empat sifat dari definisi 2.24, selanjutnya dikontruksi untuk setiap matriks *A* yang dituliskan dalam persamaan (4.2)

51

(4.2)

Sekarang akan dibuktikan bahwa memenuhi empat sifat dari definisi 2.24 sebagai invers tergeneralisasi dari matriks :

Bukti :

1. Dapat dilihat bahwa

maka

(4.3)

1. Dapat dilihat bahwa

maka

(4.4)

1. Digunakan sifat : :

maka

(4.5)

1. Digunakan sifat : :

maka

▲ (4.6)

Dengan demikian terbukti bahwa persamaan (4.2) merupakan invers tergeneralisasi dari persamaan (2.41).

Contoh 4.1

Diketahui matriks , akan dicari dekomposisi nilai singularnya (SVD).

Penyelesaian :

Dalam mencari dekomposisi nilai singular, terdapat langkah-langkah penyelesaian yaitu;

1. Membentuk matriks
2. Menetukan nilai eigen berdasarkan teorema 2.3.5 dan nilai singular berdasarkan

Maka dapat diperoleh nilai singularnya, yaitu:

1. Membentuk matriks singular S berdasarkan persamaan (2.42)

Karena matriks *A* berukuran 4 x 3, maka matriks singular pun harus berukuran 4 x 3. Sehingga harus ditambahkan satu baris nol.

dan dapat diketahui dari persamaan (4.1)

1. Membentuk matriks uniter *V* yang diperoleh dari vektor eigen atau berdasar pada teorema 2.3.2.

adalah vektor eigen dari matriks *A* yang terkait dengan λ, jika dan hanya jika λ adalah sebuah solusi tak trivial dari (*A – λI*)*x = 0*, yaitu :

(4.7)

* Untuk , maka persamaan (4.7) menjadi:

(4.8)

Jika dilakukan eliminasi Gauss, maka persamaan (4.8) menjadi:

Misalkan , maka dan

* Untuk , maka persamaan (4.7) menjadi:

Jika dilakukan eliminasi Gauss, maka persamaan (4.8) menjadi:

Misalkan , maka dan

* Jika , maka persamaan (4.7) menjadi:

Jika dilakukan eliminasi Gauss, maka persamaan (4.8) menjadi:

Misalkan , maka dan

Sehingga diperoleh :

Maka diperoleh

1. Membentuk matriks uniter *U* berdasarkan persamaan 2.41 dari nilai singular tak nol

Karena matriks *A* berukuran 4 x 3, maka matriks uniter *U* berukuran 4 x 4. Sehingga perlu diketahui nilai dan . Untuk mengetahui vektor dan akan digunakan sifat ,sehingga

Misalkan dan , maka dan

1. Sehingga bentuk SVD (persamaan 2.41) dari matriks *A* adalah:
2. Selanjutnya dapat diperoleh Invers tergeneralisasi *A* dari persamaan (4.2)

▲

1. **Dasar Perhitungan Matriks Invers Tergeneralisasi**

Berdasarkan teorema 2.6, akan dibuktikan bahwa jika maka dan maka :

Bukti :

1. Dapat dilihat bahwa dapat terdefinisi jika *invertibel* , dimana berukuran n x n. Sehingga haruslah dan disebut rank kolom penuh*.* Selanjutnya akan dibuktikan bahwa memenuhi keempat kondisi dari definisi 2.24.

Untuk kondisi 1:

Untuk kondisi 2 :

Untuk kondisi 3 :

Untuk kondisi 4 :

Jika memperhatikan pembuktian diatas, terutama pada kondisi 3, dapat disimpulkan bahwa tetapi untuk .

1. Dapat dilihat bahwa dapat terdefinisi jika *invertibel* , dimana berukuran m x m. Sehingga haruslah dan disebut rank baris penuh*.* Selanjutnya akan dibuktikan bahwa memenuhi keempat kondisi dari definisi 2.24.

Untuk kondisi 1:

Untuk kondisi 2 :

Untuk kondisi 3 :

Untuk kondisi 4 :

▲

Jika memperhatikan pembuktian diatas, terutama pada kondisi 4, dapat disimpulkan bahwa tetapi untuk .

Dalam metode ini, matriks yang digunakan haruslah rank kolom penuh atau rank baris penuh. Hal ini agar dan *invertibel*.

Contoh 4.2

Tentukanlah Invers tergeneralisasi dari !

Penyelesaian :

Karena rank (A) = 2 dan m ≤ n merupakan rank baris penuh, maka berdasarkan teorema 2.6 digunakan .

Untuk mencari invers matriks digunakan persamaan (2.31).

Dengan demikian diperoleh invers dari matriks *A* adalah ▲

Untuk selanjutnya dalam penerapan invers matriks tergeneralisasi pada Sandi Hill dibutuhkan *A* mempunyai rank kolom penuh, maka *A-* = (*AHA*)*-1AH*. Hal ini akan dijelaskan pada sub bab selanjutnya.

1. **Invers Tergeneralisasi Suatu Matriks Atas *ZP***

Selama ini pembahahasan matriks invers tergeneralisasi dibidang matematika secara umum biasanya berupa matriks yang entri-entrinya real atau kompleks. Tapi untuk aplikasi dibidang kriptografi klasik (Sandi Hill), pembahasan matriks invers tergeneralisasi atas bilangan real atau kompleks tidak banyak membantu.

Pada kriptografi klasik data-data yang berupa huruf dikonversikan ke dalam sistem *Z26*. Sedangkan pada tulisan ini dikembangkan khusus pada *Zp*, dengan *p* bilangan prima. Penulis memilih *Zp* sebab *Zp* adalah lapangan (*field*), sehingga teori-teori yang telah dibahas sebelumnya berlaku. Untuk selanjutnya pembahasan invers matriks tergeneralisasi adalah untuk matriks yang entri-entrinya atas himpunan *Zp*. untuk mempermudah implementasi, semua contoh dan aplikasi untuk selanjutnya dioperasikan atas bilangan modulo 29.

Karena semua entri-entrinya atas himpunan *Zp* , maka konsep dekomposisi nilai singular (SVD) tidak banyak membantu dalam menentukan invers tergeneralisasi sebab hasil dari perhitungan SVD akan menghasilkan himpunan bilangan *rill*. Hal ini karena sangat tidak mungkin membuat invers matriks dengan metode SVD dimana entri-entrinya atas himpunan *ZP.*

Adapun selanjutnya, proses pencarian invers biasa sebuah matriks akan digunakan operasi baris elementer agar dapatmenghasilkan matriks yang entri-entrinya atas *ZP*.

