ANALISA,DAN PERANCANGAN APLIKASI PENGAMANAN DATA MENGGUNAKAN ALGORITMA MERKLE HELLMAN

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir
Memperoleh Gelar Sarjana pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA	**: MUHAMMAD ARIF**
N. P. M	**: 1314370206**
PROGRAM STUDI	**: SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

LEMBAR PENGESAHAN

ANALISA DAN PERANCANGAN APLIKASI PENGAMANAN MENGGUNAKAN ALGORITMA MERKLE HELLMAN

Disusun Oleh :

Nama	**: MUHAMMAD ARIF**
NPM	**: 1314370206**
Program Studi	**: SISTEM KOMPUTER**
Konsentrasi	**: Keamanan Jaringan Komputer**

Skripsi telah disetujui oleh Dosen Pembimbing Skripsi
Pada tanggal 16 Agustus 2019 :

DOSEN PEMBIMBING I	DOSEN PEMBIMBING II
Dr. Muhammad Iqbal, S.Kom., M.Kom	**Zulham Sitorus, S.Kom., M.Kom**

Mengetahui

Dekan Fakultas Sains Dan Teknologi	Ketua Program Studi Sistem Komputer
Sri Shindi Indira, S.T., M.Sc	**Dr. Muhammad Iqbal, S.Kom., M.Kom**

PERNYATAAN ORISINILITAS

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang secara tertulis diacu dalam skripsi ini dan disebutkan dalam daftar pustaka.

Medan Agustus 2019

Muhammad Arif

1314370206

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama	: MUHAMMAD ARIF
NPM	: 131470206
Prodi	: SISTEM KOMPUTER
Konsentrasi	: KEAMANAN JARINGAN KOMPUTER (KJK)
Judul Skripsi	: ANALISA DAN PERANCANGAN APLIKASI PENGAMANAN DATA MENGGUNAKAN ALGORITMA MERKLE HELLMAN

Dengan ini menyatakan bahwa :

1. Tugas Akhir/Skripsi saya bukan hasil Plagiat
2. Sayat tidak akan menuntut perbaikan nilai indeks Prestasi Kumulatif (IPK) setelah ujian Sidang Meja Hijau
3. Skripsi saya dapat dipublikasikan oleh pihak lembaga, dan saya tidak akan menuntut akibat publikasi tersebut

Demikian pernyataan ini saya perbuat dengan sebenar-benarnya, terima kasih

Medan, Agustus 2019

Yang membuat pernyataan

MUHAMMAD ARIF

1314370206

Telah Diperiksa oleh LPMU
dengan Plagiarisme.....47.....%
Tanggal 30 JULI 2019
THAMRIZI HAKIM

FM-BPAA-2012-041

Hal : Permohonan Meja Hijau

Medan, 29 Juli 2019
Kepada Yth : Bapak/Ibu Dekan
Fakultas SAINS & TEKNOLOGI
UNPAB Medan
Di -
Tempat

Telah di terima berkas persyaratan dapat di proses
Medan, 29 JUL 2019
An. Ka. BPAA
TEGUH WAHYONO, SE, MM.

Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama	: MUHAMMAD ARIF
Tempat/Tgl. Lahir	: MEDAN / Medan
Nama Orang Tua	: ERWANTO
N. P. M	: 1314370206
Fakultas	: SAINS & TEKNOLOGI
Program Studi	: Sistem Komputer
No. HP	: 082273939633
Alamat	: 22 Nopember 1995

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul **ANALISA DAN PERANCANGAN APLIKASI PENGAMANAN DATA MENGGUNAKAN ALGORITMA MERKLE HELLMAN**, Selanjutnya saya menyatakan :

1. Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
2. Tidak akan menuntut ujian perbaikan nilai mata kuliah untuk perbaikan indek prestasi (IP), dan mohon diterbitkan ijazahnya setelah lulus ujian meja hijau.
3. Telah tercap keterangan bebas pustaka
4. Terlampir surat keterangan bebas laboratorium
5. Terlampir pas photo untuk ijazah ukuran 4x6 = 5 lembar dan 3x4 = 5 lembar Hitam Putih
6. Terlampir foto copy STTB SLTA dilegalisir 1 (satu) lembar dan bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkipnya sebanyak 1 lembar.
7. Terlampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
8. Skripsi sudah dijilid lux 2 examplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 examplar untuk penguji (bentuk dan warna penjilidan diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangani dosen pembimbing, prodi dan dekan
9. Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
10. Terlampir surat keterangan BKKOL (pada saat pengambilan ijazah)
11. Setelah menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
12. Bersedia melunaskan biaya-biaya uang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan perincian sbb :

1.	[102] Ujian Meja Hijau	: Rp.	0
2.	[170] Administrasi Wisuda	: Rp.	1,500,000
3.	[202] Bebas Pustaka	: Rp.	100,000
4.	[221] Bebas LAB	: Rp.	5,000
	Total Biaya	: Rp.	1,605,000

20/ Juli 2019

Ukuran Toga : XXL

Diketahui/Disetujui oleh :
Sri Shindi Indira, S.T., M.Sc.
Dekan Fakultas SAINS & TEKNOLOGI

Hormat saya
MUHAMMAD ARIF
1314370206

Catatan :

- 1.Surat permohonan ini sah dan berlaku bila ;
  - a. Telah dicap Bukti Pelunasan dari UPT Perpustakaan UNPAB Medan.
  - b. Melampirkan Bukti Pembayaran Uang Kuliah aktif semester berjalan
- 2.Dibuat Rangkap 3 (tiga), untuk - Fakultas - untuk BPAA (asli) - Mhs.ybs.

TANDA BEBAS PUSTAKA
No. 436/perp/BP/2019
Dinyatakan tidak ada sangkut paut dengan UPT. Perpustakaan
Medan, 29 JUL 2019
Ka. UPT. Perpustakaan
Dra. Arisandi, M.Hum

Hal : Surat Keterangan Pengganti Form Permohonan Pengajuan Judul Skripsi

Kepada YTH
Bapak Ka. Prodi Sistem Komputer
Universitas Pembangunan Panca Budi
Di Medan

Dengan Hormat,

Saya yang bertanda tangan di bawah ini :

Nama	: Muhammad Arif
NPM	: 1314370206
Fakultas	: Sains dan Teknologi
Prodi	: SI/Sistem Komputer

Dengan ini memohon kepada bapak Ka. Prodi Sistem Komputer agar sudi kiranya menyetujui surat kterangan pengganti Form Permohonan Pengajuan Judul Skripsi saya yang hilang.Demikian surat permohonan ini saya ajukan, dan atas perhatian bapak saya ucapkan terimaksih.

Pemohon

Muhammad Arif

Ka. Prodi Sistem Komputer

Dr. Muhammad Iqbal S.Kom., M.kom

UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI
Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
Medan - Indonesia

iversitas : Universitas Pembangunan Panca Budi
kultas : SAINS & TEKNOLOGI
sen Pembimbing I : MUHAMMAD IQBAL S.Kom M.Kom
sen Pembimbing II : ZULHAM SITORUS
ma Mahasiswa : MUHAMMAD ARIF
usan/Program Studi : Sistem Komputer
mor Pokok Mahasiswa : 1314370206
jang Pendidikan : S1
tul Tugas Akhir/Skripsi : ANALISA DAN PERANCANGAN APLIKASI PENGAMANAN DATA MENGGUNAKAN ALGORITMA MERKLE HELLMAN

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
4/04 2019	Acc Judul.		
	Acc Bab I		
8/04 2019	Acc Bab II.		
10/04 2019	Acc Bab III.		
12/04 2019	Acc Bab IV & V		
15/04 2019	Acc Seminar.		
16/04 2019	Acc Sidang		

Medan, 12 April 2019
Diketahui/Disetujui oleh :
Dekan,

UNIVERSITAS PEMBANGUNAN PANCA BUDI
UNPAB
INDONESIA
FAKULTAS SAINS & TEKNOLOGI

Sri Shindi Indira, S.T.,M.Sc.

t yang tidak perlu

UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI
Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
Medan - Indonesia

iversitas : Universitas Pembangunan Panca Budi
kultas : SAINS & TEKNOLOGI
sen Pembimbing I : MUHAMMAD IQBAL S.Kom M.Kom
sen Pembimbing II : ZULHAM SITORUS
ma Mahasiswa : MUHAMMAD ARIF
rusan/Program Studi : Sistem Komputer
mor Pokok Mahasiswa : 1314370206
njang Pendidikan : ST
dul Tugas Akhir/Skripsi : ANALISA DAN PERANCANGAN APLIKASI PENGAMANAN DATA MENGGUNAKAN ALGORITMA MERKLE HELLMAN

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
14	Acc Seminar proposal		
14	Revisi judul dan dafpus		
14	Konsep bab II di perbaiki		
14	Acc I, II, III. Lanjut IV+V		
14	Analisa lanjut ujian		
16.	Acc. Seminar hasil		
16	Acc. Sidang meja hijau		
14	Acc Jilid		

Medan, 12 April 2019
Diketahui/Disetujui oleh :
Dekan

Sri Shindi Indira, S.T.,M.Sc.

et yang tidak pertu

file:///C:/Users/LPMU/Documents/Plagiarism Detector reports/originality report 20.5.2019 12-37-30 - MUHAMMAD ARIF_1314

Plagiarism Detector v. 1092 - Originality Report:

Analyzed document: 20/05/2019 12:37:30

"MUHAMMAD ARIF_1314370206_SISTEM KOMPUTER.docx"

Licensed to: **Universitas Pembangunan Panca Budi_License4**

UNIVERSITAS PEMBANGUNAN PANCA BUDI
Beriman, Berilmu, Berkarya

LEMBAGA PENJAMIN MUTU UNIVERSITAS
UNIVERSITAS PEMBANGUNAN PANCA BUDI

Relation chart:

Distribution graph:

Comparison Preset: Rewrite. Detected language: Indonesian

13:41
20/05/2019

YAYASAN PROF. DR. H. KADIRUN YAHYA
UNIVERSITAS PEMBANGUNAN PANCA BUDI
LABORATORIUM KOMPUTER
Jl. Jend. Gatot Subroto Km 4,5 Sei Sikambing Telp. 061-8455571
Medan - 20122

KARTU BEBAS PRAKTIKUM

Yang bertanda tangan dibawah ini Ka. Laboratorium Komputer dengan ini menerangkan bahwa :

Nama	:	MUHAMMAD ARIF
N.P.M.	:	1314370206
Tingkat/Semester	:	Akhir
Fakultas	:	SAINS & TEKNOLOGI
Jurusan/Prodi	:	Sistem Komputer

Benar dan telah menyelesaikan urusan administrasi di Laboratorium Komputer Universitas Pembangunan Panca Budi Medan.

Medan, 29 Juli 2019
Ka. Laboratorium

Fachrid Wadly, S.Kom

ABSTRAK

MUHAMMAD ARIF

ANALISA DAN PERANCANGAN APLIKASI PENGAMANAN MENGGUNAKAN ALGORITMA MERKLE HELLMAN

2019

Sistem keamanan data dan kerahasiaan data merupakan salah satu aspek yang penting dalam perkembangan teknologi khususnya komunikasi yang menggunakan komputer,namun komunikasi pesan yang digunakan tersebut belum tentu aman. Maka kriptografi merupakan salah satu teknik yang digunakan dalam pengamanan data agar untuk menjaga kerahasiaan, keamanan, atau keotentikan suatu pesan yang kita kirim tidak dibaca oleh orang yang tidak berhak menerimanya.

Algoritma Merkle Hellman merupakan salah satu jenis stream cipher yang sinkron yaitu cipher yang memiliki kunci simetris dan mengenkripsi atau mendekripsi plainteks secara bit per bit dengan cara mengkombinasikan secaraoperasi biner (biasanya operasi XOR). Implementasi algoritma Algoritma Merkle Hellman pada perangkat lunak yang dirancang menggunakan Microsoft Visual.Net 2008 yang dapat diterapkan pada algoritma Algoritma Merkle Hellman untuk enkripsi dan dekripsi pada sebuah file teks.

Hasil penelitian ini penggunaan Algoritma Merkle Hellman ini cukup mudah untuk digunakan secara luas pada berbagai aplikasi dan algoritmanya dinyatakan cukup aman untuk pengamanan data. Sehingga data yang dirimkan tidak mudah dicuri dan tidak mudah dipecahkan sebagai kemanan pada data-data yang penting.

Kata Kunci : Kriptografi, Enkripsi, Dekripsi, Algoritma Merkle Hellman

DAFTAR ISI

KATA PENGANTAR

Assalamua"laikum Wr. Wb.

Alhamdulillahirobbil"alamin, segala puji bagi Allah SWT yang senantiasa memberikan rahmat dan karunia-Nya dan junjungan besar Nabi Muhammad SAW yang telah memberikan segala rahmat dan nikmat kepada penulis, sehingga penulis dapat menyelesaikan skripsi ini yang berjudul: **Analisa dan Perancangan Aplikasi Pengamanan Data Menggunakan Algoritma Merkle Hellman.** Dimana skripsi ini merupakan tugas dan syarat untuk memperoleh gelar Sarjana Strata 1 (S1) Fakultas Sains dan teknologi, Jurusan Sistem Komputer. Penulisan skripsi ini merupakan kewajiban bagi setiap mahasiswa yang akan mengakhiri masa kuliahnya pada setiap perguruan tinggi seperti halnya pada perguruan tinggi swasta Universitas Pembangunan Panca Budi Medan. Dengan penulisan skripsi ini diharapkan dapat meningkatkan mutu suatu perguruan tinggi.

Walaupun penulis telah berusaha semaksimal mungkin dalam mengerjakan skripsi ini, namun penulis menyadari sepenuhnya bahwa penulisan ini masih banyak kekurangan dan jauh dari kata sempurna. Maka dari itu penulis menerima saran dan nasehat dari pembaca guna perbaikan dan penyempurnaan isi dari skripsi ini.

Penyusunan skripsi ini tidak dapat terselesaikan atas bantuan dari beberapa pihak, terutama penulis ingin menyampaikan rasa hormat dan cinta kepada kedua orang tua yang telah mendukung selama proses penulisan skripsi ini. Ayah Erwanto yang selalu menghibur selama mengerjakan skripsi ini, dan ibuku sayang, Bu Diani yang selalu memberi semangat dan mendoakan anaknya ini supaya dapat menyelesaikan skripsi dengan cepat. Terima kasih kepada kakak dan Abang tercinta Suci Maulida dan Maulana anshari yang sudah mau menghibur di saat penulis merasa jenuh, kepada Ekky Nusantari yang selalu memberikan semangat tanpa henti serta untuk membantu penulis baik secara moril maupun materil demi menyelesaikan kuliah ini, terkhusus selama masa penulisan skripsi yang sangat melelahkan dan membutuhkan banyak pengorbanan pula. Penulis berharap nantinya skripsi ini paling tidak bisa membuat bangga semua keluarga tercinta. Semoga Allah SWT selalu mencurahkan rakhmat dan kasih sayang-Nya kepada kita sekeluarga. Amin ya Rabbal'alamin.

