Aplikasi Enkripsi dan Dekripsi Teks Menggunakan Algoritma Merkle Hellman

Lamrianto Purba
STMIK Budi Darma Medan, Jl. SM.Raja No.338 Sp.Limun Medan, Sumut, Indonesia
E-Mail: rianpurba32@gmail.com,

Guidio Leonarde Ginting
STMIK Budi Darma Medan, Jl. SM.Raja No.338 Sp.Limun Medan, Sumut, Indonesia
E-Mail: guidio_leonard@yahoo.co.id

ABSTRACT
Cryptography is the science and art of keeping messages safe. In cryptography there are two main concepts, namely encryption and decryption. The encoding is first created using the clasic algorithm. this algorithm builds security on the confidentiality of the algorithm used. But this algorithm is not efficient when used to communicate with many people because the algorithm is still very simple and still very easy to solve, so that important information or data that can be easily kept secret can be recognized by other people or irresponsible people. Merkle Hellman algorithm is a cryptographic algorithm that generally encrypts twice or also called double, namely plainteks encrypted with cipher I, then the first encryption results are encrypted again with cipher II.

Keywords: Cryptography, Encryption, Decryption, Merkle Hellman Algorithm

PENDAHULUAN

Dengan adanya perkembangan teknologi yang sangat pesat pada masa sekarang, membuat manusia membutuhkan suatu sistem untuk mempermudah bertukar informasi pada suatu instansi ataupun yang sesuai dengan kebutuhan. Kemajuan sistem sekarang ini memiliki banyak keuntungan tetapi juga rawan dari hal yang negatif seperti pencurian suatu informasi. Misalnya pada sebuah perusahaan, beberapa informasi yang sifatnya rahasia dan hanya boleh diketahui oleh orang-orang tertentu. Seandainya data yang berisi informasi tersebut jatuh kepada pihak lawan bisnis, maka perusahaan akan mengalami kerugian. Untuk mengantisipasi hal yang tidak diinginkan seperti pencurian informasi, maka dibutuhkan suatu sistem untuk mengamankan suatu informasi.

Dengan begitu memicu banyaknya algoritma kriptografi yang bermunculan sesuai dengan perkembangan zaman[1]. Sampai saat ini algoritma kriptografi modern sendiri berkembang pesat. Algoritma kriptografi modern dapat dibagi menjadi dua algoritma kunci yaitu algoritma kunci simetris dan algoritma asimetris[2]. Algoritma simetris adalah algoritma yang menggunakan kunci enkripsi dan dekripsi yang sama. Sedangkan algoritma asimetris adalah algoritma yang menggunakan kunci enkripsi dan dekripsi yang berbeda[3]. Adapula algoritma yang beroperasi dalam mode bit dapat dibagi menjadi dua, yaitu *stream chiper* (chipper aliran) dan *block chiper* (cipher blok). Stream chiper adalah algoritma yang melakukan operasi dalam bentuk bit tunggal. Sedangkan *block chiper* adalah algoritma yang melakukan operasi dalam bentuk blok bit. *Stream chiper* dan *block chiper* adalah algoritma yang digunakan pada algoritma kunci simetris.

Untuk melindungi akses dari pihak-pihak yang tidak berkepentingan tersebut maka sangat diperlukan enkripsi dan dekripsi. Agar dapat dilakukan dengan baik, dibutuhkan suatu algoritma untuk enkripsi dan dekripsi, maka algoritma yang digunakan adalah Algoritma Merkle Hellman.

Sistem kriptografi Merkle Hellman pertama kali ditemukan oleh Merkle dan Hellman pada tahun 1978. Sistem kripto ini berupa variannyan masih sangat bermanfaat untuk dipelajari terutama berkenan dengan konseptual serta teknik desain yang mendasarinya. Algoritma Merkle Hellman menggunakan kunci Asimetris dalam proses operasi enkripsi dan dekripsinya[4].

Adapun perumusan Masalah yang dibahas adalah Bagaimana mengetahui proses Algoritma Merkle Hellman dalam melakukan enkripsi dan dekripsi, dan bagaimana merancang aplikasi enkripsi dan dekripsi teks?

Adapun batasan masalah dari topik yang dibahas penulis yaitu Format teks yang dienkripsikan adalah txt, Panjang karakter yang bisa dienkripsikan minimal 1 dan maksimal 250 karakter, Teks yang dienkripsi dan didekripsi adalah Rian.