Berikut ini diberikan tabel invers untuk setiap bilangan atas *Z29* kecuali 0. Nilai ini diperoleh berdasarkan algoritma Euclid yang diperluas dari persamaan (2.48).

Tabel 4.1 Bilangan dan inversnya dalam *Z29*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Z* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| *Z-1* | 1 | 15 | 10 | 22 | 6 | 5 | 25 | 11 | 13 | 3 | 8 | 17 | 9 | 27 |
| *Z* | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| *Z-1* | 2 | 20 | 12 | 21 | 26 | 16 | 18 | 4 | 24 | 23 | 7 | 19 | 14 | 28 |

Contoh 4.3

Diberikan adalah himpunan semua matriks atas . Jika dengan , maka matriks adalah invers matriks tergeneralisasi dari *A* atas , sebab memenuhi semua sifat pada Definisi 2.24.

1. **Algoritma Sandi Hill**

Ide dari algoritma Sandi Hill adalah untuk membuat *m* kombinasi linear dari *m* karakter alfabetik di dalam suatu elemen *plainteks*, sehingga dihasilkan *m* karakter alfabetik sebagai elemen dari *cipherteks* (Stinson, 2006).

Pada metode Sandi Hill disyaratkan matriks enkripsi haruslah matriks persegi yang invertible di *Z26*. Dalam hal ini matriks digunakan sebagai kunci untuk menyandikan pesan, sedangkan invers matriks dimanfaatkan untuk menterjemahkan kembali pesan yang telah tersandikan sebelumnya.

Contoh 4.4

Diambil m = 2, maka dapat dituliskan elemen-elemen *plainteks* dalam bentuk dan elemen-elemen *cipherteks* sebagai . Dalam hal ini, dan ditentukan sebagai kombinasi linear dari . Sebagai contoh :

maka kombinasi linear diatas dapat dituliskan dalam notasi persamaan matriks sebagai berikut :

Matriks pada contoh 4.3 memiliki invers pada *Z26*.

yang diperoleh berdasarkan definisi 2.1 yaitu dengan mencari urutan operasi baris elementer tereduksi menjadi matriks identitas, kemudian melakukan urutan operasi yang sama pada *In* untuk mendapatkan *A-1*.

*A : I*  🡺 *I : A-1*

Dilakukan operasi baris elementer:

agar *a11* menjadi entri yang bernilai 1, maka harus dicari invers mod 26 dari entri tersebut yang diperoleh berdasarkan Persamaan (2.48), sehingga diperoleh invers dari 11 mod 26 adalah 19.

Jadi didapatkan invers dari matriks , yaitu .

Pasangan matriks tersebut digunakan untuk proses enkripsi dan dekripsi pada Sandi Hill. Sebelumnya dibuat dulu suatu korespondensi antara bilangan 0 – 25 dengan huruf *a – z*. untuk selanjutnya, huruf kecil digunaknan untuk *plainteks* sedangakan huruf kapital digunakan untuk *cipherteks*.

Tabel 4.2 Korespondensi antara huruf dan bilangan dalam *Z26*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Misalkan Alice mengambil kunci :

Misalkan pesan yang dienkripsi adalah *july* maka didapatkan elemen dari *plainteks* untuk dienkripsi yaitu (9 20 11 24). Perhitungan enkripsinya sebagai berikut :

Didapatkan hasilnya adalah (25 11 1 19) sehingga *cipherteks* dari *july* adalah *YLBT*. Jadi Alice mengirim *YLBT* kepada Bob. Untuk mengetahui pesan yang sesungguhnya, Bob perlu mendekripsi. Proses dekripsinya, sebagai berikut :

Didapatkan hasilnya adalah (9 20 11 24) sehingga diperoleh kembali *plainteks*nya *july*.

Dengan melihat contoh diatas, pesan yang awal memiliki panjang karakter 4 setelah dienkspansi tetap memiliki panjang karakter 4.

1. **Aplikasi Invers Tergeneralisasi pada Sandi Hill yang Diperluas**

Jika pada metode Sandi Hill disyaratkan matriks enkripsi haruslah matriks persegi yang invertible di *Z26*. Tulisan ini mencoba menerapkan teori invers matriks tergeneralisasi pada algoritma Sandi Hill, sehingga nantinya matriks yang digunakan dapat berupa matriks dapat berupa matriks persegi panjang. Pengembangan akan dibahas secara khusus pada *Zp*, dengan *p* bilangan prima.

Diberikan *m,n* bilangan bulat positif, didefinisikan dengan himupanan *plainteks* dan adalah himpunan *cipherteks*.

Diberikan *A* adalah matriks m x n (yang syarat-syaratnya akan dibahas) diambil sebagai kunci. Proses enkripsi bisa dijelaskan sebagai berikut. Untuk setiap , dihitung y dengan langkah-langkah sebagaimana persamaan (4.9)

(4.9)

Jadi dari blok *plainteks* yang panjangnya *n* akan didapatkan *cipherteks* yang panjangnya *m*. Dapat dilihat bahwa fungsi enkripsi memetakan dua himpunan yang diperlihatkan pada gambar berikut :

Gambar 4.1 Fungsi enkripsi

Banyaknya anggota dari himpunan dan dinotasikan dengan . Untuk memenuhi syarat injektif, harus dipenuhi , dengan kata lain haruslah .

Fungsi dekripsinya diturunkan dari fungsi enkripsi diatas. Karena *y = Ax,* jika *A-* ada dalam *Zp*, maka *x = A-y* . Proses dekripsi dapat dipandang sebagai suatu sistem persamaan linear yaitu : *x = A-y*, dengan .

Matriks *A* berukuran *m x n*, agar *eA(x)=Ax*  injektif maka *rank(A) = n* ( rank kolom penuh ). Jadi pemilihan kunci untuk aplikasi invers matriks tergeneralisasi pada algoritma Hill Cipher dibatasi hanya untuk matriks *m x n* yang rank kolom penuh.

Selanjutnya karena *A* mempunyai rank kolom penuh, maka invers tergeneralisasi dari *A* adalah :

(4.10)

Karena semua perhitungan dilakukan diatas *ZP*, maka *AH = AT*. Sehingga didapatkan   
 (4.11)

Telah diketahui bahwa jika *A* mempunyai rank kolom penuh*,* maka (*ATA*) *invertibel*.

1. **Algoritma Aplikasi Invers Tergeneralisasi pada Sandi Hill yang Diperluas Pada *ZP***

Algoritma dari Sandi Hill (*Hill Cipher*) yang diperluas, dikembangkan dari algoritma Sandi Hill berdasar pada syarat fungsi injektif dan persamaan (4.10) :

Dipilih bilangan prima *p*, bilangan bulat , dengan , .