Tak lupa untuk itu sudah menjadi keharusan penulis mengucapkan rasa terima kasih yang sebesar-besarnya juga kepada:

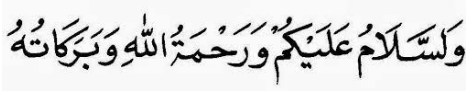
1. Bapak Dr. H. Muhammad Isa Indrawan, S.E, M.M selaku Rektor Universitas Pembangunan Panca Budi Medan.
2. Bapak Ir. Bhakti Alamsyah, M.T, Ph.D selaku Rektor I Universitas Pembangunan Panca Budi Medan.
3. Bapak Dra. Hj. Irma Fatmawati, S.H, M.Hum selaku Rektor II Universitas Pembangunan Panca Budi Medan.

4. Bapak Samrin, S.E, M.M selaku Rektor III Universitas Pembangunan Panca Budi Medan.
5. Ibu Sri Shindi Indira, ST., M.Sc. Selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
6. Bapak Dr. Muhammad Iqbal, S.Kom., M.Kom selaku Ketua Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
7. Bapak Dr. Muhammad Iqbal, S.Kom., M.Kom selaku Dosen Pembimbing I yang selalu membimbing, mendidik, mendukung dan memberi masukan dalam menyelesaikan skripsi ini.
8. Bapak Zulham Sitorus, S. Kom., M.Kom selaku Dosen Pembimbing II yang selalu membimbing, mendidik, mendukung dan memberi masukan dalam menyelesaikan skripsi ini.
9. Seluruh Dosen Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan yang tidak bisa penulis sebutkan satu persatu, terimakasih telah banyak memberikan ilmu dan masukan bagi penulis bagi penulis selama masa perkuliahan.
10. Seluruh Pegawai Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan yang telah mengarahkan penulis tentang proses perkuliahan yang berlangsung selama ini.
11. Semua pihak yang telah banyak membantu dalam penyusunan skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Semoga pihak yang telah membantu penulis dalam proses penyusunan skripsi mendapat balasan yang sebesar-besarnya dari allah SWT. Penulis mohon maaf jika penulis belum mampu membalas kalian.

Adapun penyusunan skripsi ini masih jauh dari kata sempurna, sehingga apabila di temukan kesalahan dan kekurangan di dalamnya, penulis mohon maaf sekali lagi. Karena penulis hanyalah manusia biasa dan jauh dari kata sempurna, karena kesempurnaan itu hanyalah milik Allah semata.

Akhir kata, semoga skripsi ini dapat bermanfaat bagi pembaca dan khasanah ilmu pengetahuan pada umumnya.



Medan, Agustus 2019

Hormat saya

Muhammad Arif
1314370206

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi komputer dan teknologi telekomunikasi pada saat ini telah mengubah cara masyarakat dalam berkomunikasi. Dulu, komunikasi jarak jauh masih dilakukan dengan cara konvensional, yaitu dengan cara saling mengirim surat. Sekarang, dengan adanya internet, komunikasi jarak jauh bisa dilakukan dengan cara saling mengirim email atau sms (short message service). Internet juga telah membuat komunikasi semakin terbuka dan pertukaran informasi juga semakin cepat melewati batas-batas negara dan budaya. Namun tidak semua perkembangan teknologi komunikasi ini memberikan dampak yang menguntungkan bagi dunia komunikasi. Penyadapan data merupakan hal yang paling ditakuti oleh pengguna jaringan komunikasi pada saat ini

Keamanan suatu informasi merupakan hal yang perlu diperhatikan dalam menjaga kerahasiaan informasi itu sendiri, terutama bila informasi tersebut hanya boleh diketahui pihak yang tertentu saja. Pengiriman data atau informasi tanpa dilakukan pengamanan akan beresiko terhadap penyadapan. Sehingga informasi yang ada di dalamnya dapat mudah diketahui oleh pihak-pihak yang tidak berhak, dan hal seperti itu sangat merugikan.Hingga saat ini, kriptografi merupakan salah satu solusi untuk menjamin keamanan dari suatu informasi. Kriptografi merupakan metode dengan menyandikan isi informasi (*plaintext*) menjadi isi yang sulit atau bahkan tidak dipahami melalui proses enkripsi. Untuk memperoleh

kembali informasi yang asli dapat dilakukan dengan proses dekripsi, yang tentunya dengan menggunakan kunci yang benar.

Untuk melindungi akses data 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. Algoritma yang digunakan disini adalah Algoritma Merkle Hellman. Sistem kriptografi Merkle Hellman pertama kali ditemukan oleh Merkle dan Hellman pada tahun 1978. Meskipun telah terpecahkan pada permulaan 1980-an, sistem kripto ini berikut beberapa variannya masih bermanfaat untuk dipelajari terutama berkenaan dengan keluwesan konseptual serta teknik desain yang mendasarinya. Algoritma Merkle Hellman menggunakan kunci Asimetris dalam proses operasi enkripsi dan dekripsinya.

Berdasarkan uraian di atas maka penulis membuat judul skripsi dengan judul ”**ANALISA DAN PERANCANGAN APLIKASI PENGAMANAN DATA MENGGUNAKAN ALGORITMA MERKLE HELLMAN**”

1.2 Perumusan Masalah

Berdasarkan uraian pada latar belakang diatas maka dapat diuraikan sebagai berikut :

1. Bagimana proses algoritma Merkle Hellman dalam melakukan enkripsi dan dekripsi dengan cara memakai kunci asimetrik ?

2. Bagaimana algoritma akan diimplementasikan ke dalam bahasa pemrograman sehingga menjadi sebuah aplikasi yang bisa digunakan untuk enkripsi dan dekripsi file teks ?
3. Bagaimana perancangan dan pengujian aplikasi kriptografi Merkle Hellman ?

1.3 Batasan Masalah

Dalam penyusunan skripsi ini, penulis membuat membatasi masalah sebagai berikut:

1. Algoritma yang digunakan adalah Algoritma Merkle Hellman.
2. Format teks yang dapat dienkripsi adalah .txt dan .rtf.
3. Panjang karakter yang bisa di enkripsi maksimal 250 karakter dan jumlah putaran sebanyak 4 kali putaran.
4. Bahasa pemrograman yang digunakan adalah *Visual Basic*.

1.4 Tujuan dan Manfaat

Tujuan dari penulisan skripsi ini adalah :

1. Untuk mengetahui proses enkripsi dan dekripsi algoritma Merkle Hellman dengan cara memakai kunci asimetrik.
2. Untuk mengetahui cara mengimplementasikan algoritma Merkle Hellman ke dalam bahasa pemrograman sehingga menjadi sebuah aplikasi yang bisa digunakan untuk enkripsi dan dekripsi file.

3. Untuk melakuan pengujian aplikasi kriptografi Merkle Hellman, sehingga dapat diketahui kelebihan dan kelemahannya.

Manfaat penulisan skripsi ini adalah :

1. Dapat membantu pengguna dalam melakukan enkripsi teks miliknya, sehingga tidak dapat diketahui pihak lain yang tidak berhak pada saat pengirimannya.
2. Dapat mempermudah pemahaman enkripsi dan dekripsi teks menggunakan algoritma Merkle Hellman bagi pihak-pihak yang membutuhkannya.
3. Dapat menghasilkan sebuah aplikasi yang dapat digunakan untuk enkripsi dan dekripsi teks mneggunakan algoritma Merkle Hellman.

1.5 Sistematika Penulisan

Pembahasan skripsi ini secara garis besar dibagi menjadi 5 bab penulisan, adapun bab demi bab dalam skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Dalam bab ini akan dijelaskan mengenai latar belakang identifikasi masalah, tujuan pembahasan, batasan masalah dan sistematika penulisan.

BAB II : LANDASAN TEORI

Dalam bab ini dibahas teori-teori yang memuat penjelasan yang berhubungan dengan perancangan perangkat lunak dan mengenai

aplikasi perangkat ajar pembelajaran dan algoritma Merkle Hellman.

BAB III : PEMBAHASAN DAN PERANCANGAN

Dalam bab ini membahas tentang proses kerja dari algoritma Merkle Hellmanperancangan tampilan antarmuka dari perangkat lunak pembelajaran komputer.

BAB IV : ALGORITMA DAN IMPLEMENTASI

Dalam bab ini membahas bagaimana membuat aplikasi pembelajaran, dan implementasi dari perangkat lunak yang mencakup spesifikasi software dan hardware serta pengujian program.

BAB V : KESIMPULAN DAN SARAN

Dalam bab ini memuat tentang kesimpulan dan saran-saran yang dapat diambil setelah menyelesaikan skripsi ini untuk pengembangan lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Algoritma

Istilah algoritma pertama kali di perkenalkan oleh seorang ahli matematika yaitu Abu Ja'far Muhammad Ibnu Musa Al Khawarizmi.Algoritma adalah urutan dari barisan instruksi untuk menyelesaikan suatu masalah. Ada pun algoritma dapat dinyatakan dalam bentuk *flowchart*, diagram alir, bahasa semu sedangkan secara bahasa, algoritma berarti suatu metode khusus untuk menyelesaikan suau masalah yang nyata (Doni Ariyus, 2008).

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Menurut Kamus Besar Bahasa Indonesia (KBBI) terbitan Balai Pustaka Tahun 1988, algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah. Kata logis merupakan kata kunci dalam sebuah algoritma. Langkah-langkah di dalam algoritma harus logis, berarti hasil dari urutan langkah-langkah tersebut harus dapat ditentukan, benar atau salah.

Dalam bidang pemrograman algoritma didefinisikan sebagai metode khusus yang tepat dan terdiri dari serangkaina langkah yang terstruktur dan dituliskan secara sistematis yang akan dikerjakan untuk menyelesaikan suatu masalah dengan bantuan komputer.

Ada lima komponen utama dalam algoritma yaitu *finiteness*, *definiteness*, *input*, *output* dan *effectiveness* (Knuth, 1973) :

1. *Finiteness*.

   Sebuah algoritma harus selalu berakhir setelah sejumlah langkah berhingga.

2. *Definiteness*.

   Setiap langkah dari sebuah algoritma harus didefinisikan secara tepat, tindakan yang di muat harus dengan teliti dan sudah jelas ditentukan untuk setiap keadaan.

3. *Input*.

   Sebuah algoritma memiliki nol atau lebih masukan, sebagai contoh, banyaknya masukan diberikan di awal sebelum algoritma mulai.

4. *Output*.

   Sebuah algoritma memiliki satu atau lebih keluaran, sebagai contoh, banyaknya keluaran memiliki sebuah hubungan yang ditentukan terhadap masukan.

5. *Effectiveness*.

   Pada umumnya sebuah algoritma juga diharapkan untuk efektif.

Beberapa notasi yang digunakan dalam penulisan algoritma (Munir, 2002), antara lain :

1. Notasi I, menyatakan langkah-langkah algoritma dengan untaian kalimat deskriptif.

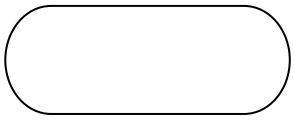
   Dengan notasi bergaya kalimat ini, deskripsi setiap langkah dijelaskan dengan bahasa yang gamblang. Proses diawali dengan kata kerja seperti ‘baca’, ‘hitung’, ‘bagi’, ‘ganti’, dan sebagainya, sedangkan pernyataan kondisional dinyatakan dengan ‘jika … maka …’. Notasi ini bagus untuk algoritma yang pendek, namun untuk masalah yang algoritmanya besar, notasi ini jelas tidak

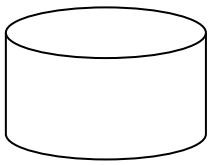
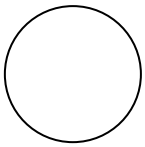
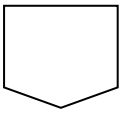
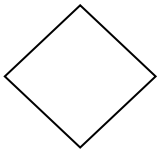
mangkus. Selain itu, pengkonversian notasi algoritma ke notasi bahasa pemrograman relatif sukar.

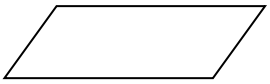
2. Notasi II, menggunakan diagram alir (*flow chart*)

Flow chart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flow chart* menolong analis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternative-alternatif lain dalam pengoperasian. *Flow chart* biasanya mempermudah penyelesaian suatu masalah, khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. (Normalina dan Nikous, 2010). Notasi ini menggunakan sejumlah simbol untuk menyatakan kegiatan-kegitan secara keseluruhan. Sama halnya dengan notasi deskriptif, notasi ini cocok untuk algoritma yang pendek, namun untuk masalah yang algoritmanya besar, notasi ini jelas tidak efektif. Selain itu, pengkonversian notasi algoritma ke notasi bahasa pemrograman cenderung relatif sukar, dapat dilihat pada table 2.1.

Tabel 2.1 Simbol Bagan Alir (*Flowchart*)

Simbol	Fungsi
	Notasi *Terminator* Menunjukkan awal dari suatu perancangan sistem baik secara manual maupun komputer.

	*Magnetic Disk (hard disk)* Menunjukkan media penyimpanan data yang menunjukkan *input/output* yang menggunakan *hard disk*.
	*Connector* (penghunbung dalam suatu halaman) Digunakan untuk menghubung ke halaman yang sama.
	*Connector off-page* Digunakan untuk penghubung ke halaman lain.
	Manual *input* (pemasukkan data dari keyboard) Digunakan untuk *input/output* yang menggunakan *online keyboard.*
	Deklarasi *variable* Menunjukkan awal proses data sebagai pengenalan notasi *field*.
	*Decision* (keputusan) Menunjukkan proses *logical* perbandingan dalam memberikan keputusan yang benar (*true*) atau salah (*false*).
	Proses dengan metode komputerisasi. Menunjukkan kegiatan proses operasi program komputer.

	Arsip Digunakan sebagai *multi document* sebagai simbol arsip data secara manual.
	Notasi dokument (laporan) Menunjukkan dokument yang digunakan untuk *input* dan *output* baik secara manual, maupun komputerisasi.
	Digunakan sebagai *input/output* data notasi *field*.
	Proses manual menunjukkan pekerjaan yang dilakukan secara manual.
	Store Data merupakan symbol proses penyetoran data ke dalam database

sumber : Janner Simarmata dan Iman Paryudi, 2005

2.2 Kriptografi

Penjagaan sebuah informasi sangatlah diperlukan agar tidak jatuh ke tangan orang-orang yang tidak berhak untuk mengaksesnya. Teknik kriptografi telah banyak digunakan sebagai salah satu cara untuk menciptakan sebuah informasi yang selalu terjaga keabsahannya (Dony Ariyus, 2005).