Adapun tujuan yang ingin dicapai adalah untuk mengetahui proses enkripsi dan dekripsi algoritma Merkle Hellman dengan cara memakai kunci asimetrik dalam pengamanan aplikasi teks.

LANDASAN TEORI

2.1 Kriptografi

Kriptografi adalah ilmu yang berguna untuk mengacak data sedemikian rupa, sehingga tidak bisa dibaca oleh pihak ketiga. Tentu saja data yang diacak harus bisa dikembalikan kebentuk semula oleh pihak yang berwenang [5]

Kriptografi memiliki sejarah yang panjang, Informasi yang lengkap mengenai sejarah kriptografi dapat ditemukan dalam buku Davin Khan yang berjudul *The Codebreakers* .Buku yang tebalnya 1000 halaman ini menulis secara rinci sejarah kriptografi mulai dari penggunaan kriptografi oleh Bangsa Mesir 4000 tahun yang lalu (berupa hieroglyph yang tidak standar pada piramid) hingga penggunaan kriptografi pada abad ke-20[6].

2.2. Tujuan Kriptografi

Tujuan kriptografi adalah melindungi data dari ancaman yang disengaja atautidak disengaja dengan mengubah suatu data informasi menjadi sebuah sandi yang hanya akan dimengerti oleh pihak pengirim dan penerima pesan[7].

2.3. Pengertian Teks

Teks adalah ungkapan bahasa yang menuntut isi, sintaksis dan prakmatik yang merupakan satu kesatuan. Dari pengertian tersebut dapat diartikan dengan benar teks adalah suatu kesatuan bahasa yang memiliki isi dan bentuk, baik lisan, maupun tulisan yang disampaikan oleh seseorang pengirim kepada penerima untuk menyampaikan pesan tertentu.

Istilah teks sebenarnya berasal dari kata text yang berarti ‘tenunan’ Teks dalam fiologi diartikan sebagai ‘tenunan kata-kata’, yakini serangkain kata-kata yang berinteraksi membentuk satu-kesatuan makna yang utuh. Teks dapat terdiri dari beberapa kata, namun dapat pula terdiri dari milyaran kata yang tertulis dalam sebuah naskah berisi cerita yang panjang.

2.4. Algoritma Merkel Hellman

Algoritma kriptografi *merkle hellman* atau umumnya yang dikenal dengan sebutan merupakan cipher yang ide awalnya dari algoritma kriptografi *one time pad*, yaitu kunci yang dibangkitkan secara random dan panjang kunci sepanjang plainteks yang akan dienkripsi. Tetapi pada algoritma kriptografi pembangkitan kunci-kunci tersebut secara otomatis dengan teknik berantai[8].

Algoritma ini memiliki aturan subtitusi berdasar pada Caesar *cipher* yaitu dengan pergeseran huruf-huruf. Kekuatan kedua terletak pada barisan bilangan-bilangan yang berfungsi sebagai penggali dengan kunci. Barisan bilangan tersebut dapat berupa bilangan tertentu seperti deret bilangan ganjil, bilangan genap, deret *fibonanci*, deret bilangan prima, serta deret bilangan yang dibuat sendiri[3].

Pada kenyataan cipher subtitusi tidak dibuat secara sederhana, tetapi dengan mengenkripsi ganda (menenkripsi dua kali), jadi plainteks dienkripsi dengan *cipher* I, kemudian hasil enkripsi pertama dienkripsi kembali dengan cipher II yang arah II kebalikan arah I.

Untuk itu maka standard untuk *cipher* ini adalah *cipher* ini adalah cipher ganda yaitu *cipher* yang melakukan enkripsi ganda , yaitu dengan memuat pola enkripsi pertama denganm menegrucut kearah kanan dan enkripsi kedua mengerucut ke arah kiri.

Secara matematis pola enkripsi dapat digambarkan dengan matriks N X N dengan N merupakan panjang plainteks yang akan dienkripsi dan operasi pada alphabet ASCII.

Matriks dilambangkan dengan Mij dengan 1 I N dan j N nilai integer kunci dengan K, Faktor p[engali merupakan tabel integer R. Plainteks dengan p dimana P merupakan tabel plainteks dengan panjang N yaitu P[N].