Untuk suatu , untuk setiap , terdapat y sehingga dapat didefinisikan :

(4.12)

dan

(4.13)

dengan

dimana semua operasi tersebut atas ▲

Dapat dilihat bahwa dalam algoritma Sandi Hill yang diperluas ini terjadi ekspansi pesan. Setiap pesan yang panjangnya *n*, setelah diekspansi mempunyai panjang *m*. Hal ini akan lebih menyamarkan pesan jika dibandingkan dengan metode Sandi Hill pada umumnya. Karena pada contoh ini digunakan operasi atas *Z29* maka dilakukan konversi sebagaimana ditunjukkan pada Tabel 4.3.

Tabel 4.3 korespondensi antara huruf dan bilangan dalam *Z29*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H | I | J |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K | L | M | N | O | P | Q | R | S | T |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| U | V | W | X | Y | Z | (spasi) | (koma) | (titik) |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |

Agar dapat dilakukan proses enkripsi dan dekripsi, ada beberapa tahapan yang perlu diperhatikan, yakni:

Tahap pertama adalah pembuatan kunci enkripsi dan dekripsi

1. Dipilih sebarang matriks berukuran m x n sebagai kunci enkripsi untuk n < m

misal matriks berukran 5 x 3.

1. Untuk entri-entrinya dipilih dari anggota *ZP*, dalam hal ini *Z29* dan kunci haruslah rank kolom penuh*.*

misal

1. Digunakan persamaan (4.11) untuk menemukan kunci dekripsinya

Jika dilakukan perhitungan, diperoleh

Dalam mencari invers digunakan operasi baris elementer dan terlebih dahulu dibentuk matriks segitiga atas

B1 x 17-1 (mod 29)

B2 x 27-1 (mod 29)

B3 – 21.B2

B3 x 13-1 (mod 29)

B1 – 5.B2

setelah matriks segitiga atas terbentuk, matriks kiri diubah menjadi matriks identitas

B1 – 8.B3

B2 – 4.B3

Sehingga diperoleh

Melalui perhitungan diperoleh

Dengan demikian kita telah menemukan kunci enkripsi dan dekripsi yaitu

Tahap kedua adalah dilakukan enkripsi pada pesan. Proses enkripsi di gambarkan pada Gambar 4.2.

------------------------

Plainteks

Cipherteks

Enkripsi

*Ke*

Gambar 4.2: Proses Enkripsi

Misalkan Andi akan mengirimkan pesan rahasia kepada Bob yang sebelumnya telah diberikan kunci dekripsinya kepada Bob. Misalkan pesan Andi adalah *CIPHER*.

1. Mengkonversi karakter-karakter pada pesan tersebut ke dalam bentuk *Z29* berdasarkan pada tabel 4.3.

CIPHER = 2, 8, 15, 7, 4, 17

1. Membagi *plainteks* yang telah dikonversi ke dalam bentuk matriks sesuai dengan ukuran kunci enkripsi yaitu ukuran kunci 5 x 3, maka *plainteks* berukuran 3 x *q*

Maka diperoleh

1. Melakukan perhitungan enkripsi dengan menggunakan kunci enkripsi *Ke*, sebagai berikut :

yang mana diperoleh nilai (7, 6, 26, 5, 8, 3, 19, 21, 2, 4) dan jika nilai tersebut dikonversi didapatkan *cipherteks* dari kata *CIPHER* adalah *HG VIDTVCE*. Jadi A mengirim *HG VIDTVCE* kepada B.

Tahap ketiga adalah dilakukan dekripsi pesan. Proses dekripsi di gambarkan pada Gambar 4.3.

------------------------

Plainteks

Cipherteks

Dekripsi

*Kd*

Gambar 4.3: Proses dekripsi

Untuk mengetahui pesan yang sesungguhnya, maka B perlu melakukan langkah-langkah dekripsi.

1. Karakter-karakter pada pesan yang telah disandikan, dikonversi ke dalam bentuk *Z29* berdasarkan pada Tabel 4.3.

*HG VIDTVCE* = 7, 6, 26, 5, 8, 3, 19, 21, 2, 4

1. *Cipherteks* yang telah dikonversi dibagi dalam bentuk matriks sesuai dengan ukuran kunci dekripsinya yaitu ukuran kunci 3 x 5, maka *cipherteks* berukuran 5 x *q*

maka diperoleh

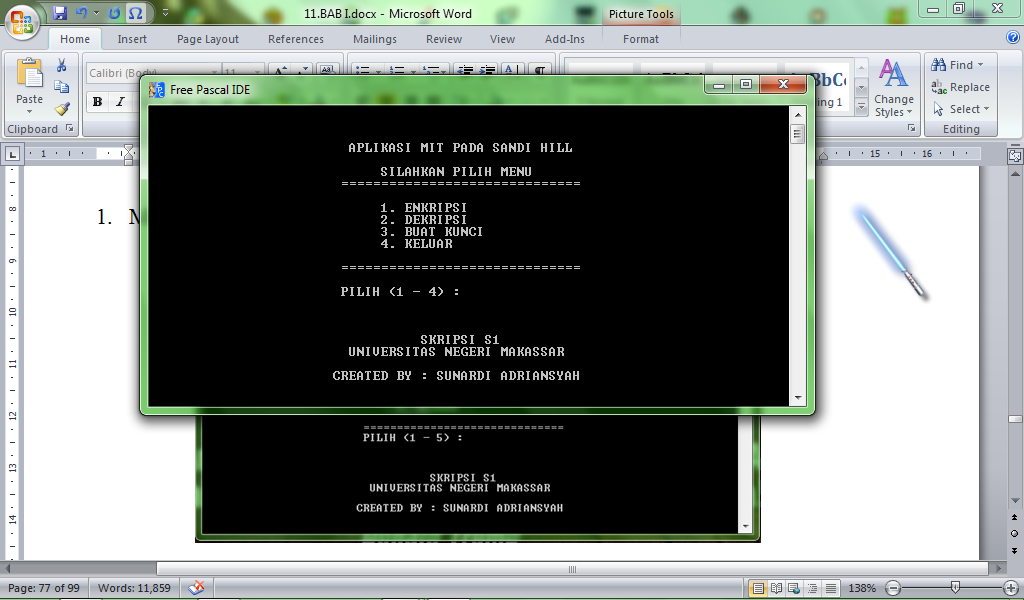
1. Perhitungan dekripsi dilakukan menggunakan kunci dekripsi *Kd*, sebagai berikut :

Berdasarkan perhitungan diatas diperoleh nilai: (2, 8, 15, 7, 4, 17). Nilai tersebut jika dikonversi, didapatkan kembali plainteknya adalah *CIPHER.* Jadi B menerima pesan dari A yang berisikan kata *CIPHER*.