2.2.1 Defenisi Kriptografi

Kriptografi pada awalnya merupakan ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Kemudian seiring dengan berkembangnya kriptografi yaitu kriptografi tidak lagi sebatas mengenkripsikan pesan, tetapi juga memberikan aspek keamanan yang lain seperti serangan dari kriptanalisis. Oleh karena itu pengertian kriptografi pun berubah menjadi ilmu sekaligus seni untuk menjaga keamanan pesan.

Cryptography (kriptografi) berasal dari bahasa Yunani yaitu dari kata *crypto* yang berarti penulisan *screet* (rahasia), sedangkan *graphein* artinya *writing* (tulisan). Jadi secara sederhana dapat diartikan *screetwriting* (tulisan rahasia). Definisi lain dari kriptografi adalah sebuah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data serta otentikasi (Rinaldi Munir, Kriptografi, 2006).

Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain (Dony Ariyus, 2005).

2.2.2 Sejarah Kriptografi

Kriptografi memiliki sejarah yang panjang dan mengagumkan. Penulisan rahasia ini dapat dilacak kembali ke 3000 tahun SM saat digunakan oleh bangsa Mesir. Mereka menggunakan hieroglyphcs untuk menyembunyikan tulisan dari mereka yang tidak diharapkan. *Hieroglyphcs* diturunkan dari bahasa Yunani

hieroglyphica yang berarti ukiran rahasia. *Hieroglyphs* berevolusi menjadi *hieratic*, yaitu *stylized script* yang lebih mudah untuk digunakan. Sekitar 400 SM, kriptografi militer digunakan oleh bangsa Spartan dalam bentuk sepotong papirus atau perkamen dibungkus dengan batang kayu. Sistem ini disebut *Scytale* (Dony Ariyus, 2005).

Sekitar 50 SM, Julius Caesar, kaisar Roma, menggunakan *cipher* substitusi untuk mengirim pesan ke Marcus Tullius Cicero. Pada *cipher* ini, huruf-huruf alfabet disubstitusi dengan huruf-huruf yang lain pada alfabet yang sama. Karena hanya satu alfabet yang digunakan, *cipher* ini merupakan substitusi *monoalfabetik*. *Cipher* semacam ini mencakup penggeseran alfabet dengan 3 huruf dan mensubstitusikan huruf tersebut. Substitusi ini kadang dikenal dengan C3 (untuk Caesar menggeser 3 tempat).

Disk mempunyai peranan penting dalam kriptografi sekitar 500 tahun yang lalu. Di Italia sekitar tahun 1460, Leon Battista Alberti mengembangkan disk *cipher* untuk enkripsi, sistemnya terdiri dari dua *disk* konsentris. Setiap *disk* memiliki alfabet di sekelilingnya, dan dengan memutar satu disk berhubungan dengan yang lainnya, huruf pada satu alfabet dapat ditransformasi ke huruf pada alfabet yang lain.

Bangsa Arab menemukan *cryptanalysis* karena kemahirannya dalam bidang matematika, statistik, dan lingiustik. Karena setiap orang muslim harus menambah pengetahuannya, mereka mempelajari peradaban terdahulu dan mendekodekan tulisan-tulisannya ke huruf-huruf Arab. Pada tahun 815, Caliph al-

Mamun mendirikan House of Wisdom di Baghdad yang merupakan titik pusat dari usaha-usaha translasi. Pada abad ke-9, filsuf Arab al-Kindi menulis risalat (ditemukan kembali th 1987) yang diberi judul "*A Manuscript on Deciphering Cryptographic Messages*".

Pada 1790, Thomas Jefferson mengembangkan alat enkripsi dengan menggunakan tumpukan yang terdiri dari 26 disk yang dapat diputar secara individual. Pesan dirakit dengan memutar setiap disk ke huruf yang tepat dibawah batang berjajar yang menjalankan panjang tumpukan disk. Kemudian, batang berjajar diputar dengan sudut tertentu, A, dan huruf-huruf dibawah batang adalah pesan yang terenkripsi. Penerima akan menjajarkan karakter-karakter *cipher* di bawah batang berjajar, memutar batang kembali dengan sudut A dan membaca pesan *plaintext*.

Sistem disk digunakan secara luas selama perang sipil US. Federal Signal Officer mendapatkan hak paten pada sistem disk mirip dengan yang ditemukan oleh Leon Battista Alberti di Italia, dan dia menggunakannya untuk mengkode dan mendekodekan sinyal-sinyal bendera diantara unit-unit.

Sistem Unix menggunakan *cipher* substitusi yang disebut ROT 13 yang menggeser alfabet sebanyak 13 tempat. Penggeseran 13 tempat yang lain membawa alfabet kembali ke posisi semula, dengan demikian mendekodekan pesan.

Mesin kriptografi mekanik yang disebut *Hagelin Machine* dibuat pada tahun 1920 oleh Boris Hagelin di Scockholm, Swedia. Di US, mesin Hagelin

dikenal sebagai M-209. Pada tahun 20-an, Herbert O. Yardley bertugas pada organisasi rahasia US MI-8 yang dikenal sebagai *Black Chamber*. MI-8 menjebol kode-kode sejumlah megara. Selama konferensi Angkatan Laut Washington tahun 1921-1922, US membatasi negosiasi dengan Jepang karena MI-8 telah memberikan rencana negosiasi Jepang yang telap disadap kepada sekretaris negara US. Departemen negara menutup MI-8 pada tahun 1929 sehingga Yardley merasa kecewa. Sebagai wujud kekecewaanya, Yardley menerbitkan buku The American Black Chamber, yang menggambarkan kepada dunia rahasia dari MI-8. Sebagai konsekuensinya, pihak Jepang menginstal kode-kode baru. Karena kepeloporannya dalam bidang ini, Yardley dikenal sebagai "Bapak Kriptografi Amerika".

2.2.3 Tujuan Kriptografi

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu (Rinaldi Munir, 2006):

1. Kerahasiaan (*confidentiality*), adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
2. Integritas data (*data integrity*), adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

3. Outentikasi (*authentication*), adalah usaha yang berhubungan dengan identifikasi/pengenalan, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication* atau *entity authentication*) maupun mengidentifikasi kebenaran sumber pesan (data *origin authentication*).
4. Nirpenyangkalan (*non-repudiasi*) adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

Dari keenam aspek keamanan data tersebut, empat diantaranya dapat diatasi dengan menggunakan *cryptography* yaitu *confidentiality*, *integrity*, *authentication*, dan *nonrepudiation.*

2.2.4 Algoritma Kriptografi

Algoritma kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut. Algoritma kriptografi terdiri dari tiga fungsi dasar (Doni Ariyus, 2008) :

1. Enkripsi

   Merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirim agar terjaga kerahasiaannya. Pesan asli disebut *plaintext*, yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan sebagai *cipher* atau kode dengan menggunakan algoritma yang untuk mengkodekan data yang kita inginkan.

2. Dekripsi

   Merupakan kebalikan dari proses enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (teks asli), disebut dengan dekripsi pesan. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan algoritma yang digunakan untuk enkripsi.

3. Kunci

   Kuci adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*).

Bedasarkan kunci yang dipakai dalam proses kriptografi, maka algoritma kriptografi dibagi menjadi (Doni Ariyus, 2008) :

1. Algoritma Simetri

   Algoritma ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsi. Bila mengirim pesan dengan menggunakan algoritma ini, si penerima pesan harus diberitahu kunci dari pesan tersebut agar bisa mendekripsikan pesan yang dikirim. Keamanan dari pesan yang menggunakan algoritma ini tergantung pada kunci. Algoritma yang menggunakan kunci simteris misalnya DES, Kode Rivest's, IDEA, AES, OTP, A5 dan lain-lain.

2. Algoritma Asimetri

   Algoritma asimetri sering juga disebut dengan algoritma kunci publik, dengan arti kata kunci yang digunakan untuk melakukan enkripsi dan dekripsi berbeda. Pada algoritma asimetri kunci terbagi menjadi dua bagian yaitu kunci

umum (*public key*) yang bisa diketahui oleh umum dan kunci rahasia (*private key*) yaitu kunci yang dirahasiakan dan hanya boleh diketahui oleh satu orang saja.

3. Fungsi Hash

   Fungsi hash sering disebut dengan fungsi has satu arah (*one way function*), *message digest, fingerprint*, fungsi kompresi dan *Message Authentication Code* (MAC) yang merupakan suatu fungsi matematika yang mengambil masukan panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap.

Algoritma kriptografi tersebut harus memiliki kekuatan untuk melakukannya (Doni Ariyus, 2008) :

1. Konfusi/pembingungan (*confusion*), dari teks terang sehingga sulit untuk direkonstruksikan secara langsung tanpa menggunakan algoritma dekripsinya
2. Difusi/peleburan (*difusion*), dari teks terang sehingga karakteristik dari teks terang tersebut hilang sehingga dapat digunakan untuk mengamankan informasi.

2.2.5 Algoritma Kriptografi Klasik

Sebelum komputer ada, kriptografi dilakukan dengan menggunakan pensil dan kertas. Algoritma kriptografi (*cipher*) yang digunakan saat itu, dinamakan juga algoritma klasik, adalah berbasis karakter, yaitu enkripsi dan dekripsi dilakukan pada setiap karakter pesan. Semua algoritma klasik termasuk ke dalam sistem kriptografi simetris dan digunakan jauh sebelum kriptografi kunci publik ditemukan.

Kriptogarfi klasik memiliki beberapa ciri (Ariyus Dony, 2008) :

1. Berbasis karakter
2. Menggunakan pena dan kertas saja, belum ada computer
3. Termasuk ke dalam kriptografi kunci simetris.

Tiga alasan mempelajari algoritma klasik :

1. Memahami konsep dasar kriptografi
2. Dasar algoritma kriptografi modern
3. Memahami kelemahan sistem kode.

Pada dasarnya, algoritma kriptografi klasik dapat dikelompokkan ke dalam dua macam *cipher*, yaitu (Dony Ariyus, 2008) :

1. *Cipher* substitusi (*substitution cipher*)

   Di dalam *cipher* substitusi setiap unit plainteks diganti dengan satu unit *ciphertext*. Satu "unit" di isini berarti satu huruf, pasangan huruf atau dikelompokkan lebih dari dua huruf. Algoritma substitusi tertua yang diketahui adalah *Caesar cipher* yang digunakan oleh kaisar Romawi, Julius Caesar (sehingga dinamakan juga *casear cipher*), untuk mengirimakan pesan yang dikirimkan kepada gubernurnya.

2. *Cipher* transposisi (*transposition cipher*)

   Pada *cipher* transposisi, huruf-huruf di dalam plainteks tetap saja, hanya saja urutannya diubah. Dengan kata lain algoritma ini melakukan *transpose* terhadap rangkaian karakter di dalam teks. Nama lain untuk metode ini adalah

permutasi atau pengacakan (*scrambling*) karena transpose setiap karakter di dalam teks sama dengan mempermutasikan karakter-karkater tersebut.

a. Jenis-jenis *Cipher* Subtitusi

Beberapa jenis *cipher* yang termasuk dalam jenis *cipher* subtitusi adalah (Rinaldi Munir, 2006) :

1. *Cipher* Alfabet-Tunggal (*monoalphabetic cipher*)

Pada *cipher* alphabet-tunggal atau disebut juga *cipher* subtitusi sederhana (*simple substitution cipher*), satu huruf di plainteks diganti dengan tepat satu huruf *cipher*teks. Jadi fungsi *ciphering*-nya adalah fungsi satu-satu.

Caesar *cipher* adalah kasus khusus dari *cipher* alphabet tunggal dimana susunan huruf *cipher*teks diperoleh dengan menggeser huruf-huruf alphabet sejauh 3 (tiga) karakter. Dalam hal ini kuncinya adalah jumlah pergeseran huruf yaitu 3 (tiga). Susunan alphabet setelah digeser sejauh 3 (tiga) huruf membentuk sebuah tabel subtitusi sebagai berikut :

Tabel 2.2 *Tabel Subtitusi Caesar Cipher*

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

sumber : Rinaldi Munir, 2006

Demikian halnya juga dengan ROT13 (rotare by 13 places) adalah program Caesar *cipher* sederhana yang ditemukan pada system UNIX. ROT13 menggunakan Caesar *cipher* dengan pergeseran k=13 (jadi, huruf A diganti dengan N, B diganti dengan O, dan seterusnya).

Tabel 2.3 Tabel *cipherteks* dalam ROT 13

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

sumber : Rinaldi Munir, 2006

2. *Cipher* Alfabet-Majemuk (*polyalphabetic cipher*)

   Cipher alphabet-majemuk merupakan *cipher* subtitusi ganda (*multiple-subtitutioncipher*) yang melibatkan penggunaan kunci berbeda. *Cipher* alphabet-majemuk dibuat dari sejumlah *cipher* alfabet-tunggal, masing-masing dengan kunci yang berbeda.

3. *Cipher* Subtitusi Homofonik (*homophonic substitution cipher*)

   Cipher subtitusi homofonik adalah seperti *cipher* subtitusi tunggal, kecuali bahwa setiap huruf di dalam plainteks dapat dipetakan ke dalam salah satu dari unit *cipher*teks yang mungkin. Maksudnya, setiap huruf plainteks dapat memiliki lebih dari satu kemungkinan unit *cipher*teks.

4. *Cipher* Subtitusi Poligram (*polygram substitution cipher*)

   Pada *cipher* ini, setiap kelompok huruf disubtitusi dengan kelompok huruf *cipher*teks (jadi tidak mensubtitusi setiap huruf seperti pada *cipher* subtitusi sebelumnya).

b. One-Time Pad (OTP)

One Time Pad cipher adalah salah satu algoritma kriptografi yang tidak terpecahkan. One Time Pad (OTP) ditemukan pada tahu 1917 oleh Major Joseph

Mauborgone. *Cipher* ini termasuk algoritma kriptografi kunci simetri (Rinaldi Munir, 2006).

One Time Pad merupakan algoritma kriptografi yang memiliki kunci berupa deretan-deretan karakter yang dibangkitkan secara acak. Kunci pada OTP hanya digunakan sekali saja untuk mengenkripsi pesan yang kemudian dipakai lagi untuk mendekripsi pesan itu. Setelah selesai maka kunci tersebut dihancurkan.

Aturan enkripsi OTP sama seperti pada *cipher* substitusi abjad majemuk, yaitu untuk proses enkripsi (Rinaldi Munir, 2006) :

$$c_i = (p_i + k_i) \bmod 26$$

sedangkan untuk dekripsi :

$$p_i = (c_i - k_i) \bmod 26$$

sistem *one time pad* tidak dapat dipecahkan karena :

1. Barisan kunci acak yang ditambahkan ke pesan plainteks yang tidak acak menghasilkan *cipher*teks yang seluruhnya acak.
2. Beberapa barisan kunci yang digunakan mendekripsi *cipher*teks mungkin menghasilkan pesan-pesan plainteks yang mempunyai makna, sehingga kriptanalis tidak punya cara untuk menentukan plainteks mana yang benar.