2.5. ASCII (*American Standard Code for Information Interchane*)

ASCII (American Standard Code for Information Interchane), merupakan suatu standard internasional dalam kode huruf dan symbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter “I’’. ia selalu digunakan oleh computer dan alat komunaksi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 8bit.Dimulai dari 00000000 hingga 11111111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam system bilangan Desimal [9].

PEMBAHASAN

3.1. Analisa dan Logika Metode

Algoritma merkle hellman proses enkripsi dua tahap (ganda) yaitu enkripsi pertama dan enkripsi kedua menggunakan karakter ASCII(*American Standard Code for Information Interchane*) yang panjang karakternyabisa di enkripsikan maksimal 250 karakter. Dimana secara pola enkripsi dapat digambarkan

dengan NxN dengan N merupakan panjang plainteks yang akan dienkripsi dan operasi pada alphabet ASCII.

Langkah selanjutnya Matriks dilambangkan dengan Mij dengan 1 i N dan 1 j N, nilai integer kunci dengan K dan factor pengali merupakan tabel ineger R. Plainteks dengan P dimana P merupakan tabel plainteks dengan panjang N yaitu P[N]

Keterangan :
P = Plainteks
N = Jumlah karakter plainteks
M = Matriks penampung hasil penyandian
K = Kunci
R = Row (baris perkalalian factor pengali dengan kunci)
i = Indeks factor pengali

Algoritma merkle hellman proses enkripsi dua tahap (ganda) yaitu enkripsi pertama dan enkripsi kedua menggunakan karakter ASCII (*American Standard Code for Information Interchane*) yang panjang karakternyabisa di enkripsikan maksimal 250 karakter. Dimana secara pola enkripsi dapat digambarkan dengan NxN dengan N merupakan panjang plainteks yang akan dienkripsi dan operasi pada alphabet ASCII.

Langkah selanjutnya Matriks dilambangkan dengan Mij dengan 1 i N dan 1 j N, nilai integer kunci dengan K dan factor pengali merupakan tabel ineger R. Plainteks dengan P dimana P merupakan tabel plainteks dengan panjang N yaitu P[N]

Keterangan :
P = Plainteks
N = Jumlah karakter plainteks
M = Matriks penampung hasil penyandianK
= Kunci
R = Row (baris perkalalian factor pengali dengan kunci)
i = Indeks factor pengali
j = Indeks karakter plainteks

3.2 Enkripsi Algoritma Merkle Hellman

Adapun tahap- tahap yang dilakukan oleh penulis dalam melakukan analisa terhadap struktur, dapat diuraikan seperti berikut : Proses enkripsi dilakukan dengan dua tahap yaitu enkripsi pertama dan enkripsi ke dua, sehingga dihasilkan cipher akhir yang nantinya menjadi berbentuk data. Penyelesain tahap enkripsi diatas dapat diuraikan melaui contoh kasus penyandian berikut :

1. Matriks enkripsi pertama

Plainteks adalah RIAN
Kunci adalah 3(bilangan integer asli)
Faktor pengali dengan kunci adalah deret bilangan asli(1, 2, 3…,n)

a. Langkah pertama yang dilakukan untuk proses enkripsi pertama ini adalah menentukan nilai desimal masing –masing karakter plainteks dalam ASCII :

R	I	A	N
82	73	65	78

b. Langkah kedua adalah membentukan tabel faktor pengali ;

Seperti pada kasus diatas, maka faktor pengali yang digunakan adalah deretan bilangan asli, Jumlah deret bilangan akan disesuaikan dengan jumlah banyaknya karakter dari plainteks. Jadi jumlah karakter plainteks (N) adalah 4. Deret bilangan asli (R) yang menjadi faktor pengalia adalah 1,2,3,4.

c. Langkah ketiga adalah melakukan proses enkripsi pertama sesuai dengan formulanya.