1. **Simulasi**

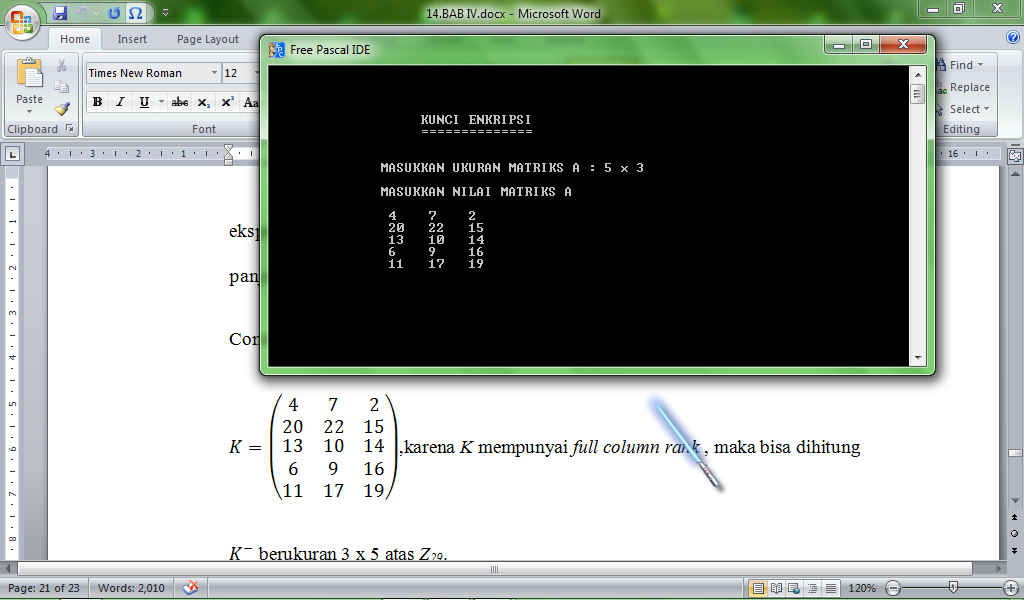
Simulasi enkripsi dan dekripsi pesan dilakukan menggunakan bahasa pemrograman Pascal. Simulasi dilakukan dengan membuat sebuah kunci penterjemah terlebih dahulu kemudian digunakan untuk menterjemahkan pesan yang telah disandikan. Simulasi penelitian ini dapat dijelaskan sebagai berikut:

1. Membuat kunci enkripsi dan kunci dekripsi



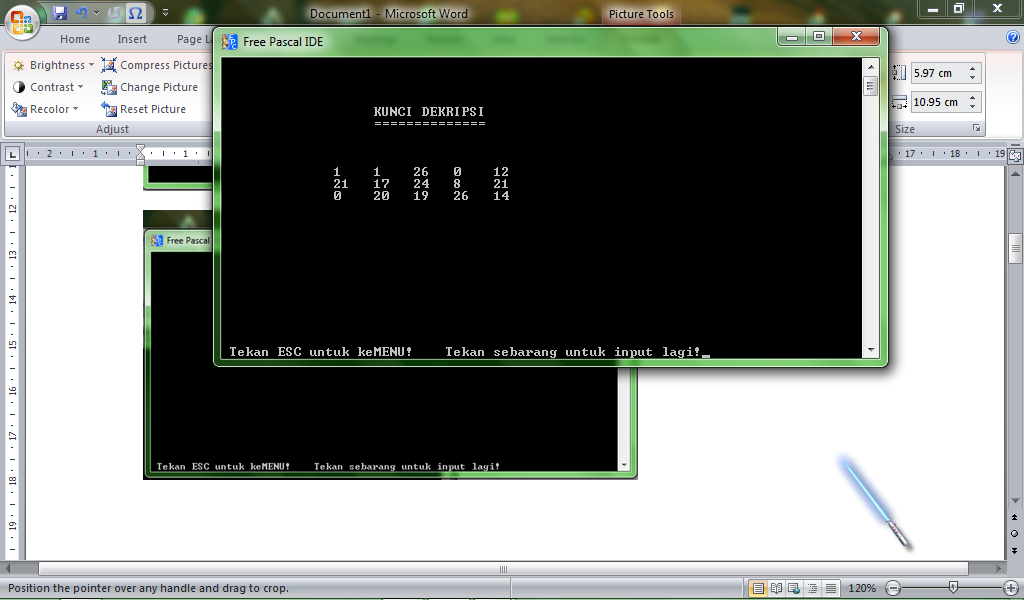
Gambar 4.4 Tampilan awal program

Tampilan awal dari program ditunjukkan pada Gambar 4.4. Apabila belum memiliki kunci enkripsi dan dekripsi maka dapat dibuat dengan memilih opsi nomor 3 lalu menekan enter. Dengan demikian akan muncul tampilan seperti Gambar 4.5.



Gambar 4.5 Tampilan kunci enkripsi

Setelah dilakukan pemilihan ukuran kunci, selanjutnya pengguna dapat langsung memasukkan nilai kunci enkripsi yang nantinya dapat digunakan untuk mengenkripsi pesan. Pada Gambar 4.5 diambil contoh ukuran kunci enkripsi 5 x 3.