2.3 Algoritma *Merkle Hellman*

Algoritma kriptografi *Merkle Hellman* atau umumnya 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 kriptogarfi pembangkitan kunci-kunci tersebut secara otomatis dengan teknik berantai.

Algoritma ini memiliki aturan substitusi berdasar pada caesar *cipher* yaitu dengan pergeseran huruf-huruf. Kekuatan *cipher* ini terletak pada kunci yaitu nilai integer yang menunjukkan pergeseran karakter-karakter sesuai dengan operasi pada caesar *cipher*. Kekuatan kedua terletak pada barisan bilangan-bilangan yang berfungsi sebagai pengali dengan kunci. Barisan bilangan tersebut dapat berupa bilangan tertentu seperti deret bilangan ganjil, deret bilangan genap, deret *fibonancci*, deret bilangan prima, serta deret bilangan yang dapat dibuat sendiri (Mohamad Firda Fauzan.Studi dan Implementasi Chiper Subtitusi. 12 Februari 2012.http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/20062007/Makalah1/Makalah1-088.pdf).

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

Untuk itu maka standar untuk *cipher* ini adalah *cipher* ganda yaitu *cipher* yang melakukan enkripsi ganda, yaitu dengan membuat pola enkripsi pertama dengan mengerucut ke arah 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 alfabet ASCII.

Matriks dilambangkan dengan M_{ij},dengan $1 \leq i \leq N$ dan $1 \leq j \leq N$, nilai integer kunci dengan K, faktor pengali merupakan tabel integer R. Plainteks dengan P dimana P merupakan tabel plainteks dengan panjang N yaitu P[N].

Berikut operasi matriks untuk proses enkripsi (Mohamad Firda Fauzan.Studi dan Implementasi Chiper Subtitusi. 12 Februari 2012. http://informatika.stei.itb.ac.id/ ~ rinaldi.munir/Kriptografi/2006-2007/Makalah1 / Makalah1-088.pdf) :

1. Matriks enkripsi pertama

   Untuk baris ke-1 :

   $M_{1j} = P[j] + (K * R[1]) \bmod 256$

   untuk baris ke-2 dan selanjutnya untuk nilai $j \geq i$:

   $M_{ij} = M_{(i-1)j} + (K * R [i]) \bmod 256$

   sehingga nilai *cipher*teks yang diperoleh adalah :

   M_{ij}pada nilai $j = (N+i)-N$.

2. Matriks enkripsi kedua

   Nilai P diperoleh dari nilai M_{ij} pada i = j

   Untuk baris ke-1 :

   $M_{1j} = P[j] + (K*R[1])) \bmod 256$

   untuk baris ke 2 dan selanjutnya untuk nilai $j \leq (N+1) - i$:

   $M_{ij} = M_{(i-1)j}+(K*R[i])) \bmod 256$

   sehingga nilai *cipher*teks yang diperoleh adalah :

   M_{ij}pada nilai $j = (N+1)-i$.

Keterangan :

P = Plainteks

N = Jumlah karakter *plain*teks

M = Matriks penampung hasil penyandian

K = Kunci

R = *Row* (baris perkalian faktor pengali dengan kunci)

i = Indeks faktor pengali

j = Indek karakter p*laint*eks

Sedangkan untuk proses dekripsi merupakan kebalikan dari proses enkripsi. Berikut operasi matriks untuk proses dekripsi Mohamad Firda Fauzan. Studi dan Implementasi Chiper Subtitusi 12 Februari 2012.

1. Matriks dekripsi pertama operasinya merupakan kebalikan dari matriks enkripsi, jadi operasi ini kebalikan operasi matriks enkripsi kedua. Nilai C merupakan tabel dari *cipher*teks dengan panjang N yaitu C[N].

   Untuk baris ke-1, berlaku formula :

   $j \leq (N + 1) - i$

   $M_{1j} = C [j] - (K * R[1])) \bmod 256$

   sedangkan untuk baris kedua dan selanjutnya dimana nilai $j \geq i$, berlaku formula : $M_{ij} = (M_{(i-1)\,j} - K * (R[i])) \bmod 256$.

   sehingga nilai *plain*teks yang diperoleh adalah :

   M_{ij}pada nilai j = (N+i)-i.

2. Matriks dekripsi kedua

   Untuk baris pertama berlaku formula :

$M_{1j} = C [j] –(K * R[1])) \bmod 256$

sedangkan untuk baris kedua dan seterusnya nilai $j \geq i$, berlaku formula :

$M_{ij} = C_{[i-1]j}–(K * R[i])) \bmod 256.$

nilai *plain*teks yang diperoleh adalah :

M_{ij}pada nilai j = (N+1)-i.

sehingga nilai *plain*teks yang diperoleh adalah :

M_{ij}pada nilai j = (N+i)-N.

Keterangan :

C = *Cipherteks*

N = Jumlah karakter *cipher*teks

M = Matriks penampung hasil *cipher* yang dijadikan sebagai *plaintext*

K = Kunci

R = *Row* (baris perkalian faktor pengali dengan kunci)

i = Indeks faktor pengali

j = Indek karakter cipherteks

2.4 ASCII (*American Standard Code for Information Interchange)*

ASCII (*American Standard Code for Information Interchange*) merupakan suatu standar internasional dalam kode huruf dan simbol seperti hexa, desimal dan notasi biner. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 8 bit. Dimulai dari 0000 0000 hingga 1111 1111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam sistem bilangan

desimal. Adapun kombinasi kode ASCII yang dikenali komputer dapat dilihat pada tabel ASCII yang menjadi lampiran dalam skripsi ini.

2.5 *Database*

Pada sistem informasi, *database* merupakan sebuah komponen yang paling berperan penting dalam menampung semua data-data yang telah diolah dengan tujuan untuk menghasilkan informasi-informasi yang dibutuhkan oleh para pengguna. Dengan adanya *database*, maka pemutakhiran informasi yang disajikan dapat dilakukan.

2.5.1 Pengertian *Database*

Basis adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi (Abdul Kadir dan Terra CH. Triwahyuni, 2003). Berikut ini adalah beberapa alasan mengapa *database* sangat diperlukan :

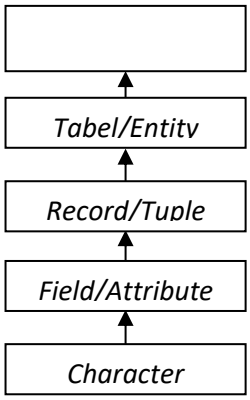
1. Salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
2. Menentukan kualitas informasi yang akurat, tepat waktu dan relevan. Informasi dapat dikatakan bernilai bila manfaatnya lebih aktif dibandingkan biaya mendapatkannya.
3. Mengurangi duplikasi data.
4. Hubungan data dapat ditingkatkan.
5. Mengurangi pemborosan tempat simpanan luar.

2.5.2 Komponen Penyusun *Database* (Basisdata)

Adapun komponen dasar yang menyusun *database* yang umum pada saat ini adalah sebagai beirkut (Janner Simarmata dan Iman Paryudi, 2005) :

1. Skema basisdata
2. Objek-objek skema
3. Tabel
4. *Field* dan kolom
5. *Record* dan baris
6. Kunci
7. Relasi
8. Tipe data

Suatu *database* terdiri dari *file*, *record*, *field* dan *character*, berikut ini adalah jenjang dari suatu *database* (Janner Simarmata dan Iman Paryudi, 2005)terlihat pada gambar dibawah ini :



Gambar 2.1 Jenjang Data dari *Database*
sumber : Janner Simarmata dan Iman Paryudi, 2005

Pengertian dari jenjang data *database* tersebut untuk setiap bagiannya adalah sebagai berikut :

1. *Character*

   Merupakan bagian terkecil dapat berupa karakter *numeric*, huruf ataupun karakter-karakter khusus (*special characters*) yang membentuk suatu item data*/field.*

2. *Field*

   Mempresentasikan suatu atribut dari *record* yang menunjukkan suatu item data/*field*, misalnya nama, alamat dan lain sebagainya. Kumpulan dari *field* membentuk suatu *record*.

3. *Record*

Kumpulan dari *field* membentuk suatu *record. Record* menggambarkan suatu unit data individu yang tertentu. Misalnya *file* personalia, tiap-tiap *record* dapat mewakili data tiap-tiap karyawan.

4. *File*

   File terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Misalnya *file* mata pelajaran berisi data semua tentang mata pelajaran yang ada.

5. *Database*

   Kumpulan dari *file/table* yang membentuk suatu *database*

2.5.3 Keamanan *Database*

Ada beberapa cara yang umum dapat dilakukan untuk mengamankan *database* dari pengaksesan orang lain yang tidak berhak, diantaranya (Janner Simarmata dan Iman Paryudi, 2005) :

1. Kriptografi, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita.
2. *Password* adalah kumpulan karakter atau *string* yang digunakan oleh pengguna jaringan/sebuah sistem operasi yang mendukung banyak pengguna (*multiuser*) untuk memverifikasi identitas dirinya kepada sistem keamanan yang dimiliki oleh jaringan/sistem tersebut.
3. Enkripsi adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus.
4. MD5 (*Message-Digest algortihm 5*) adalah sebuah fungsi hash kriptografik yang digunakan secara luas dengan *hash value* 128-bit.

2.5.4 Tindakan Kemanan *Database*

Keamanan *database* merupakan suatu proteksi terhadap pengrusakan data dan pemakaian data oleh pemakai yang tidak punya kewenangan. Ada beberapa tindakan yang dapat dilakukan untuk mengamankan *database*, diantaranya (Janner Simarmata dan Iman Paryudi, 2005) :

1. Tindakan untuk melindungi sumber daya basis data dari pengaksesan orang-orang yang tidak berhak, modifikasi, atau bentuk intervensi lainnya.
2. Sekumpulan perangkat lunak yang dirancang untuk melindungi *record-record* data dan sumber daya basis data lainnya dari orang-orang yang tidak berhak.

Selanjutnya disebutkan bahwa ada beberapa hal yang menjadi ancaman terhadap kemanana database (Janner Simarmata dan Iman Paryudi, 2005) adalah sebagai berikut :

1. *Interuption*

   Merupakan sumber daya basis data dirusak atau menjadi tidak dapat dipakai (ancaman terhadap *availability*).

2. *Interception*

   Merupakan pemakai atau bagian yang tidak berhak mengakses sumber daya basis data (ancaman *secrecy*).

3. *Modification*

   Merupakan pemakai atau bagian yang tidak berhak tidak hanya mengakses tapi juga merusak sumber daya sistem komputer (ancaman *integrity*).

4. *Fabrication*

   Merupakan pemakai atau bagian yang tidak berhak menyisipkan objek palsu kedalam sistem (ancaman *integrity*).

2.6 Pengertian File Teks

File teks merupakan file yang berisi informasi-informasi dalam bentuk teks. Data yang berasal dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca. Masukan dan keluaran data teks direpresentasikan sebagai set karakter atau sistem kode yang dikenal oleh system komputer. Ada tiga macam set karakter yang umum

digunakan untuk masukan dan keluaran pada komputer, yaitu ASCII, EBCDIC, dan Unicode. ASCII(*American Codefor Information Interchange*) merupakan suatu standar internasional dalam kode huruf dan symbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal. ASCII digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII 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. EBCDIC (*Extended Binary Code Decimal Interchange Code*) merupakan set karakter yang diciptakan oleh computer merk IBM. EBCDIC terdiri dari 256 karakter yang masing-masing berukuran 8 bit. Adanya keterbatasan pada kode ASCII dan EBCDIC, dibuat standar kode internasional baru yang merupakan kode 16 bit yang disebut Unicode. Unicode adalah suatu standar industry yang dirancang untuk mengizinkan teks dan simbol dari semua system tulisan didunia untuk ditampilkan dan dimanipulasi secara konsisten oleh computer (Sudewa,IdaBagusAdi,2003).

2.6.1 Format Teks

Secara umum, format data teks dibagi menjadi dua bagian, yaitu (Purnomo, Herry*etal*, 2005, hal: 410):

a. Teks sederhana (*plaintext*)

   Format data teks(*.txt) merupakan contoh format teks jenis ini yang paling populer.

b. Teks terformat (*formatted text*)

Merupakan teks yang terformat dan mengandung styles. Format data dokumen Microsoft Word (*.doc) merupakan contoh format teks jenis ini yang paling populer.

Contoh format data teks diatas beserta perangkat lunak yang biasa digunakan diantaranya adalah (Purnomo, Herry*etal*, 2005, hal : 410):

1. Format data teks(*.txt)

Format data teks merupakan format teks yang digunakan untuk menyimpan huruf, angka, karakter kontrol (tabulasi, pindah baris, dan sebagainya) atau simbol-simbol lain yang biasa digunakan dalam tulisan seperti titik, koma, tanda petik, dan sebagainya. Satu huruf, angka, karakter kontrol atau symbol pada arsip teks memakan tempat satu byte. Berbeda dengan jenis teks terformat yang satu huruf saja dapat memakan tempat beberapa byte untuk menyimpan format dari huruf tersebut seperti font, ukuran, tebal atau tidak dan sebagainya. Kelebihan dari format data teks ini adalah ukuran datanya yang kecil karena tiadanya fitur untuk memformat tampilan teks. Saat ini perangkat lunak yang paling banyak digunakan untuk memanipulasi format data ini adalah Notepad.

2. Format data dokumen (*.doc)

Doc merupakan ekstensi arsip dokumen perangkat lunak Microsoft Word yang paling banyak digunakan dalam menulis laporan, makalah dan sebagainya. Doc merupakan jenis teks terformat yang tidak hanya dapat

mengatur tampilan teks seperti styles (font, ukuran huruf, dan sebagainya), namun juga dapat menyisipkan gambar. Kekurangan format teks dokumen ini terletak pada ukuran datanya yang besar.

3. *Hyper Text Markup Language* (*.htmatau*.html)

   Merupakan format teks standard untuk tampilan dokumen web.

4. *RichTextFormat*(*.rtf)

   Format teks ini dikembangkan oleh Microsoft yang dapat dibaca oleh berbagai macam *platform*, seperti Windows, Linux, Mac OS dan sebagainya.

2.7 *Microsoft Visual Basic*

Microsoft Visual Basic adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *.NETFramework*, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para *programmer* dapat membangun aplikasi *WindowsForms*, Aplikasi *web* berbasis *ASP.NET*, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++, Visual C#, atau Visual J#),* atau juga dapat diperoleh secara terpadu dalam *Microsoft Visual Studio 2008*.