Plainteks (P) =RIAN
N = 4
K = 3
R = 1,2,3,4.
Untuk baris pertama (i=1), maka :

M11= (P[1]+3*R[1] mod 256
= (R+3*(1)) mod 256
= (82+3) mod 256
= 85 (huruf ‘’U’’ dalam karakter ASCII 256)

M12 = (P[2]+3*R[1] mod 256
= (I+3*(1)) mod 256
= (73+3) mod 256
= 76 (huruf ‘’L’’ dalam karakter ASCII 256)

M13 = (P[3]+3*R[1] mod 256
= (R+3*(1)) mod 256
= (65+3) mod 256
=68 (huruf ‘’D’’ dalam karakter ASCII 256)

M14= (P[4]+3*R[1] mod 256
= (R+3*(1)) mod 256
= (78+3) mod 256
= 81 (huruf ‘’Q’’ dalam karakter ASCII 256)

Hasil sandi pada tahap I = 1 (baris pertama) adalah ULDQ Sampai pada tahap ini hasil penyandian dapat ditunjukkan di bawah ini :
RIAN (nilai desimal dalam ASCII : 82 73 65 78) i = 0
ULDQ (nilai desimal dalam ASCII : 85 76 68 81) i = 1
Hasil penyandian baris pertama (I = 1) akan digunakan sebagai plainteks pada proses enkripsi baris ke dua (i = 2) maka nilai j i, sehingga :
i= 2,j = 2
M22 = (M(2-1)2 + 3*(2)) mod 256
= (M(1)2 + 3*(2)) mod 256
= (L+6) mod 256
= (76+6) mod 256
= 82(huruf ‘’R’’ dalam katakter ASCII 256)

M23 = (M(2-1)3 + 3*(2)) mod 256
= (M(1)3 + 3*(2)) mod 256
= (D+6) mod 256
= (68+6) mod 256
= 74(huruf ''J'' dalam katakter ASCII 256)
M24 = (M(2-1)4 + 3*(2)) mod 256
= (M(1)4 + 3*(2)) mod 256
= (Q+6) mod 256
= (81+6) mod 256
= 87(huruf ''W'' dalam katakter ASCII 256)
Hasil dari enkripsi kedua ini adalah RJW.
Hasil enkripsi sampai pada tahap ini (I = 2) dapat dilihat dibawah ini :
RIAN (nilai desimal dalam ASCII : 82 73 65 78) i = 0
ULDQ (nilai desimal dalam ASCII : 85 76 68 81) i = 1
RJW (dalam nilai ASCII 82 74 87) i = 2
Hasil enkripsi pada baris ke dua (i=1) akan digunakan sebagai plainteks pada proses enkripsi baris ke tiga (1=3), sehingga :
i=3,j=3
M33 = (M(3-1)5 +3*(3)) mod 256
= (M(2)5+3*(3)) mod 256
= (74+9) mod 256
=83 (huruf''S'' dalam karakter ASCII 256)
M34 = (M(3-1)6 +3*(3)) mod 256
= (M(2)6+3*(3)) mod 256
= (87+9) mod 256
=96 (huruf''`'' dalam karakter ASCII 256)
Hasil dari enkripsi baris ketiga(i=3) adalah S'.
Hasil enkripsi sampai pada tahap ketiga (i=3) dapat dilihat dibawah ini :
RIAN (nilai desimal dalam ASCII : 82 73 65 78) i = 0
ULDQ (nilai desimal dalam ASCII : 85 76 68 81) i = 1
RJW (dalam nilai ASCII 82 74 87) i = 2
S` (dalam nilai ASCII) 83 96)i=3
HaSIL enkripsi pada baris ke tiga(i=3) akan digunakan sebagai plainteks pada proses enkripsi baris ke empat(i=4), sehingga :
I=4, j =4
M44 = (M(4-1)6 +3*(4)) mod 256
= (M(3)6+3*(4)) mod 256
= (96+12) mod 256
=108 (huruf''L'' dalam karakter ASCII 256)
Hasil dari enkripsi baris ke tiga ini adalah L.
Hasil enkripsi sampai tahap ke empat(i=4) dapat dilihat dibawah ini
RIAN (nilai desimal dalam ASCII : 82 73 65 78) i = 0
ULDQ (nilai desimal dalam ASCII : 85 76 68 81) i = 1
RJW (dalam nilai ASCII 82 74 87) i = 2
S' (dalam nilai ASCII) 83 96)i=3
L (dala, nilai ASCII 108) I=4
Huruf pertama dari masing-masing baris sebanyak satu karakter sesuai dengan formula Mij pada nilai j = (N+i)-N akan menjadi chiperteks pada proses enkripsi pertama ,sehingga :
RIAN (nilai desimal dalam ASCII : 82 73 65 78) i = 0
ULDQ (nilai desimal dalam ASCII : 85 76 68 81) i = 1
RJW (dalam nilai ASCII 82 74 87) i = 2
S' (dalam nilai ASCII) 83 96)i=3
L (dala, nilai ASCII 108) I=4
Maka yang menjadi cipeher pada proses enkripsi pertama adalah URSL dimana dapat dilihat bahwa susunan dari baris dan kolomnya bebrbentuk yang mengerucut kekiri.