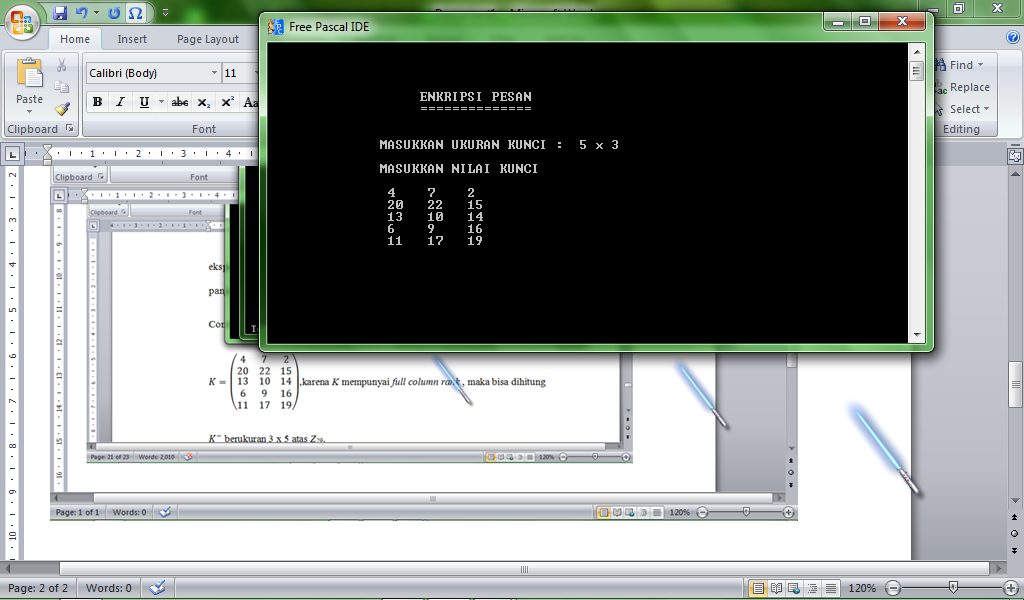


Gambar 4.6 Tampilan kunci dekripsi

Setelah menekan enter pada keyboard, maka akan tercipta sebuah kunci rahasia seperti pada Gambar 4.6. Kunci ini nantinya dapat digunakan dalam melakukan dekripsi pesan. Kunci ini diharapkan dijaga agar tidak diketaui oleh orang yang tidak memiliki hak untuk mengetahuinya.

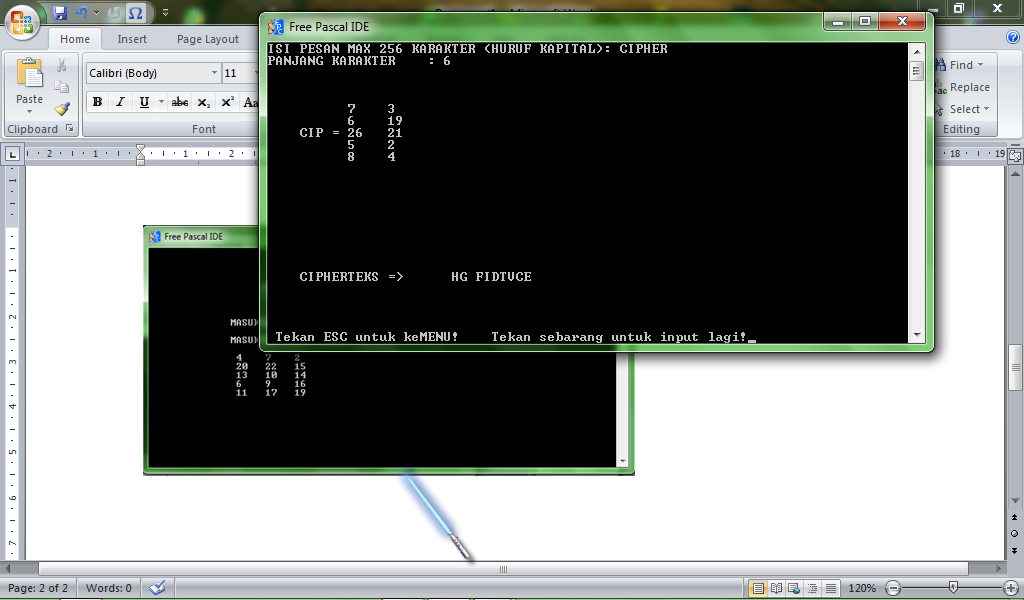
1. Enkripsi pesan

Setelah kunci dibuat, maka pengguna telah dapat melakukan enkripsi pesan. Untuk melakukan enkripsi pesan, pengguna harus memilih opsi 1 pada tampilan awal program yang ditunjukkan pada Gambar 4.4 dan kemudian menekan enter sehingga akan muncul Gambar 4.7.



Gambar 4.7 Enkripsi pesan

Gambar 4.7 adalah tampilan untuk memasukkan kunci enkripsi. Pada tahapan ini (Gambar 4.7) pengguna harus memasukkan kembali kunci enkripsi yang telah dibuat sebelumnya kemudian menekan enter maka akan mucul tampilan seperti pada Gambar 4.8.



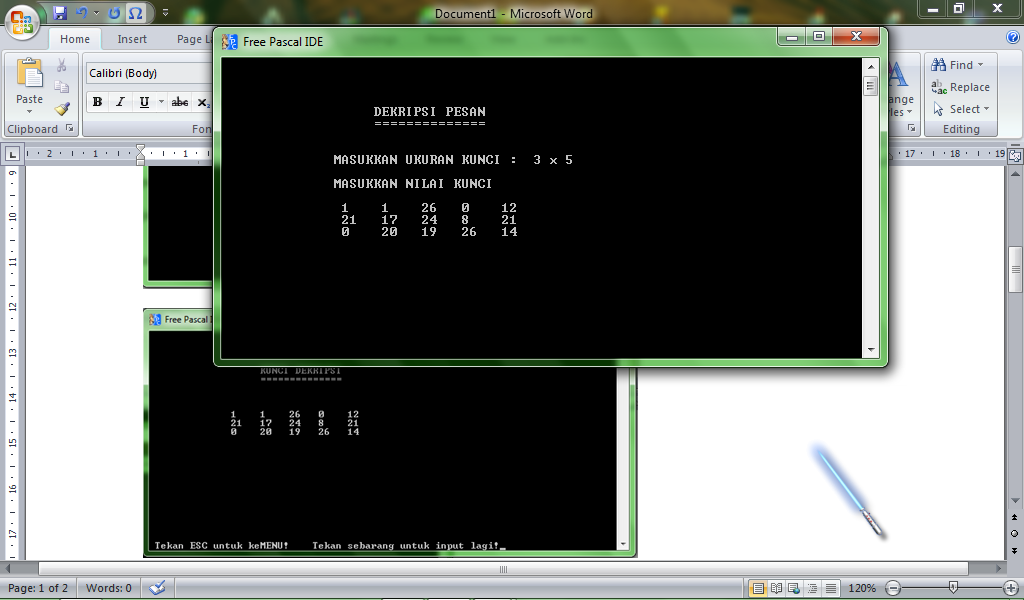
Gambar 4.8 Hasil enkripsi pesan

Selanjutnya pengguna harus memasukkan pesan yang ingin dienkripsi dimana pengguna hanya boleh mengenkripsi pesan maksimal 256 karakter. Karakter yang ingin dienkripsi dibatasi pada huruf kapital saja dan tiga karakter lainnya yaitu spasi, koma, dan titik.

Misalkan pada simulasi ini, pesan yang akan dienkripsi (*plainteks*) adalah kata “*CIPHER*”, maka setelah menekan enter akan muncul pesan yang telah dienkripsi (*cipherteks*) yaitu kata “*HG FIDTVCE*”.