Visual Basic juga salah satu *development tools* untuk membangun aplikasi dalam lingkungan *Windows. Visual Basic* menggunakan pendekatan Visual untuk merancang *user interface* dalam bentuk *form*, sedangkan untuk lodingnya menggunakan dialek bahasa *Basic* yang cenderung mudah dipelajari. Pada pemprograman Visual, pengembangan aplikasi dimulai dengan pembentukan *user*

interface, kemudian mengatur properti dari objek-objek yang digunakan dalam *user interface*, dan baru dilakukan penulisan kode program untuk menangani kejadian-kejadian. Tahap pengembangan aplikasi demikian dikenal dengan istilah pengembangan aplikasi dengan pendekatan *Bottom Up*.

Bahasa *Visual Basic* sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *MicrosoftVisualBasic* versi sebelumnya yang diimplementasikan di atas *.NETFramework*. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh *Microsoft*, dan versi baru ini tidak kompatibel dengan versi terdahulu. Versi ini merupakan versi terbaru yang dirilis oleh *Microsoft* pada tanggal 19 November 2007, bersamaan dengan dirilisnya *MicrosoftVisualC#* 2008, *MicrosoftVisualC++* 2008, dan Microsoft *.NETFramework* 3.5.

Keunggulan dari *Visual Studio* adalah pada interface desainnya yang menggunakan IDE (*Integreted Development Environment*). Jika diterjemahkan, maka artinya adalah lingkungan pengembangan yang terintegrasi. IDE *Visual Studio* telah terintegrasi dengan *compiler*, *.Net*, fasilitas *debug*, fasilitas pendistribusian file program yang telah jadi, fasilitas *interlisense (autocomplete modern)*, dan lain sebagainya.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Metode Penelitian

Untuk memyelesaikan penelitian ini, dibutuhkan langkah-langkah penyelesaian sebagai berikut :

1. Studi Literatur

   Tahap ini merupakan tahap pembelajaran konsep tentang sistem, konsep pemograman. Konsep ini didapat baik dari buku-buku referensi, paper maupun beberapa artikel di internet. Pada tahap ini juga akan dipelajari metode dan algoritma yang akan digunakan dalam pembuatan perangkat lunak sehingga membantu pada tahap perancangan dan pembuatan perangkat lunak.

2. Perancangan Sistem

   Perancangan sistem ini dimulai dari perancangan alir progam/*flowchart*, perancangan database, dan perancangan antar muka/*interface*

3. Pembuatan Perangkat Lunak

   Dari hasil perancangan dilakukan realisasi pembuatan perangkat lunak, yaitu dengan menerapkan hasil perancangan kedalam bahasa pemograman.

4. Implemetasi Sistem

   Implemetansi merupakan tahap pembuatan kode-kode program sehingga menjadi sebuah aplikasi.

5. Pengujian sistem

   Untuk sistem yang telah dirancang, akan dicoba dan kinerjanya sampai dianggap memadai dan dapat berjalan dengan baik.

3.2 Analisis

3.2.1 Enkripsi Algoritma Merkle Hellman

Sebelum proses perancangan dimulai, maka diperlukanlah beberapa analisis terhadap sistem, metode ataupun teknik-teknik yang digunakan dalam tahap perancangan. Analisa dapat memberi uraian secara utuh tentang masalah yang sedang di analisa dengan melakukan identifikasi dan evaluasi terutama hambatan-hambatan yang terjadi serta kebutuhkan dalam memberi solusi penyelesaian masalah yang sedang dibahas. Berikut ini akan ditampilkan tabel ascii yang digunakan dalam proses penyandian data.

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

Tabel 3.1 Tabel ASCII

Adapun tahap-tahap yang dilakukan oleh penulis dalam melakukan analisa terhadap struktur, dapat diuraikan seperti berikut :

1. Proses enkripsi dilakukan dengan dua tahap yaitu enkripsi pertama dan enkripsi ke dua, sehingga dihasilkan *chiper* akhir yang nantinya menjadi *data*. Penyelesaian tahap enkripsi di atas dapat diuraikan melalui contoh kasus penyandian sebuah di bawah ini :

   a. Matriks enkripsi pertama

   Plainteks adalah SARONI

   Kunci adalah 3 (bilangan integer asli)

Faktor pengali dengan kunci adalah deret bilangan asli (1, 2, 3,…, n).

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

S	A	R	O	N	I
83	65	82	79	78	73

Langkah ke dua adalah membentuk tabel faktor pengali :

Seperti pada kasus di atas, maka faktor pengali yang diguanakan adalah deretan bilangan asli. Jumlah deret bilangan akan disesuaikan dengan jumlah banyaknya karakter dari plainteks.

Jadi, jumlah karakter plainteks (N) adalah 6. Deret bilangan asli (R) yang menjadi faktor pengali adalah 1, 2, 3, 4, 5, 6.

Langkah ke tiga adalah melakukan proses enkripsi pertama sesuai dengan formulanya.

Plainteks (P) = SARONI

N = 6

K = 3

R = 1,2,3,4,5,6

Untuk baris pertama (i = 1), maka :

M_{11} = (P[1] + 3 * R[1]) mod 256

= (T + 3 * (1)) mod 256

= (83 + 3) Mod 256

= 86 (huruf “V” dalam karakter ASCII 256)

M_{12} = (P[2] + 3 * R[1]) mod 256

= (A + 3 * (1)) Mod 256

= (65 + 3) Mod 256

= 68 (huruf "D" dalam karakter ASCII 256)

M_{13} = (P[3] + 3 * R[1]) mod 256

= (R + 3 * (1)) Mod 256

= (82 + 3) Mod 256

= 85 (huruf "U" dalam karakter ASCII 256)

M_{14} = (P[4] + 3 * R[1]) mod 256

= (O + 3 * (1)) mod 256

= (79 + 3) Mod 256

= 82 (huruf "R" dalam karakter ASCII 256)

M_{15} = (P[5] + 3 * (1)) mod 256

= (N + 3 * (1)) mod 256

= (78 + 3) Mod 256

= 81 (huruf "Q" dalam karakter ASCII 256)

M_{16} = (P[6] + 3 * (1)) mod 256

= (I + 3 * (1)) mod 256

= (73 + 3) Mod 256

= 76 (huruf "L" dalam karakter ASCII 256)

hasil sandi pada tahap i = 1 (baris pertama) adalah WDURQL.

Sampai pada tahap ini hasil penyandian dapat ditunjukkan di bawah ini :

SARONI (nilai desimal dalam ASCII : 83 65 82 79 78 73) → i = 0

VDURQL (nilai desimal dalam ASCII : **86** 68 85 82 81 76) → i = 1

Hasil penyandian baris pertama (i = 1) akan digunakan sebagai plainteks pada proses enkripsi baris ke dua (i = 2), dimana nilai j ≥ i, sehingga :

i = 2, j = 2

M_{22} = ($M_{(2-1)2}$ + 3 * (2)) mod 256

= ($M_{(1)2}$ + 3 * (2)) mod 256

= (D + 6) mod 256

= (68 + 6) mod 256

= 74 (huruf “J ” dalam karakter ASCII 256)

M_{23} = ($M_{(2-1)3}$ + 3 * (2)) mod 256

= ($M_{(1)3}$ + 3 * (2)) mod 256

= (85 + 6) mod 256

= 91 (huruf “[” dalam karakter ASCII 256)

M_{24} = ($M_{(2-1)4}$ + 3 * (2)) Mod 256

= ($M_{(1)4}$ + 3 * (2)) Mod 256

= (82 + 6) mod 256

= 88 (huruf “X” dalam karakter ASCII 256)

M_{25} = ($M_{(2-1)5}$ + 3 * (2)) Mod 256

= ($M_{(1)5}$ + 3 * (2)) Mod 256

= (81 + 6) mod 256

= 87 (huruf “W” dalam karakter ASCII 256)

M_{26} = ($M_{(2-1)6}$ + 3 * (2)) Mod 256

= ($M_{(1)6}$ + 3 * (2)) Mod 256

= (76 + 6) mod 256

= 82 (huruf "R" dalam karakter ASCII 256)

hasil dari enkripsi baris ke dua ini adalah J[XWR.

Hasil enkripsi sampai pada tahap ini (i = 2) dapat dilihat di bawah di bawah ini :

SARONI (dalam nilai ASCII 84 65 82 79 78 73) → i = 0

WDURQL (dalam nilai ASCII **87** 68 85 82 81 76) → i = 1

J [XWR (dalam nilai ASCII **74** 91 88 87 82) → i = 2

Hasil enkripsi pada baris ke dua (i=2) akan digunakan sebagai plainteks pada proses enkripsi baris ke tiga (i=3), sehingga :

i = 3, j = 3

M_{33} = ($M_{(3-1)3}$ + 3 * (3)) Mod 256

= ($M_{(2)3}$ + 3 * (3)) Mod 256

= (91 + 9) mod 256

= 100 (huruf "d" dalam karakter ASCII 256)

M_{34} = ($M_{(3-1)4}$ + 3 * (3)) Mod 256

= ($M_{(2)4}$ + 3 * (3)) Mod 256

= (88 + 9) mod 256

= 97 (huruf "a" dalam karakter ASCII 256)

M_{35} = ($M_{(3-1)5}$ + 3 * (3)) Mod 256

= ($M_{(2)5}$ + 3 * (3)) Mod 256

= (87 + 9) mod 256

= 96 (huruf " ` " dalam karakter ASCII 256)

M_{36} = ($M_{(3-1)6}$ + 3 * (3)) Mod 256

= ($M_{(2)6}$ + 3 * (3)) Mod 256

= (82 + 9) mod 256

= 91 (huruf "[" dalam karakter ASCII 256)

hasil dari enkripsi baris ke tiga (i=3) ini adalah da`[.

Hasil enkripsi sampai pada tahap baris ke tiga (i =3) dapat dilihat di bawah ini:

SARONI (dalam nilai ASCII 84 65 82 79 78 73) → i = 0

WDURQL (dalam nilai ASCII **87** 68 85 82 81 76) → i = 1

J [XWR (dalam nilai ASCII **74** 91 88 87 82) → i = 2

d a ` [(dalam nilai ASCII **100** 97 96 91) → 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

M_{44} = ($M_{(4-1)4}$ + 3 * (4)) Mod 256

= ($M_{(3)4}$ + 3 * (4)) Mod 256

= (97 + 12) mod 256

= 109 (huruf "m" dalam karakter ASCII 256)

M_{45} = ($M_{(4-1)5}$ + 3 * (4)) Mod 256

= ($M_{(3)5}$ + 3 * (4)) Mod 256

= (96 + 12) mod 256

= 108 (huruf "l" dalam karakter ASCII 256)

M_{46} = ($M_{(4-1)6}$ + 3 * (4)) Mod 256

= ($M_{(3)6}$ + 3 * (4)) Mod 256

= (91 + 12) mod 256

= 103 (huruf “g” dalam karakter ASCII 256)

hasil dari enkripsi baris ke tiga ini adalah mlg.

Hasil enkripsi sampai pada tahap baris ke empat (i = 4) dapat dilihat di bawah ini:

SARONI	(dalam nilai ASCII	84	65	82	79	78	73)	→ i = 0
WDURQL	(dalam nilai ASCII	**87**	68	85	82	81	76)	→ i = 1
J [XWR	(dalam nilai ASCII		**74**	91	88	87	82)	→ i = 2
d a ` [	(dalam nilai ASCII			**100**	97	96	91)	→ i = 3
mlg	(dalam nilai ASCII				**109**	108	103)	→ i = 4

Hasil enkripsi pada baris ke empat (i=4) akan digunakan sebagai plainteks pada proses enkripsi baris ke lima (i=5), sehingga :

i = 5, j = 5

M_{55} = ($M_{(5-1)5}$ + 3 * (5)) Mod 256

= ($M_{(4)5}$ + 3 * (5)) Mod 256

= (108 + 15) mod 256

= 123 (huruf “{ ” dalam karakter ASCII 256)

M_{56} = ($M_{(5-1)6}$ + 3 * (5)) Mod 256

= ($M_{(4)6}$ + 3 * (5)) Mod 256

= (103 + 15) mod 256

= 118 (huruf “v” dalam karakter ASCII 256)

hasil dari enkripsi baris ke lima ini adalah {v.

Hasil enkripsi sampai pada tahap baris ke lima (i = 5) dapat dilihat di bawah ini:

SARONI (dalam nilai ASCII 83 65 82 79 78 73) → i = 0

VDURQL (dalam nilai ASCII **86** 68 85 82 81 76) → i = 1

J [XVR (dalam nilai ASCII **74** 91 88 86 82) → i = 2

d a ` [(dalam nilai ASCII **100** 97 87 91) → i = 3

mlg (dalam nilai ASCII **109** 99 103) → i = 4

{v (dalam nilai ASCII **123** 118) → i = 5

Hasil enkripsi pada baris ke lima (i=5) akan digunakan sebagai plainteks pada proses enkripsi baris ke enam (i=6), sehingga :

i = 6, j = 6

M_{66} = ($M_{(6-1)6}$ + 3 * (6)) Mod 256

= ($M_{(5)6}$ + 3 * (6)) Mod 256

= (118 + 18) mod 256

= 136 (huruf " ˆ " dalam karakter ASCII 256)

hasil dari enkripsi baris ke tiga ini adalah ˆ .

Hasil enkripsi sampai pada tahap baris ke enam (i = 6) dapat dilihat di bawah ini:

SARONI (dalam nilai ASCII 83 65 82 79 78 73) → i = 0

VDURQL (dalam nilai ASCII **86** 68 85 82 81 76) → i = 1

J [XVR (dalam nilai ASCII **74** 91 88 86 82) → i = 2

d a ` [(dalam nilai ASCII **100** 97 86 91) → i = 3

mlg (dalam nilai ASCII **109** 99 103) → i = 4

{v (dalam nilai ASCII **114** 118) → i = 5

ˆ (dalam nilai ASCII**136**) → i = 6

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 :

SARONI plainteks

VDURQL hasil sandi pada i = 1 dan j = (6+1) - 6 = 1

J [XVR hasil sandi pada i = 2 dan j = (6+2) - 6 = 2

d a ` [hasil sandi pada i = 3 dan j = (6+3) - 6 = 3

mlg hasil sandi pada i = 4 dan j = (6+4) - 6 = 4

{v hasil sandi pada i = 5 dan j = (6+5) - 6 = 5

ˆ hasil sandi pada i = 6 dan j = (6+6) - 6 = 6

maka yang menjadi *cipher* pada proses enkripsi pertama adalah **VJdm{ˆ** dimanadapatdilihat bahwasusunan dari baris dan kolomnya berbentuk yang mengerucut ke kiri.

a. Matriks enkripsi ke dua

Langkah-langkah yang dilakukan pada proses enkripsi kedua hampir sama dengan proses pada enkripsi pertama. Faktor pengali dan kunci yang digunakan tetap sama. Pada proses ini yang menjadi plainteks adalah *cipher* yang dihasilkan dari proses enkripsi pertama (**VJdm{ˆ**) kemudian dienkrip lagi sesuai dengan formula yang berlaku pada proses enkripsi ke dua.