2. Matriks enkripsi ke dua

Langkah- langkah yang dilakukan pada proses enkripsi kedua hampir sama dengan proses enkripsi pertama. Faktor pengali dan kunci yang digunakan tetap sama. Pada proses ini yang mnenjadi plainteks adalah cipher yang dihasilkan dari proses enkripsi pertama URSL kemudia dienkripsi lagi dengan formula yang berlaku pada proses enkripsi ke dua.
Plainteks =U R S L
(cipher hasil enkripsi pertama)
85 82 83 76
(nilai decimal dalam ASCII)
Untuk baris pertama (i=1):
M11 = (p[1] +(3*1)) mod 256
= (U+(3*1)) mod 256
= (85+3) mod 256
= 88(huruf''X'' dalam karakter ASCII 256)

M12 = (p[2] +(3*1)) mod 256
= (R+(3*1)) mod 256
= (82+3) mod 256
= 85(huruf''U'' dalam karakter ASCII 256)

M13 = (p[3] +(3*1)) mod 256
= (S+(3*1)) mod 256
= (83+3) mod 256
= 86(huruf''V'' dalam karakter ASCII 256)

M14 = (p[4] +(3*1)) mod 256
= (L+(3*1)) mod 256
= (76+3) mod 256
= 79(huruf''O'' dalam karakter ASCII 256)

hasil dari enkripsi baris pertama (i=1) adalah xuvo.
Hasil enkripsi sampai pada tahap ini baris pertama(i=1) dapat dilihat dibawah ini:
U R S L (8582 83 76) i = 0
X U V O (88 85 86 79) i = 1
Hasil enkripsi baris pertama (i=1) akan digunakan sebagai plainteks pada proses enkripsi baris ke dua dimana nilai j (N+1)-1 ,sehingga :
i = 2;j (4+1)-2 = j 3
M21 = (M(2-1)1) + (K*R[i])) mod 256
= (M(1)1) + ((K*R[i])) mod 256)

= (X+(3*2)) mod
= (88+6) mod 256
= 94(simbol"^" dalam karakter ASCII)
M22 = (M(2-1)2) + (K*R[i])) mod 256
= (M(1)2) + ((K*R[i])) mod 256)
= (U+(3*2)) mod
= (85+6) mod 256
= 91(huruf"D" dalam karakter ASCII)

M23 = (M(2-1)3) + (K*R[i])) mod 256
= (M(1)3) + ((K*R[i])) mod 256)
= (V+(3*2)) mod
= (86+6) mod 256
= 92(simbol"\" dalam karakter ASCII)
Hasil enkripsi baris ke dua(i=2) D^\.
Hasil enkripsi sampai pada tahap beris ke dua (i=2) dapat dilihat dibawah ini :

U R S L
i = 0
X U V O = (88 85 86 79)i = 1
D ^ \ = (100 94 92) i =2

Hasil enkripsi baris ke dua (i=2) akan digunakan sebagai plainteks pada proses enkripsi baris ke tiga, sehingga :

I=3 ; j (4+1) -3 = j 2
M31 = (M(3-1)1) + (K*R[i])) mod 256
= (M(2)1) + ((K*R[i])) mod 256)
= (D+(3*2)) mod
= (100+6) mod 256
= 106(symbol "j" dalam karakter ASCII)
M32 = (M(3-1)2) + (K*R[i])) mod 256
= (M(2)2) + ((K*R[i])) mod 256)
= (^+(3*2)) mod
= (94+6) mod 256
= 100(symbol "D" dalam karakter ASCII)
Hasil enkripsi baris ke tiga (i=3) adalah Jd.
Hasil enkripsi sdampai pada tahap baris ke tiga(i=3) dapat dilihat dibawah ini :

U R S L
i = 0
X U V O = (88 85 86 79) i = 1
D ^ \ = (100 94 92) i =2
J D= (106 100) I =3

Hasil enkripsi baris ke tiga (i=3) akan digunakan sebgai plainteks pada proses enkripsi baris ke empat sehingga :