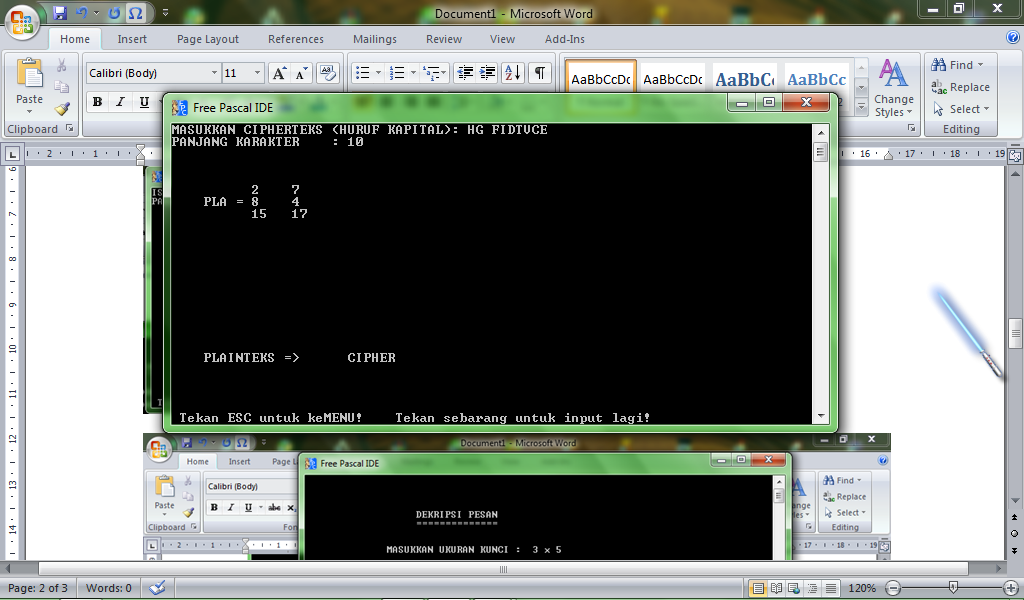
1. Dekripsi pesan

Setelah pengguna melakukan enkripsi dan memperoleh *cipherteks*nya, pengguna dapat mengirimkan pesan yang telah disandikan tersebut ke penerima pesan. Dalam hal ini penerima pesan akan memperoleh *cipherteks*. Agar dapat membacanya, maka pesan harus didekripsi atau diterjemahkan, seperti tampilan pada Gambar 4.9.



Gambar 4.9 Dekripsi pesan

Untuk dekripsi pesan, maka harus dipilih opsi 2 pada tampilan awal program yang ditunjukkan pada Gambar 4.4 sehingga Gambar 4.9 akan ditampilkan. Sebelum dilakukan dekripsi pesan, maka terlebih dahulu harus dimasukkan kunci dekrispsi.



Gambar 4.10 Hasil dekripsi pesan

Sebelumnya telah diperoleh sebuah *cipherteks* yaitu “*HG FIDTVCE*”. *Cipherteks* ini dimasukkan ke dalam program dan secara otomatis program akan menterjemahkan *cipherteks* tersebut dan memperoleh kembali *cipherteks* atau pesan awal yaitu “*CIPHER*”.

BAB V

**PENUTUP**

1. **Kesimpulan**

Berdasarkan hasil dan pembahasan, maka dapat disimpulkan sebagai berikut :

1. Invers dari matriks persegi panjang dapat diperoleh dengan cara :

dimana matriks singular V berisikan vektor eigen dari *AHA*, kolom-kolom matriks singular U diperoleh dari persamaan (2.42), dan matriks singular S- dapat diperoleh dari persamaan (4.1).

1. untuk dimana *AHA* invertibel atau untuk dimana *AAH* invertibel.
2. Berdasarkan penerapan Matriks Invers Tergeneralasasi didapatkan algoritma Sandi Hill yang diperluas sebagai berikut:

Dipilih bilangan prima *p*, bilangan bulat , dengan , .

Untuk suatu , untuk setiap , terdapat y sehingga dapat didefinisikan :

dan

dengan

dimana semua operasi tersebut atas .

83

1. **Saran**
2. Bagi peliti selanjutnya dapat dikembangkan tentang bagaimana perluasan Sandi Hill pada ring komutatif, sehingga dapat diterapkan di sebarang *Zn*.
3. Pengembangan diharapkan pada bagaimana Sandi Hill dan sandi-sandi lainnya dapat diaplikasikan selain pada teks, seperti data numerik maupun citra.

DAFTAR PUSTAKA

Anton, H. 1987. *Aljabar Linear Elementer Edisi Kelima*. Jakarta: Erlangga.

Anton, H dan Rorres, C. 1987. *Penerapan Aljabar Linear*. Jakarta: Erlangga.

Defls, H., dan Knebl, H. 2007. *Introduction to Cryptography*. New York: Springer.

Leon, S. J. 1998. *Aljabar Linear dan Aplikasinya*. Jakarta: Erlangga.

Murtiyasa, B. 2001. Aplikasi Matriks Invers Tergeneralisasi pada Kriptografi. *Makalah* pada Seminar Nasional dan konferda Matematika VII di FMIPA UII Yogyakarta tanggal 3 Februari 2001.

Sibaroni, Y. 2002. *Buku Ajar Aljabar Linear*. Bandung: STT Telkom.

Stinson, D. R. 2006. *Cryptography Theory and Practice*. Boca Raton: CRC Press.

Strang, G. 1988. *Linear Algebra and Its Applications*. USA: Thomson Learning, Inc.

Tiro, M. A., dkk. 2008. *Pengenalan Teori Bilangan*. Makassar: Andira Publisher.

Tahmir, S. 2005. *Teori Ring*. Makassar: Andira Publisher.

Tahmir, S. 2004. *Teori Grup*. Makassar: Andira Publisher.

Wijna. 2009. *Dekomposisi Matriks*. Yogyakarta: Universitas Gadjah Mada.

Yanai, H., Takeuchi, K., dan Takane, Y. 2011. *Projection Matrices, Generalized Inverse Matrices, and Singular Value Decomposition*. New York: Springer.