Plainteks = V J d m { ˆ (cipher hasil enkripsi pertama)

86 74 100 109 123 136 (nilai desimal dalam ASCII)

Untuk baris pertama (i = 1) :

M_{11} = (P[1] + (3 * 1)) mod 256

= (V + (3 * 1)) mod 256

= (86 + 3) mod 256

= 89 (huruf “Y” dalam karakter ASCII 256)

M_{12} = (P[2] + (3 * 1)) mod 256

= (J + (3 * 1)) mod 256

= (74 + 3) mod 256

= 77 (huruf “M” dalam karakter ASCII 256)

M_{13} = (P[3] + (3 * 1)) mod 256

= (d + (3 * 1)) mod 256

= (100 + 3) mod 256

= 103 (huruf “g” dalam karakter ASCII 256)

M_{14} = (P[4] + (3 * 1)) mod 256

= (m + (3 * 1)) mod 256

= (109 + 3) mod 256

= 112 (huruf “p” dalam karakter ASCII 256)

M_{15} = (P[5] + (3 * 1)) mod 256

= ({ + (3 * 1)) mod 256

= (123 + 3) mod 256

= 126 (huruf “~” dalam karakter ASCII 256)

M_{16} = (P[6] + (3 * 1)) mod 256

= (ˆ + (3 * 1)) mod 256

= (136 + 3) mod 256

= 139 (huruf " ‹ " dalam karakter ASCII 256)

hasil dari enkripsi baris pertama (i = 1) adalah ZMgp~‹ .

Hasil enkripsi sampai pada tahap baris pertama (i = 1) dapat dilihat di bawah ini:

V J d m { ˆ (86 74 100 109 123 136 → i = 0

Y M g p ~ ‹ (89 77 103 112 126 **139** → i = 1

Hasil enkripsi baris pertama (i = 1) akan digunakan sebagai plainteks pada proses enkripsi baris ke dua dimana nilai $j \leq (N + 1) - i$, sehingga :

$i = 2;\ j \leq (6 + 1) - 2 \rightarrow j \leq 5$

M_{21} = $(M_{(2-1)1} + (K * R[\ i\ ]))$ mod 256

= $(M_{(1)1} + (K * R[\ i\ ]))$ mod 256

= (Y + (3 * 2)) mod 256

= (90 + 6) mod 256

= 96 (huruf " ` " dalam karakter ASCII 256)

M_{22} = $(M_{(2-1)2} + (K * R[\ i\ ]))$ mod 256

= $(M_{(1)2} + (K * R[\ i\ ]))$ mod 256

= (M + (3 * 2)) mod 256

= (77 + 6) mod 256

= 83 (huruf "S" dalam karakter ASCII 256)

M_{23} = $(M_{(2-1)3} + (K * R[\ i\ ]))$ mod 256

= $(M_{(1)3} + (K * R[\ i\ ]))$ mod 256

= (g + (3 * 2)) mod 256

= (103 + 6) mod 256

= 109 (huruf “m” dalam karakter ASCII 256)

M_{24} = $(M_{(2-1)4} + (K * R[\ i\ ]))$ mod 256

= $(M_{(1)4} + (K * R[\ i\ ]))$ mod 256

= (p + (3 * 2)) mod 256

= (112 + 6) mod 256

= 118 (huruf “v” dalam karakter ASCII 256)

M_{25} = $(M_{(2-1)5} + (K * R[\ i\ ]))$ mod 256

= $(M_{(1)5} + (K * R[\ i\ ]))$ mod 256

= (~ + (3 * 2)) mod 256

= (126 + 6) mod 256

= 132 (huruf “„” dalam karakter ASCII 256)

hasil dari enkripsi baris ke dua (i = 2) adalah `SmV**„** .

Hasil enkripsi sampai pada tahap baris ke dua (i=2) dapat dilihat di bawah ini:

V J d m { ˆ → i = 0

Z M g p ~ ‹ → 90 77 103 112 126 **139** → i = 1

` S m V **„** → 96 83 109 118 **132** → i = 2

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

i = 3 ; j ≤ (6 + 1) – 3 → j ≤ 4

M_{31} = $(M_{(3-1)1} + (K * R[\ i\ ]))$ mod 256

= $(M_{(2)1} + (K * R[\ i\ ]))$ mod 256

= (` + (3 * 3)) mod 256

= (96 + 9) mod 256

= 105 (huruf " i " dalam karakter ASCII 256)

M_{32} = $(M_{(3-1)2}$ + (K * R[i])) mod 256

= $(M_{(2)2}$ + (K * R[i])) mod 256

= (S + (3 * 3)) mod 256

= (83 + 9) mod 256

= 92 (huruf " \ " dalam karakter ASCII 256)

M_{33} = $(M_{(3-1)3}$ + (K * R[i])) mod 256

= $(M_{(2)3}$ + (K * R[i])) mod 256

= (m + (3 * 3)) mod 256

= (109 + 9) mod 256

= 118 (huruf " v " dalam karakter ASCII 256)

M_{34} = $(M_{(3-1)4}$ + (K * R[i])) mod 256

= $(M_{(2)4}$ + (K * R[i])) mod 256

= (V + (3 * 3)) mod 256

= (118 + 9) mod 256

= 127 (huruf "Del " dalam karakter ASCII 256 / delete)

hasil dari enkripsi baris ke tiga (i = 3) adalah i\v Del .

Hasil enkripsi sampai pada tahap baris ke tiga (i = 3) dapat dilihat di bawah ini:

W J d m { ˆ → i = 0

Z M g p ~ ‹ → 90 77 103 112 126 139 → i = 1

` S m V „ → 96 83 109 118 132 → i = 2

i \ v Del → 105 92 118 127 → i = 3

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

$i = 4 ; j \leq (6 + 1) - 4 \rightarrow j \leq 3$

$M_{41} = (M_{(4-1)1} + (K * R[i])) \bmod 256$

$= (M_{(3)1} + (K * R[i])) \bmod 256$

$= (i + (3 * 4)) \bmod 256$

$= (105 + 12) \bmod 256$

= 117 (huruf " u " dalam karakter ASCII 256)

$M_{42} = (M_{(4-1)2} + (K * R[i])) \bmod 256$

$= (M_{(3)2} + (K * R[i])) \bmod 256$

$= (\backslash + (3 * 4)) \bmod 256$

$= (92 + 12) \bmod 256$

= 104 (huruf " h " dalam karakter ASCII 256)

$M_{43} = (M_{(4-1)3} + (K * R[i])) \bmod 256$

$= (M_{(3)3} + (K * R[i])) \bmod 256$

$= (v + (3 * 4)) \bmod 256$

$= (118 + 12) \bmod 256$

= 130 (huruf " , " dalam karakter ASCII 256)

hasil dari enkripsi baris ke empat (i = 4) adalah uh**,** .

Hasil enkripsi sampai pada tahap baris ke empat (i = 4) dapat dilihat di bawah ini:

V J d m { ^ → i = 0

Z M g p ~ ‹ → 90 77 103 112 126 **139** → i = 1

` S m V „ → 96 83 109 118 **132** → i = 2

i \ v Del → 105 92 118 **127** → i = 3

u h , → 117 104 **130** → i = 4

Hasil enkripsi baris ke empat (i=4) akan digunakan sebagai plainteks pada proses enkripsi baris ke lima, sehingga :

i = 5 ; j ≤ (6 + 1) – 5 → j ≤ 2

M_{51} = $(M_{(5-1)1}$ + (K * R[i])) mod 256

= $(M_{(4)1}$ + (K * R[i])) mod 256

= (u + (3 * 5)) mod 256

= (117 + 15) mod 256

= 132 (huruf “ „ ” dalam karakter ASCII 256)

M_{52} = $(M_{(5-1)2}$ + (K * R[i])) mod 256

= $(M_{(4)2}$ + (K * R[i])) mod 256

= (r + (3 * 5)) mod 256

= (104 + 15) mod 256

= 119 (huruf “ w ” dalam karakter ASCII 256)

hasil dari enkripsi baris ke lima (i = 5) adalah „**w** .

Hasil enkripsi sampai pada tahap baris ke lima (i = 5) dapat dilihat di bawah ini:

V J d m { ˆ → i = 0

Z M g p ~ ‹ → 90 77 103 112 126 **139** → i = 1

` S m V „ → 96 83 109 118 **132** → i = 2

i \ v Del → 105 92 118 **127** → i = 3

u h **,** → 117 104 **130** → i = 4

„ **w** → 132 **119** → i = 5

hasil enkripsi baris ke lima (i = 5) akan digunakan sebagai plainteks pada proses enkripsi baris ke enam, sehingga :

i = 6 ; j ≤ (6 + 1) – 6 → j ≤ 1

M_{61} = $(M_{(6-1)1} + (K * R[\ i\ ]))$ mod 256

= $(M_{(5)1} + (K * R[\ i\ ]))$ mod 256

= („ + (3 * 6)) mod 256

= (132 + 18) mod 256

= 150 (huruf " – " dalam karakter ASCII 256)

hasil dari enkripsi baris ke enam (i = 6) adalah – .

Hasil enkripsi sampai pada tahap baris ke enam (i = 6) dapat dilihat di bawah ini:

V J d m { ˆ → i = 0

Z M g p ~ ‹ → 90 77 103 112 126 **139** → i = 1

` S m V **„** → 96 83 109 118 **132** → i = 2

i \ v Del → 105 92 118 **127** → i = 3

u h **,** → 117 104 **130** → i = 4

„ **w** → 132 **119** → i = 5

– →**150** → i = 6

karakter yang menjadi hasil enkrip kedua adalah huruf terakhir dari masing-masing baris dan diperoleh berdasarkan formula M_{ij} pada nilai j = (N+1) – i, sehingga :

Z M g p ~ ‹ → i = 1 dan j = (6 + 1) – 1 = 6 → ‹

` S m V „ → i = 2 dan j = (6 + 1) – 2 = 5 → ,,

i \ v Del → i = 3 dan j = (6 + 1) – 3 = 4 →Del

u r , → i = 4 dan j = (6 + 1) – 4 = 3 → ,

,, **w** → i = 5 dan j = (6 + 1) – 5 = 2 → w

– → i = 6 dan j = (6 + 1) – 6 = 1 → –

Dapat dilihat bahwa hasil penyandian pada proses enkripsi kedua membentuk yang mengerucut ke kanan dan menghasilkan cipherteks akhir adalah **–w,□„‹** . *Cipher* terakhir inilah yang nantinya disimpan menjadi di dalam *data* .

3.2.2 Dekripsi Algoritma Merkle Hellman

Proses dekripsi merupakan kebalikan dari proses enkripsi yang telah dilakukan sebelumnya. Kunci dan faktor pengali yang digunakan tetap sama seperti pada proses enkripsi. Proses pengembalian data tersandi ke data asli dilakukan sebanyak dua kali, terdiri dari dekripsi pertama dan dekripsi kedua. Penyelesaian tahap dekripsi di atas dapat diuraikan melalui contoh kasus di bawah ini dimana data sandi adalah hasil akhir dari penyandian contoh enkripsi :

1. Dekripsi pertama

   chipertext adalah – **w** , Del „ ‹ pada i = 0

   150 119 130 127 132 139 (nilai desimal dalam ASCII)

i = 1 $j \leq (6 + 1) - 1$

$j \leq 6$

M_{11} = C [1] – (3 * [1])) mod 256

= (– - (3 * 1)) mod 256

= (150 – 3) mod 256

= 147 (huruf “ “ ” dalam karakter ASCII 256)

M_{12} = C [2] – (3 * [1])) mod 256

= (w – (3 * 1)) mod 256

= (119 – 3) mod 256

= 116 (huruf “ t ” dalam karakter ASCII 256)

M_{13} = C [3] – (3 * [1])) mod 256

= (, – (3 * 1)) mod 256

= (130 – 3) mod 256

= 127 (huruf “Del ” dalam karakter ASCII 256)

M_{14} = C [4] – (3 * [1])) mod 256

= (Del – (3 * 1)) mod 256

= (127 – 3) mod 256

= 124 (huruf “ | ” dalam karakter ASCII 256)

M_{15} = C [5] – (3 * [1])) mod 256

= (,, – (3 * 1)) mod 256

= (132 – 3) mod 256

= 129 (huruf “¥” dalam karakter ASCII 256)

M_{16} = C [6] – (3 * [1])) mod 256

= (‹ – (3 * 1)) mod 256

= (139 – 3) mod 256

= 136 (huruf “ ˆ ” dalam karakter ASCII 256)

hasil dari dekripsi baris pertama (i = 1) adalah “ t Del | Del ˆ .

Hasil dekripsi sampai pada tahap baris pertama (i = 1) dapat dilihat di bawah ini :

– w , Del **„ ‹** (150 119 130 127 132 139) → i = 0

“ t Del | ¥ ˆ (147 116 127 124 129 **136**) → i = 1

Hasil dekripsi baris pertama (i=1) akan digunakan sebagai *chipertext* pada proses dekripsi baris ke dua, sehingga :

i = 2; $j \leq (6 + 1) - 2$

$j \leq 5$;

M_{21} = ($M_{(2-1)\,1}$ – K * (R[2])) mod 256.

= ($M_{(1)1}$ – 3 * (2)) mod 256.

= (“ – 6) mod 256

= (147 – 6) mod 256

= 141 (huruf “Ž ” dalam karakter ASCII 256)

M_{22} = ($M_{(2-1)\,2}$ – K * (R[2])) mod 256.

= ($M_{(1)2}$ – 3 * (2)) mod 256.

= (t – 6) mod 256

= (116 – 6) mod 256

= 110 (huruf “ n ” dalam karakter ASCII 256)

M_{23} = ($M_{(2-1)\,3}$ – K * (R[2])) mod 256.

= ($M_{(1)3}$ – 3 * (2)) mod 256.

= (Del – 6) mod 256

= (127 – 6) mod 256

= 121 (huruf “ y ” dalam karakter ASCII 256)

M_{24} = ($M_{(2-1)4}$ – K * (R[2])) mod 256.

= ($M_{(1)4}$ – 3 * (2)) mod 256.

= (| – 6) mod 256

= (124 – 6) mod 256

= 118 (huruf “ v ” dalam karakter ASCII 256)

M_{25} = ($M_{(2-1)5}$ – K * (R[2])) mod 256.

= ($M_{(1)5}$ – 3 * (2)) mod 256.

= (Del – 6) mod 256

= (129 – 6) mod 256

= 123 (huruf “ { ” dalam karakter ASCII 256)

hasil dari dekripsi baris ke dua (i = 2) adalah Deln y v **{** .