I=4; j(4+1)-4 = j 1
M41 = (M(4-1)1) + (K*R[i])) mod 256
= (M(3)1) + ((K*R[i])) mod 256)
= (D+(3*2)) mod
= (106+6) mod 256
= 112(huruf "P" dalam karakter ASCII)
Hasil enkripsi baris ke tiga (i=4) adalah P.
Hasil enkripsi sdampai pada tahap baris ke tiga(i=3) dapat dilihat dibawah ini :

U R S L
i = 0
X U V O = 88 85 86 79 i = 1
D ^ \ = 100 94 92 i =2
J D= 106 100 I =3
P = 112 I = 4

Karakter yang menjadi hasil kedua enkripsia adalah huruf terahir dari masing-masing baris dan diperoleh berdasarkan formula Mij pada nilai j = (N+1)- I,sehingga:

X U V O, i = 1 dan j = (4+1)-1 = 4
D ^ \ i = 2 dan j = (4+1)-2 = 3
J D i = 3 dan j = (4+1)-3 = 2
P i = 4 dan j = (4+1)-4 = 1

Dapat dilihat bahwa hasil penyandian pada proses enkripsi kedua membentuk yang menerucut ke kanan dan menghasilkan cipherteks akhir adalah PD\O. cipher terakhir inilah yang nantinya disimpan menjadi didalam data.

3.3.1. Deskripsi Algoritma Merkle Hellman

Proses deskripsi merupakan kebalikan dari proses enkripsi yang telah dilakukan sebelumnya. Kunci dan factor pengali yang digunakan tetap sama seperti pada proses enkripsi. Proses pengembalian data tersandi kedata asli dilakukan sebanyak dua kali, terdsiri dari enkripsi pertama dan deskripsi kedua. Penyelesaian tahap deskripsi diatas dapat diuraikan melalui contoh kasus dibawah yang dimanan data sandi adalah hasil akhir dari penyandian contoh enkripsi :

1. Dekripsi pertama
Chipertext adalah
p d \ o pada i = 0
80 68 92 79 (nilai decimal dalam ASCII)
I = 1 j (4+1) – 1
J 4
M11 = C[1] –(3*[1])) mod 256
= (P-(3*1)) mod 256
= (80 – 3) mod 256
= 77("M" dalam karakter ASCII)
M12 = C[2] –(3*[1])) mod 256
= (D-(3*1)) mod 256
= (68 – 3) mod 256
= 65("A" dalam karakter ASCII)
M13 = C[3] –(3*[1])) mod 256
= (\-(3*1)) mod 256
= (92 – 3) mod 256
= 89("Y" dalam karakter ASCII)
M14 = C[4] –(3*[1])) mod 256
= (O-(3*1)) mod 256
= (79 – 3) mod 256
= 76("L" dalam karakter ASCII)

Hasil dari deskripsi baris pertama (i= 1) adalah mayl.

Hasil deskripsi sampai pada tahap baris pertama (i=1) dapat dilihat dibawah ini :

p d \ o (80 68 92 79) i=0
m a y l (77 65 89 76) i=1

Hasil deskripsi baris pertama (i=1) akan digunakan sebagai ciphertxt pada proses deskripsi baris ke dua sehingga ,
i=2 j (4+1) – 2

j 3;
M21 =(M(2-1)1 – K*(R[2])) mod 256
=(M(1)1 -3*(2)) mod 256
=(m-6) mod 256
= (77-6) mod 256
= 71(“G” dalam karakter ASCII 256)
M22 =(M(2-1)2 – K*(R[2])) mod 256
=(M(1)2 -3*(2)) mod 256
=(a-6) mod 256
= (65-6) mod 256
= 59(“;” dalam karakter ASCII 256)
M23 =(M(2-1)3 – K*(R[2])) mod 256
=(M(1)3 -3*(2)) mod 256
=(y-6) mod 256
= (89-6) mod 256
= 83(“S” dalam karakter ASCII 256)
Hasil dari deskripsi baris ke dua(i=2) adalah g;s.
Hasil deskripsi sampai pada tahap ke dua(i=2) dapat dilihat dibawah ini :

p d \ o (80 68 92 79) i=0
m a y l (77 65 89 76) i=1
g ; s (71 59 83) i=2

Hasil deskripsi baris ke dua (i=2) akan digunakan sebagai ciphertxt pada proses deskripsi baris ke tiga, sehingga :

i=3; j(4+1) – 3
j 2;