LAMPIRAN

Listing Program

Program Operasi\_Matriks;

Uses Crt;

Label lompat,MENU;

Var

STR : STRING;

I,J,K,L,M,N,O,P,Q,R : INTEGER;

S,X,pilihan : INTEGER;

A,B,C,D,E : ARRAY [0..99,0..99] OF INTEGER;

KODE : ARRAY [1..256] OF INTEGER;

JOL : CHAR;

Const

KURUNG = 179;

MO = 29;

{FUNGSI UNTUK MENCARI MODULO}

FUNCTION MODULO (X:INTEGER) : INTEGER;

VAR R : INTEGER;

BEGIN

R:=ABS(X) MOD MO;

IF X>=0 THEN

MODULO:=R;

IF (X<0) AND (R<>0) THEN

MODULO:=MO-R;

IF (X<0) AND (R=0) THEN

MODULO:=0;

END;

{FUNGSI UNTUK MENCARI INVERS MODULO}

FUNCTION INVMOD (X:INTEGER) : INTEGER;

VAR N0,B0,T0,T,Q,R,TEMP : INTEGER;

BEGIN

N0:=MO;

B0:=X;

T0:=0;

T:=1;

Q:=N0 DIV B0;

R:=N0 - Q\*B0;

WHILE R>0 DO

BEGIN

TEMP:=T0-Q\*T;

TEMP:=MODULO(TEMP);

T0:=T;

T:=TEMP;

N0:=B0;

B0:=R;

Q:=N0 DIV B0;

R:=N0 - Q\*B0;

END;

INVMOD:=MODULO(T);

END;

{PROSEDURE KONVERSI STRING KE NILAI INTEGER}

PROCEDURE SKEN;

BEGIN

R:=(S DIV N + S MOD N)\*N;

FOR I:=1 TO S DO

BEGIN

IF ORD(STR[I])=46 THEN

KODE[I]:=ORD(STR[I])-18

ELSE

IF ORD(STR[I])=44 THEN

KODE[I]:=ORD(STR[I])-17

ELSE

IF ORD(STR[I])=32 THEN

KODE[I]:=ORD(STR[I])-6

ELSE

KODE[I]:=ORD(STR[I])-65;

END;

IF S MOD N <> 0 THEN

BEGIN

FOR I:=S+1 TO R DO

KODE[I]:=26;

END;

END;

{PROSEDURE KONVERSI NILAI KE STRING INTEGER}

PROCEDURE NKES;

BEGIN

IF KODE[I]=26 THEN

WRITE(CHR(KODE[I]+6))

ELSE

IF KODE[I]=27 THEN

WRITE(CHR(KODE[I]+17))

ELSE

IF KODE[I]=28 THEN

WRITE(CHR(KODE[I]+18))

ELSE

WRITE(CHR(KODE[I]+65));

END;

{Program Utama}

BEGIN

MENU:

ClrScr;

GotoXy (25,4); Write (' APLIKASI MIT PADA SANDI HILL');

GotoXy (30,6); Write ('SILAHKAN PILIH MENU');

GotoXy (25,7); write ('==============================');

GotoXy (30,9); Write ('1. ENKRIPSI');

GotoXy (30,10); Write ('2. DEKRIPSI');

GotoXy (30,11); Write ('3. BUAT KUNCI');

GotoXy (30,12); write ('4. KELUAR');

GotoXy (25,14); Write ('==============================');

GotoXy (35,20); Write ('SKRIPSI S1');

GotoXy (26,21); Write ('UNIVERSITAS NEGERI MAKASSAR');

GotoXy (24,23); Write ('CREATED BY : SUNARDI ADRIANSYAH');

GotoXy (25,16); Write ('PILIH (1 - 4) : ');readln(pilihan);

Case pilihan of

1: BEGIN

REPEAT

CLRSCR;

GOTOXY(20,5);WRITE('ENKRIPSI PESAN');

GOTOXY(20,6);WRITE('==============');

GOTOXY(15,9);WRITE('MASUKKAN UKURAN KUNCI :');

GOTOXY(40,9);READ(M);

GOTOXY(42,9);WRITE('x');GOTOXY(44,9);READLN(N);

IF N < M THEN

BEGIN

GOTOXY(15,11);WRITE('MASUKKAN NILAI KUNCI');

FOR I := 1 TO M DO

BEGIN

FOR J := 1 TO N DO

BEGIN

GOTOXY(J\*5+11,I+12); READLN(A[I,J]);

END;

END;

END

ELSE

BEGIN

GOTOXY(15,15); WRITE('UKURAN MATRIKS SALAH !');

GOTO lompat;

END;

CLRSCR;

{ALGORITMA KONFERSI}

WRITE('ISI PESAN MAX 256 KARAKTER (HURUF KAPITAL): '); READLN(STR);

WRITELN('PANJANG KARAKTER : ',LENGTH(STR));

WRITELN;

S:=LENGTH(STR);

SKEN;

L:=0;

FOR I:=1 TO (R DIV N) DO

FOR J:=1 TO N DO

BEGIN

L:=L+1;

B[J,I]:=KODE[L];

END;

FOR I:= 1 TO M DO

BEGIN

FOR K:= 1 TO (R DIV N) DO

BEGIN

C[I,K]:=0;

For J:= 1 TO N DO

C[I,K]:=MODULO(C[I,K] + A[I,J] \* B[J,K])

END;

END;

{CETAK *CIPHERTEKS*}

IF (M MOD 2 = 0) THEN

BEGIN GOTOXY(5,M DIV 2+5); WRITE('CIP = '); END

ELSE

BEGIN GOTOXY(5,M DIV 2+6); WRITE('CIP = '); END;

FOR I:=1 TO M DO

BEGIN

FOR J:=1 TO (R DIV N) DO

BEGIN

GOTOXY(J\*5+6,I+5); WRITELN (C[I,J]);

END;

END;

GOTOXY(5,20); WRITE ('*CIPHERTEKS* => ');

I:=0;

FOR J:=1 TO (R DIV N) DO

BEGIN

FOR K:=1 TO M DO

BEGIN

I:=I+1;

KODE[I]:=C[K,J];

NKES;

END;

END;

GotoXy(2,25);Write('Tekan ESC untuk keMENU! Tekan sebarang untuk input lagi!');

jol:=readkey;

Until Jol=#27;

END;

2: BEGIN

REPEAT

CLRSCR;

GOTOXY(20,5);WRITE('DEKRIPSI PESAN');

GOTOXY(20,6);WRITE('==============');

GOTOXY(15,9);WRITE('MASUKKAN UKURAN KUNCI :');

GOTOXY(40,9);READ(M);

GOTOXY(42,9);WRITE('x');GOTOXY(44,9);READLN(N);

IF M < N THEN

BEGIN

GOTOXY(15,11);WRITE('MASUKKAN NILAI KUNCI');

FOR I := 1 TO M DO

BEGIN

FOR J := 1 TO N DO

BEGIN

GOTOXY(J\*5+11,I+12); READLN(A[I,J]);

END;

END;

END

ELSE

BEGIN

GOTOXY(15,15); WRITE('UKURAN MATRIKS SALAH !');

GOTO lompat;

END;

CLRSCR;