Hasil dekripsi sampai pada tahap baris ke dua (i = 2) dapat dilihat di bawah ini :

– w , Del**„** ‹ (150 119 130 127 132 139) → i = 0

“ t Del | Del^ (147 116 127 124 129 **136**) → i = 1

Del n y v **{** (141 110 121 118 **123**) → i = 2

Hasil dekripsi baris ke dua (i=2) akan digunakan sebagai *chipertext* pada proses dekripsi baris ke tiga, sehingga :

i = 3; j ≤ (6 + 1) – 3

j ≤ 4;

M_{31} = ($M_{(3-1)\,1}$ – K * (R[3])) mod 256.

= ($M_{(2)1}$ – 3 * (3)) mod 256.

= (Ž – 9) mod 256

= (141 – 9) mod 256

= 132 (huruf „ ” dalam karakter ASCII 256)

M_{32} = ($M_{(3-1)\,2}$ – K * (R[3])) mod 256.

= ($M_{(2)2}$ – 3 * (3)) mod 256.

= (n – 9) mod 256

= (110 – 9) mod 256

= 101 (huruf “ e ” dalam karakter ASCII 256)

M_{33} = ($M_{(3-1)\,3}$ – K * (R[3])) mod 256.

= ($M_{(2)3}$ – 3 * (3)) mod 256.

= (y – 9) mod 256

= (121 – 9) mod 256

= 112 (huruf “ p ” dalam karakter ASCII 256)

M_{34} = ($M_{(3-1)\,4}$ – K * (R[3])) mod 256.

= ($M_{(2)4}$ – 3 * (3)) mod 256.

= (v – 9) mod 256

= (118 – 9) mod 256

= 109 (huruf “ m ” dalam karakter ASCII 256)

hasil dari dekripsi baris ke dua (i = 2) adalah „epm.

Hasil dekripsi sampai pada tahap baris ke tiga (i = 3) dapat dilihat di bawah ini :

– **w ,** Del **„ ‹** (150 119 130 127 132 139) → i = 0

“ t Del | ¥ ˆ (147 116 127 124 129 **136**) → i = 1

Ž n y v **{** (141 110 121 118 **123**) → i = 2

„ e p **m** (132 101 112 **109**) → i = 3

Hasil dekripsi baris ke tiga (i = 3) akan digunakan sebagai *chipertext* pada proses dekripsi baris ke empat, sehingga :

i = 4; $j \leq (6 + 1) - 4$

$j \leq 3$;

M_{41} = ($M_{(4-1)\,1}$ – K * (R[4])) mod 256.

= ($M_{(3)1}$ – 3 * (4)) mod 256.

= („ – 12) mod 256

= (132 – 12) mod 256

= 120 (huruf “ x ” dalam karakter ASCII 256)

M_{42} = ($M_{(4-1)\,2}$ – K * (R[4])) mod 256.

= ($M_{(3)2}$ – 3 * (4)) mod 256.

= (e – 12) mod 256

= (101 – 12) mod 256

= 89 (huruf “ Y ” dalam karakter ASCII 256)

M_{43} = ($M_{(4-1)\,3}$ – K * (R[4])) mod 256.

= ($M_{(3)3}$ – 3 * (4)) mod 256.

= (p – 12) mod 256

= (112 – 12) mod 256

= 100 (huruf “ d ” dalam karakter ASCII 256)

hasil dari dekripsi baris ke empat (i = 4) adalah xYd.

Hasil dekripsi sampai pada tahap baris ke empat (i = 4) dapat dilihat di bawah ini :

– **w ,** Del**„ ‹** (150 119 130 127 132 139) → i = 0

“ t Del | ¥ ˆ (147 116 127 124 129 **136**) → i = 1

Ž n y v **{** (141 110 121 118 **123**) → i = 2

„ e p **m** (132 101 112 **109**) → i = 3

x Y **d** (120 89 **100**) → i = 4

Hasil dekripsi baris ke empat (i = 4) akan digunakan sebagai *chipertext* pada proses dekripsi baris ke lima, sehingga :

i = 5; $j \leq (6 + 1) – 5$

$j \leq 2$;

M_{51} = ($M_{(5-1)\,1}$ – K * (R[5])) mod 256.

= ($M_{(4)1}$ – 3 * (5)) mod 256.

= (x – 15) mod 256

= (120 – 15) mod 256

= 105 (huruf “ i ” dalam karakter ASCII 256)

M_{52} = ($M_{(5-1)\,2}$ – K * (R[5])) mod 256.

= ($M_{(4)2}$ – 3 * (5)) mod 256.

= (Y – 15) mod 256

= (89 – 15) mod 256

= 74 (huruf " J " dalam karakter ASCII 256)

hasil dari dekripsi baris ke empat (i = 5) adalah iJ.

Hasil dekripsi sampai pada tahap baris ke lima (i = 5) dapat dilihat di bawah ini :

– **w ,** Del „ ‹ (150 119 130 127 132 139) → i = 0

" t Del | ¥ ˆ (147 116 127 124 129 **136**) → i = 1

Ž n y v **{** (141 110 121 118 **123**) → i = 2

„ e p **m** (132 101 112 **109**) → i = 3

x Y **d** (120 89 **100**) → i = 4

i **J** (105 **74**) → i = 5

Hasil dekripsi baris ke lima (i = 5) akan digunakan sebagai *chipertext* pada proses dekripsi baris ke enam, sehingga :

i = 6; $j \leq (6 + 1) - 6$

$j \leq 1$;

$M_{61} = (M_{(6-1)\,1} - K * (R[6])) \bmod 256.$

$= (M_{(5)1} - 3 * (6)) \bmod 256.$

$= (i - 18) \bmod 256$

$= (105 - 18) \bmod 256$

= 87 (huruf " W " dalam karakter ASCII 256)

hasil dari dekripsi baris ke enam (i = 6) adalah W.

Hasil dekripsi sampai pada tahap baris ke enam (i = 6) dapat dilihat di bawah ini :

" t Del | ¥ ‹ ˆ (147 116 127 124 129 **136**) → i = 1 ; j = 6

Ž n y v **{** (141 110 121 118 **123**) → i = 2 ; j = 5

„ e p **m** (132 101 112 **109**) → i = 3 ; j = 4

x Y **d** (120 89 **100**) → i = 4 ; j = 3

i **J** (105 **74**) → i = 5 ; j = 2

W (**87**) → i = 6 ; j = 1

sehingga pada proses dekripsi pertama diperoleh plainteks sesuai dengan formula M_{ij} pada nilai j = (N+i)-i adalah **WJdm{ˆ**.

2. Dekripsi ke dua

Proses dekripsi ke dua merupakan kebalikan dari hasil proses enkripsi pertama. Chiperteks sumber adalah hasil akhir dari proses dekripsi pertama (**WJdm{ˆ**).

Chipertext = **W J d m { ˆ** (hasil dekrip pertama)

87 74 100 109 123 136 (nilai desimal dalam ASCII)

untuk baris pertama (i = 1) :

M_{11} = C [1] – (3 * R[1])) mod 256

= (W – (3 * 1)) mod 256

= (87 – 3) mod 256

= 84 (huruf " T " dalam karakter ASCII 256)

M_{12} = C [2] – (3 * R[1])) mod 256

= (J – (3 * 1)) mod 256

= (74 – 3) mod 256

= 71 (huruf " G " dalam karakter ASCII 256)

M_{13} = C [3] – (3 * R[1])) mod 256

= (d – (3 * 1)) mod 256

= (100 – 3) mod 256

= 97 (huruf " a " dalam karakter ASCII 256)

M_{14} = C [4] – (3 * R[1])) mod 256

= (m – (3 * 1)) mod 256

= (109 – 3) mod 256

= 106 (huruf " j " dalam karakter ASCII 256)

M_{15} = C [5] – (3 * R[1])) mod 256

= ({ – (3 * 1)) mod 256

= (123 – 3) mod 256

= 120 (huruf " x " dalam karakter ASCII 256)

M_{16} = C [6] – (3 * R[1])) mod 256

= (ˆ – (3 * 1)) mod 256

= (136 – 3) mod 256

= 133 (huruf "… " dalam karakter ASCII 256)

hasil dari dekripsi baris pertama (i = 1) adalah TGa jx ….

Hasil dekripsi sampai pada tahap baris pertama (i = 1) dapat dilihat di bawah ini :

W J d m { ˆ (dalam nilai ASCII 87 74 100 109 123 136) → i = 0

S G a j x … (dalam nilai ASCII **84** 71 97 106 120 133) → i = 1

Hasil dekripsi baris pertama (i = 1) akan digunakan sebagai *chipertext* pada proses dekripsi baris ke dua, sehingga :

i = 2; j ≥ 2;

M_{22} = $(C_{[2-1]2} – (3 * R[\ 2\ ])) \bmod 256$

= $(C_{[1]2} – (3 * 2)) \bmod 256$

= $(G – 6) \bmod 256$

= $(71 – 6) \bmod 256$

= 65 (huruf “ A ” dalam karakter ASCII 256)

M_{23} = $(C_{[2-1]3} – (3 * R[\ 2\ ])) \bmod 256$

= $(C_{[1]3} – (3 * 2)) \bmod 256$

= $(a – 6) \bmod 256$

= $(97 – 6) \bmod 256$

= 91 (huruf “ [” dalam karakter ASCII 256)

M_{24} = $(C_{[2-1]4} – (3 * R[\ 2\ ])) \bmod 256$

= $(C_{[1]4} – (3 * 2)) \bmod 256$

= $(\ j – 6) \bmod 256$

= $(106 – 6) \bmod 256$

= 100 (huruf “ d ” dalam karakter ASCII 256)

M_{25} = $(C_{[2-1]5} – (3 * R[\ 2\ ])) \bmod 256$

= $(C_{[1]5} – (3 * 2)) \bmod 256$

= $(\ x – 6) \bmod 256$

= $(120 – 6) \bmod 256$

= 114 (huruf “ r ” dalam karakter ASCII 256)

M_{26} = $(C_{[2-1]6} – (3 * R[\ 2\ ])) \bmod 256$

= $(C_{[1]6} – (3 * 2)) \bmod 256$

= $(… – 6) \bmod 256$

= (133 – 6) mod 256

= 127 (huruf “Del” dalam karakter ASCII 256)

hasil dari dekripsi baris pertama (i = 1) adalah A [d r Del

Hasil dekripsi sampai pada tahap baris pertama (i = 1) dapat dilihat di bawah ini :

W J d m { ^ (dalam nilai ASCII 87 74 100 109 123 136) → i = 0

S G a j x … (dalam nilai ASCII **83** 71 97 106 120 133) → i = 1

A [d r Del (dalam nilai ASCII **65** 91 100 114 127) → i = 2

Hasil dekripsi baris ke dua (i = 2) akan digunakan sebagai *chipertext* pada proses dekripsi baris ke tiga, sehingga :

i = 3; j ≥ 3;

M_{33} = $(C_{[3-1]3} – (3 * R[\ 3\ ])) \bmod 256$

= $(C_{[2]3} – (3 * 3)) \bmod 256$

= ([– 9) mod 256

= (91 – 9) mod 256

= 82 (huruf “ R ” dalam karakter ASCII 256)

M_{34} = $(C_{[3-1]4} – (3 * R[\ 3\ ])) \bmod 256$

= $(C_{[2]4} – (3 * 3)) \bmod 256$

= (d – 9) mod 256

= (100 – 9) mod 256

= 91 (huruf “ [” dalam karakter ASCII 256)

M_{35} = $(C_{[3-1]5} – (3 * R[\ 3\ ])) \bmod 256$

= $(C_{[2]5} – (3 * 3)) \bmod 256$

= (r – 9) mod 256

= (114 – 9) mod 256

= 105 (huruf " i " dalam karakter ASCII 256)

M_{36} = $(C_{[3-1]6} – (3 * R[3])) \bmod 256$

= $(C_{[2]6} – (3 * 3)) \bmod 256$

= (Del – 9) mod 256

= (127 – 9) mod 256

= 118 (huruf " v " dalam karakter ASCII 256)

hasil dari dekripsi baris ke tiga (i = 3) adalah **R [i v**

Hasil dekripsi sampai pada tahap baris ke tiga (i = 3) dapat dilihat di bawah ini :

W J d m { ˆ (dalam nilai ASCII 87 74 100 109 123 136) → i = 0

SG a j x … (dalam nilai ASCII **83** 71 97 106 120 133) → i = 1

A [d r Del (dalam nilai ASCII **65** 91 100 114 127) → i = 2

R [i v (dalam nilai ASCII **82** 91 105 118) → i = 3

Hasil dekripsi baris ke tiga (i = 3) akan digunakan sebagai *chipertext* pada proses dekripsi baris ke empat, sehingga :

i = 4; j ≥ 4;

M_{44} = $(C_{[4-1]4} – (3 * R[4])) \bmod 256$

= $(C_{[3]4} – (3 * 4)) \bmod 256$

= ([– 12) mod 256

= (91 – 12) mod 256

= 79 (huruf " O " dalam karakter ASCII 256)

M_{45} = $(C_{[4-1]5} – (3 * R[\ 4\ ]))$ mod 256

= $(C_{[3]5} – (3 * 4))$ mod 256

= (i – 12) mod 256

= (105 – 12) mod 256

= 93 (huruf "] " dalam karakter ASCII 256)

M_{46} = $(C_{[4-1]6} – (3 * R[\ 4\ ]))$ mod 256

= $(C_{[3]6} – (3 * 4))$ mod 256

= (v – 12) mod 256

= (118 – 12) mod 256

= 106 (huruf " j " dalam karakter ASCII 256)

hasil dari dekripsi baris ke empat (i = 4) adalah **O**] j

Hasil dekripsi sampai pada tahap baris ke empat (i = 4) dapat dilihat di bawah ini :

W J d m { ^ (dalam nilai ASCII 87 74 100 109 123 136) → i = 0

SG a j x … (dalam nilai ASCII **83** 71 97 106 120 133) → i = 1

A [d r Del (dalam nilai ASCII **65** 91 100 114 127) → i = 2

R [i v (dalam nilai ASCII **82** 91 105 118) → i = 3

O] j (dalam nilai ASCII **79** 93 106) → i = 4

Hasil dekripsi baris ke empat (i = 4) akan digunakan sebagai *chipertext* pada proses dekripsi baris ke lima, sehingga :

i = 5; j ≥ 5;

M_{55} = $(C_{[5-1]5} – (3 * R[\ 5\ ]))$ mod 256

= $(C_{[4]5} – (3 * 5))$ mod 256

= ( ] – 15) mod 256

= (93 – 15) mod 256

= 78 (huruf " N " dalam karakter ASCII 256)

M_{56} = $(C_{[5-1]6} – (3 * R[\ 5\ ]))$ mod 256

= $(C_{[4]6} – (3 * 5))$ mod 256

= (j – 15) mod 256

= (106 – 15) mod 256

= 91 (huruf " [" dalam karakter ASCII 256)

hasil dari dekripsi baris ke lima (i = 5) adalah N[

Hasil dekripsi sampai pada tahap baris ke lima (i = 5) dapat dilihat di bawah ini :

W J d m { ˆ	(dalam nilai ASCII 87 74 100 109 123 136)	→ i = 0
SG a j x …	(dalam nilai ASCII **83** 71 97 106 120 133)	→ i = 1
A [d r Del	(dalam nilai ASCII **65** 91 100 114 127)	→ i = 2
R [i v	(dalam nilai ASCII **82** 91 105 118)	→ i = 3
O] j	(dalam nilai ASCII **79** 93 106)	→ i = 4
N[	(dalam nilai ASCII **78** 91)	→ i = 5

Hasil dekripsi baris ke lima (i = 5) akan digunakan sebagai *chipertext* pada proses dekripsi baris ke lima, sehingga :

i = 6; j ≥ 6;

M_{56} = $(C_{[6-1]6} – (3 * R[\ 6\ ]))$ mod 256

= $(C_{[5]6} – (3 * 6))$ mod 256

= ([– 18) mod 256

= (91 – 18) mod 256

= 73 (huruf " I " dalam karakter ASCII 256)

hasil dari dekripsi baris ke enam (i = 6) adalah I

Hasil dekripsi sampai pada tahap baris ke enam (i = 6) dapat dilihat di bawah ini :

W J d m { ^ (dalam nilai ASCII 87 74 100 109 123 136) → i = 0

SG a j x … (dalam nilai ASCII **83** 71 97 106 120 133) → i = 1

A [d r Del (dalam nilai ASCII **65** 91 100 114 127) → i = 2

R [i v (dalam nilai ASCII **82** 91 105 118) → i = 3

O] j (dalam nilai ASCII **79** 93 106) → i = 4

N[(dalam nilai ASCII **78** 91) → i = 5

I (dalam nilai ASCII **73**) → i = 6

Penentuan karakter yang ditetapkan sebagai plainteks (asli) dilakukan berdasarkan formula M_{ij} pada nilai j = (N+i)-N pada masing-masing baris. Sehingga didapatkan *plaintext* adalah SARONI(sama seperti teks aslinya).