M31 =(M(3-1)1-K*(R[3])) mod 256
=(M(2)1-3*(3)) mod 256
=(G-9) mod 256
=(71-9) mod 256
=62(“>” dalam karakter ASCII 256)
M32 =(M(3-1)2-K*(R[3])) mod 256
=(M(2)2-3*(3)) mod 256
=(;-9) mod 256
=(59-9) mod 256
=50(“2” dalam karakter ASCII 256)
Hasil dari deskripsi baris ke dua(i=2) adalah >2.
Hasil deskripsi sampai pada tahap baris ke tiga (i=3) dapat dilihat dibawah ini :

p d \ o (80 68 92 79) i=0
m a y l (77 65 89 76) i=1
g ; s (71 59 83) i=2
> 2 (62 50) i=3

Hasil deskripsi baris ketiga (i=3) akan digunakan sebagai ciphertext pada proses deskripsi baris ke empat, sehingga:

I=4; j (4+1) -4
J 1 ;

M41 =(M(4-1)1-K*(R[4]) mod 256
=(M(3)1 -3*(4)) mod 256
=(>-12) mod 256
=(62-12) mod 256
=50(“2” dalam karakter ASCII 256)

Hasil darideskripsi baris ke empat(i=4) adalah 2.

Hasil deskripsi sampai pada tahap baris ke empat(i=4) dapat dilihat dibawah ini :

p d \ o (80 68 92 79) i=0
m a y l (77 65 89 76) i=1
g ; s (71 59 83) i=2
> 2 (62 50) i=3
2 (50) i=4

Sehingga pada proses deskripsi pertama diperoleh plainteks sesuai dengan formula Mij pada nilai j =(N+i)-I adalah 22SL.

2. Deskripsi ke dua

Proses deskripsi ke dua merupakan kebalikan dari hasil proses enkripsi pertama. Cipherteks sumber adalah hasil akhir dari proses deskripsi pertama 22sl.
Ciphertext = 2 2 s l (hasil deskripsi pertamas)
50 50 83 76 (nilai decimal dalam ASCII)
Untuk baris pertama (i=1) :
M11 = C[1] – (3*R[1])) mod 256
= (2-(3*1)) mod 256
= (50-3) mod 256
= 47(“/” dalam karakter ASCII 256)
M12 = C[2] – (3*R[1])) mod 256
= (2-(3*1)) mod 256
= (50-3) mod 256
= 47(“/” dalam karakter ASCII 256)
M13 = C[3] – (3*R[1])) mod 256
= (S-(3*1)) mod 256
= (83-3) mod 256
= 80(“P” dalam karakter ASCII 256)
M14 = C[4] – (3*R[1])) mod 256
= (L-(3*1)) mod 256
= (76-3) mod 256
= 73(“I” dalam karakter ASCII 256)
Hasil dari deskripsi pertama (i=1) adalah = / p i.
Hasil deskripsi sampai pada tahap baris pertama(i=1) dapat dilihat dibawah ini :

2 2 s l (50 50 83 76) i=0
/ P I (47 80 73) i=1

Hasil dari deskripsi pertama (i=1) akan digunkan sebagai ciphertext pada proses deskripsi baris ke dua, sehingga :
i=2;j 2;
M22 = (C[2-1]2) – (3*R[2])) mod 256
= (C[1]2 – (3*2)) mod 256
= (/-6) mod 256
= (47-6) mod 256
= 41(“)” dalam karakter ASCII 256)
M23 = (C[2-1]3) – (3*R[2])) mod 256
= (C[1]3 – (3*2)) mod 256
= (P-6) mod 256
= (80-6) mod 256
= 74(“J” dalam karakter ASCII 256)
M24 = (C[2-1]4) – (3*R[2])) mod 256
= (C[1]4 – (3*2)) mod 256
= (I-6) mod 256
= (73-6) mod 256
= 67(“C” dalam karakter ASCII 256)
Hasil dari deskripsi baris pertama (i=1) adalah) J C

Hasil deskripsi sampai tahap baris pertama (i=1) dapat dilihat dibawah ini :

```
2   2  s  l      (50  50 83  76)  i=0
/   P  I         (47  80  73)     i=1
)   J  C         (41  80  67 )    i=2
```

Hasil deskripsi baris ke dua (i=2) akan digunakan sebagai ciphertext pada proses deskripsi baris ke tiga, sehingga :

I=3;j 3;

M33 = (C[3-1]3 – (3*R[3])) mod 256
= (C[2]3 – (3*3)) mod 256
= (J-9) mod 256
=(80-9) mod 256
=71(“G” dalam karakter ASCI 256)

M33 = (C[3-1]4 – (3*R[3])) mod 256
= (C[2]4 – (3*3)) mod 256
= (C-9) mod 256
=(67-9) mod 256
=58(“:” dalam karakter ASCI 256)

Hasil dari deskripsi baris ke tiga (i=3) adalah G :.