{ALGORITMA KONFERSI}

WRITE('MASUKKAN *CIPHERTEKS* (HURUF KAPITAL): '); READLN(STR);

WRITELN('PANJANG KARAKTER : ',LENGTH(STR));

WRITELN;

S:=LENGTH(STR);

SKEN;

L:=0;

FOR I:=1 TO (R DIV N) DO

FOR J:=1 TO N DO

BEGIN

L:=L+1;

B[J,I]:=KODE[L];

END;

FOR I:= 1 TO M DO

BEGIN

FOR K:= 1 TO (R DIV N) DO

BEGIN

C[I,K]:=0;

For J:= 1 TO N DO

C[I,K]:=MODULO(C[I,K] + A[I,J] \* B[J,K])

END;

END;

{CETAK *CIPHERTEKS*}

IF (M MOD 2 = 0) THEN

BEGIN GOTOXY(5,M DIV 2+5); WRITE('PLA = '); END

ELSE

BEGIN GOTOXY(5,M DIV 2+6); WRITE('PLA = '); END;

FOR I:=1 TO M DO

BEGIN

FOR J:=1 TO (R DIV N) DO

BEGIN

GOTOXY(J\*5+6,I+5); WRITELN (C[I,J]);

END;

END;

GOTOXY(5,20); WRITE ('*PLAINTEKS* => ');

I:=0;

FOR J:=1 TO (R DIV N) DO

BEGIN

FOR K:=1 TO M DO

BEGIN

I:=I+1;

KODE[I]:=C[K,J];

NKES;

END;

END;

GotoXy(2,25);Write('Tekan ESC untuk keMENU! Tekan sebarang untuk input lagi!');

jol:=readkey;

Until Jol=#27;

END;

3: BEGIN

REPEAT

CLRSCR;

GOTOXY(20,5);WRITE('KUNCI ENKRIPSI');

GOTOXY(20,6);WRITE('==============');

GOTOXY(15,9);WRITE('MASUKKAN UKURAN MATRIKS A :');

GOTOXY(43,9);READ(M);

GOTOXY(45,9);WRITE('x');GOTOXY(47,9);READLN(N);

IF N < M THEN

BEGIN

GOTOXY(15,11);WRITE('MASUKKAN NILAI MATRIKS A');

FOR I := 1 TO M DO

BEGIN

FOR J := 1 TO N DO

BEGIN

GOTOXY(J\*5+11,I+12); READLN(A[I,J]);

END;

END;

END

ELSE

BEGIN

GOTOXY(15,15); WRITE('UKURAN MATRIKS SALAH !');

GOTO lompat;

END;

{C = A TRANSPOSE}

FOR I:= 1 TO N DO

BEGIN

FOR J:= 1 TO M DO

C[I,J]:= A[J,I];

END;

{D = A TRANSPOSE X A}

FOR I:= 1 TO N DO

BEGIN

FOR K:= 1 TO N DO

BEGIN

D[I,K]:=0;

For J:= 1 TO M DO

BEGIN

D[I,K]:=MODULO(D[I,K] + A[J,I] \* A[J,K]);

E[I,K]:= D[I,K];

END;

END;

END;

{Algoritma INVERS D}

{MATRIKS IDENTITAS}

FOR I:=1 TO N DO

BEGIN

For J:=(N+1) TO (2\*N) DO

BEGIN

IF (I+N)=J THEN

D[I,J]:= 1

ELSE

D[I,J]:= 0;

END;

END;

{MEMBUAT MATRIKS SEGITIGA ATAS}

FOR I:= 1 TO N DO

BEGIN

IF D[I,I]=0 THEN

BEGIN

GOTOXY (15,WHEREY+1); WRITE('MATRIKS TIDAK FULL COLUMN RANK');

GOTO LOMPAT;

END;

X:=INVMOD(D[I,I]);

FOR K:=I TO (2\*N) DO

BEGIN

D[I,K]:=MODULO(X\*D[I,K]);

END;

FOR J:=(I+1) TO N DO

BEGIN

For L:=(2\*N) DOWNTO I DO

BEGIN

D[J,L]:= MODULO(D[J,L] - D[J,I]\*D[I,L]);

END;

END;

END;

{PENYELESAIAN MATRIKS SEGITIGA ATAS UNTUK MEMBENTUK

MATRIKS IDENTITAS KIRI}

FOR I:=1 TO N DO

BEGIN

FOR J:=(I+1) TO N DO

BEGIN

For K:=(2\*N) DOWNTO J DO

BEGIN

D[I,K]:= MODULO(D[I,K] - D[I,J]\*D[J,K]);

END;

END;

END;

{PERBAIKAN POSISI ENTRI}

FOR I:=1 TO N DO

BEGIN

FOR J:=(N+1) TO (2\*N) DO

BEGIN

D[I,J-N]:= D[I,J];

END;

END;

{INVERS TERGENERALISASI}

FOR I:= 1 TO N DO

BEGIN

FOR K:= 1 TO M DO

BEGIN

B[I,K]:=0;

For J:= 1 TO N DO

B[I,K]:=B[I,K] + D[I,J] \* C[J,K]

END;

END;

{CETAK INVERS TERGENERALISASI}

CLRSCR;

GOTOXY(20,5); WRITE('KUNCI DEKRIPSI');

GOTOXY(20,6); WRITE('==============');

FOR I:=1 TO N DO

BEGIN

FOR J:=1 TO M DO

BEGIN

GOTOXY(J\*5+10,I+9); WRITELN (MODULO(B[I,J]));

END;

END;

lompat:

GotoXy(2,25);Write('Tekan ESC untuk keMENU! Tekan sebarang untuk input lagi!');

jol:=readkey;

Until Jol=#27;

END;

4: EXIT;

END;

GOTO MENU;

END.

**RIWAYAT HIDUP**

Sunardi Adriansyah, lahir di Polmas, tanggal 16 Agustus 1989 sebagai anak kedua dari lima bersaudara. Anak dari pasangan Sahabuddin dan Dahlia H.N ini, memasuki jenjang pendidikan Sekolah Dasar di SDN 002 Monginsidi Balikpapan tahun 1996 sampai tahun 1999 kemudian pindah di SDI Togo-Togo Kab. Jeneponto dan tamat pada tahun 2002. Pada tahun 2002 diterima di SMPN 1 Arungkeke dan tamat pada tahun 2005. Pada tahun yang sama penulis melanjutkan pendidikan dan diterima di SMAN 1 Polewali dan tamat pada tahun 2008. Pada tahun 2008 melalui jalur SNMPTN diterima sebagai mahasiswa pada jurusan Matematika di Universitas Negeri Makassar (UNM).