Hasil deskripsi sampai pada tahap baris ke tiga (i=3) dapat dilihat dibawah ini :

```
2   2  s  l      (50  50 83  76)  i=0
/   P  I         (47  80  73)     i=1
)   J  C         (41  80  67 )    i=2
G   :     (71  58)                i=3
```

Hasil deskripsi baris ke tiga(i=3) akan digunakan sebagai ciphertext pada proses deskripsi baris ke empat, sehingga :

i=4 ;j 4;

M44 =(C[4-1]4 – (3*R[4])) mod 256
= (C[3]4 – (3*4)) mod 256
= (:-12) mod 256
= (58-12) mod 256
=46(“.” Dalam karakter ASCII 256)

Hasil deskripsi baris kle empat (i=4) adalah ..

Hasil deskripsi sampai pada tahap baris ke empat (i=4) dapat dilihat dibawah ini :

```
2   2  s  l      (50  50 83  76)  i=0
    /  P  I      (47  80  73)     i=1
    )  J  C      (41  80  67 )    i=2
   G   :  (71  58)                i=3
          ` (46)                  i=4
```

Penentuan karakter yang ditetapkan sebagai plainteks (asli) dilakukan berdasarkan formula Mij pada nilai j = (N+i)-N pada masing –masing baris. Sehingga didapatkan Plaintext adalah RIAN (sama seperti teks aslinya).

KESIMPULAN

Berdasarkan pembahasan sebelumnya, maka dapat diambil kesimpulan sebagai Perancangan Aplikasi Enkrisi dan Dekripsi pada Teks menggunakan Algoritma Merkle Hellman sangat membantu setiap pengguna dalam mengamankan data-data yang tidak perlu diketahui pihak-pihak lain. Dimana setiap teks dienkripsikan sebanyak dua kali (secara ganda) sehingga menghasilkan simbol yang berbeda dengan teks aslinya, dan akan kembali lagi kebentuk semula saat pengguna memasukkan kata kunci yang sudah ditetapkan.

DAFTAR PUSTAKA

[1] T. Limbong, “Pengujian kriptografi klasik caesar chipper menggunakan matlab,” no. September 2015, 2017.

[2] D. Ariyus, “PENGANTAR ILMU KRIPTOGRAFI Teori Analisis Dan Informasi, FI,” *Yogyakarta CV ANDI OFFSET*, 2008.

[3] R. Munir, “Kriptografi,” *Inform. Bandung*, 2006.

[4] A. F. Helmi, S. Arifianto, J. T. Informatika, and U. M. Malang, “ANALISA KOMBINASI ALGORITMA MERKLE-HELLMAN KNAPSACK DAN ANALYSIS OF A COMBINATION OF MERKLE-HELLMAN ALGORITHMS AND,” vol. 5, no. 3, pp. 325–334, 2018.

[5] A. Sukmaaji, S. Kom, and R. R. S. Kom, “Jaringan komputer: konsep dasar pengembangan jaringan dan keamanan jaringan,” *Yogyakarta Andi Offset*, 2008.

[6] D. Ariyus, “Kriptografi keamanan data dan komunikasi,” *Yogyakarta Graha Ilmu*, 2006.

[7] R. Sadikin, “Kriptografi untuk keamanan jaringan,” *Penerbit Andi, Yogyakarta*, 2012.

[8] A. Hidayat and R. Rosyadi, “Cryptography Asymmetries Merkle-Hellman Knapsack Digunakan untuk Enkripsi dan Dekripsi Teks,” pp. 27–28, 2016.

[9] I. Wibowo, B. Susanto, and J. Karel T, “Penerapan algoritma Kriptografi Asimetris RSA untuk keamanan data di Oracle,” *J. Inform.*, vol. 5, no. 1, 2009.