3.3 Perancangan Sistem

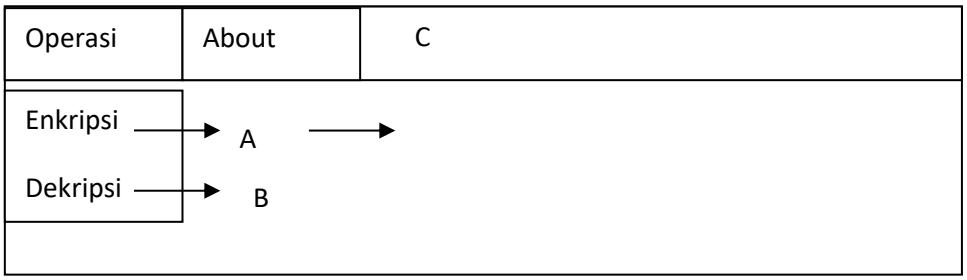
Setelah melakukan analisa terhadap algoritma *Markle Hellman* pada penyandian data, maka tahap selanjutnya adalah melakukan perancangan sistem yang akan digunakan sebagai *interface* proses implementasi. Adapun tahapan perancangan yang dilakukan meliputi perancangan proses serta perancangan antarmuka (*interface*) program. Perancangan sistem bertujuan untuk menggambarkan proses dan implementasi penyandian dengan menggunakan algoritma ini.

3.3.1 Perancangan *Interface* Program

Setelah perancangan *data* dilakukan, maka tahap selanjutnya adalah melakukan perancangan *interface* program. Penulis merancang satu buah *form* untuk mengimplementasikan penyandian *data*. *Form* ini dilengkapi dengan komponen untuk memilih operasi *data*. Perancangan *form* dapat digambarkan pada gambar 3.1 di bawah ini :

1. Form Menu Utama

   Form ini digunakan sebagai menu utama dari proses penyandian data. Dimana dalam form ini ada dua menu utama yaitu operasi dan about



Gambar 3.1 Form Menu Utama

Keterangan :

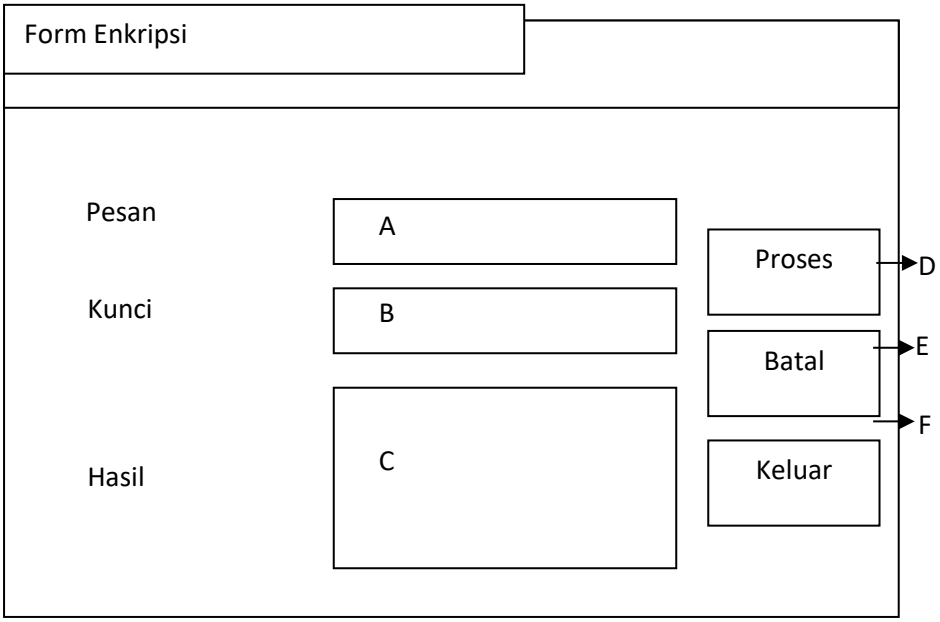
A. Enkripsi. Form ini digunakan untuk melakukan proses penyandian data.

B. Dekripsi. Form ini digunakan untuk melakukan proses pengembalian pesan yang telah disandikan.

C. About Form ini digunakan untuk melihat indentitas programmer.

2. Form Enkripsi

Form ini digunakan untuk menyandikan data, dimana dalam proses ini terdiri dari pesan yang akan disandikan dan kunci yang digunakan untuk menyandikan data



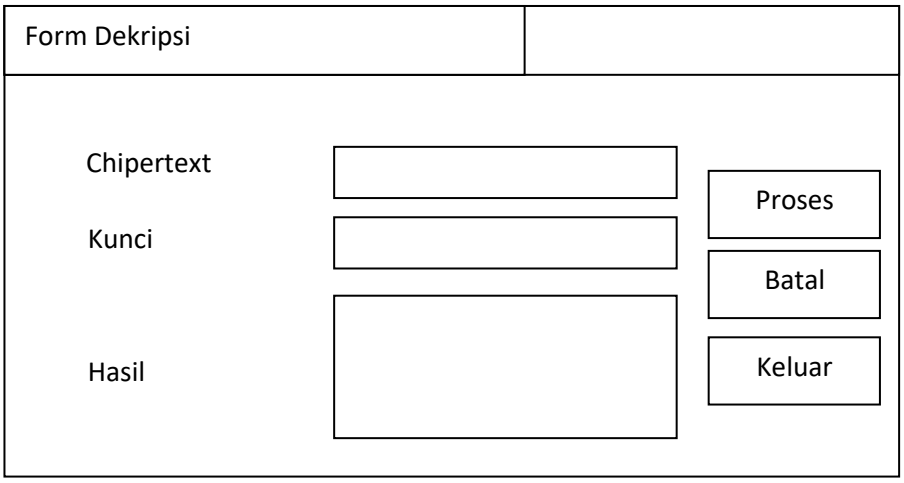
Gambar 3.2 Form Enkripsi

Keterangan :

A. Textbox ini digunakan untuk memasukkan pesan yang akan disandikan.
B. Textbox ini digunakan untuk memasukkan kunci yang digunakan untuk menyandikan data.
C. Textbox ini digunakan untuk menampilkan hasil dari penyandian.
D. Tombol proses digunakan untuk mengeksekusi jalannya program.
E. Tombol batal digunakan untuk membatalkan proses penyandian data.
F. Tombol keluat digunakan untuk keluar dari aplikasi.

3. Form Dekripsi

Form ini digunakan untuk mengembalikan data kebentuk semula, dimana dalam proses ini hanya cukup memasukkan kunci, maka sistem akan mencari apa pesan semula.



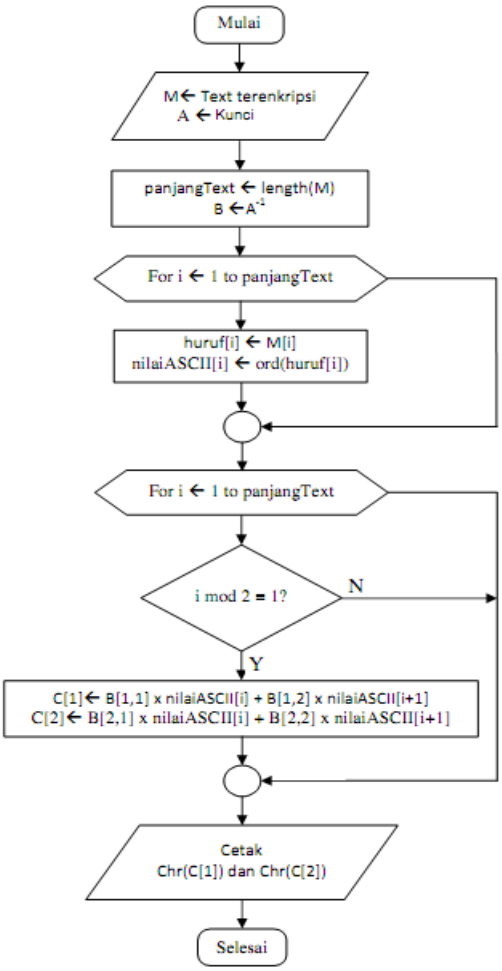
Gambar 3.3 Form Dekripsi

Keterangan :

A. Textbox ini digunakan untuk memasukkan pesan yang telah disandikan.
B. Textbox ini digunakan untuk memasukkan kunci yang digunakan untuk menyandikan data.
C. Textbox ini digunakan untuk menampilkan hasil dari proses enkripsi.
D. Tombol proses digunakan untuk mengeksekusi jalannya program.
E. Tombol batal digunakan untuk membatalkan proses penyandian data.
F. Tombol keluar digunakan untuk keluar dari aplikasi.

3.3.2 Flowchart Enkripsi dan Dekripsi Merkle Hellman

Adapun flowchart dari proses enkripsi dan dekripsi dari algoritma Merkle Hellman adalah seperti gambar dibawah ini



Gambar 3.4 Flowchart Dekripsi Algoritma Merkle Hellman

BAB IV

ALGORITMA DAN IMPLEMENTASI

4.1 Algoritma

Sebagai upaya dalam mempermudah proses penulisan *coding* dengan bahasa pemrograman yang telah ditetapkan sebelumnya, maka penulis merancang algoritma program yang digunakan sebagai bahan acuan. Bentuk rancangan algoritma program dari implementasi algoritma *Merkle Hellman* pada penyandian *record database* dapat diuraian sebagai berikut :

1. Algoritma Penyandian Pesan (Enkripsi)

   Algoritma ini merupakan cara kerja penyandian dengan menggunakan algoritma *Merkle Hellman* pada setiap pesan yang dipilih. Aksi ini akan dieksekusi setelah menerima input dari penekanan tombol Sandikan.

   Mulai

   Input :

   pPes : jumlah karakter pesan

   K : kunci/nilai transposisi karakter record

   R : baris faktor pengali

   i,j : posisi baris dari R, posisi karakter pada cRec

Proses :

{untuk proses pertama}

For i = 1 to pPes

if i = 1 then {untuk baris pertama }

For j = i to pPes

M_{1j} ← char((asc($cRec_{(j)}$)) + (K * R[1])) Mod 256)

Endfor

h 1 ← char (M_{1j})

elseif i >=2 {untuk baris kedua dan seterusnya }

j ←>= i

For j = i to pPes

M_{ij} ← char((asc(h 1 $_{(i-1)j}$)+ (K * R[i])) Mod 256)

Endfor

tempM1 ← char((pPes + asci(M_{ij} [i]))- pPes)

Endfor

end if

hSandi1 ← tempM

Endfor

{untuk proses kedua}

For i = 1 to pPes

if i = 1 then {untuk baris pertama }

For j = i to pPes

M_{1j} ← char((asc($hSandi1_{(j)}$)) + (K * R[1])) Mod 256)

Endfor

h 1 ← char (M_{1j})

elseif i >=2 {untuk baris kedua dan seterusnya }

j ← ≤ (cRec + 1) – i

For j = i to pPes

M_{ij} ← (asc($h\ 1_{[i-1]j}$)+ K + R[i]) Mod 256

Endfor

TempM2 ← char ((cRec + 1) – M_{ij})

Endfor

end if

hSandi2 ← tempM2

Endfor

Output :

Enkripsi ← hSandi2

Selesai

2. Algoritma Pengembalian Pesan(Dekripsi)

Algoritma ini merupakan algoritma pengembalian pesan tersandi menjadi teks pesan asli (pesan sebelum disandikan). Aksi ini akan dieksekusi setelah menerima input dari penekanan tombol Kembalikan Pesan ke Teks Asli.

Mulai

Input :

cCip = karakter pesan tersandi

pCip = jumlah karakter cCip

K = kunci/nilai transposisi karakter pesan tersandi

R = baris faktor pengali

i,j = posisi bari dari R, posisi karakter pada cCip

Proses :

{untuk proses pertama}

For i = 1 to pCip

j ← ≤ (pCip + 1) – 1

For j = i to pCip

Mij ← (asc($cCip_{[i]}$)– K * R[i]) Mod 256

Endfor

hDec1 ← char (M_{ij})

Endfor

{untuk proses Kedua}

For i = 1 to pCip

For j = i to pCip

Mij ← (asc(hDec1 [i]) – K * R[i]) Mod 256

Endfor

Endfor

hDec2 ← char (M_{ij})

Output :

Dekrip ← hDec2

Selesai

4.2 Implementasi

Pada proses implementasi akan dilakukan penyandian terhadap *pesan* Proses penyandian pesan akan dilakukan berdasarkan kaidah dan aturan algoritma *Merkle Hellman* hingga menghasilkan pesan yang tersandi.

4.2.1 Tampilan Hasil Implementasi

Sesuai dengan perancangan *form* sebagai *interface* pengguna dalam mengimplementasikan penyandian pesandengan algoritma *merkle Hellman*, maka dihasilkan tampilan hasil implementasi sesuai dengan yang dilakukan pada tahap perancangan sebelumnya

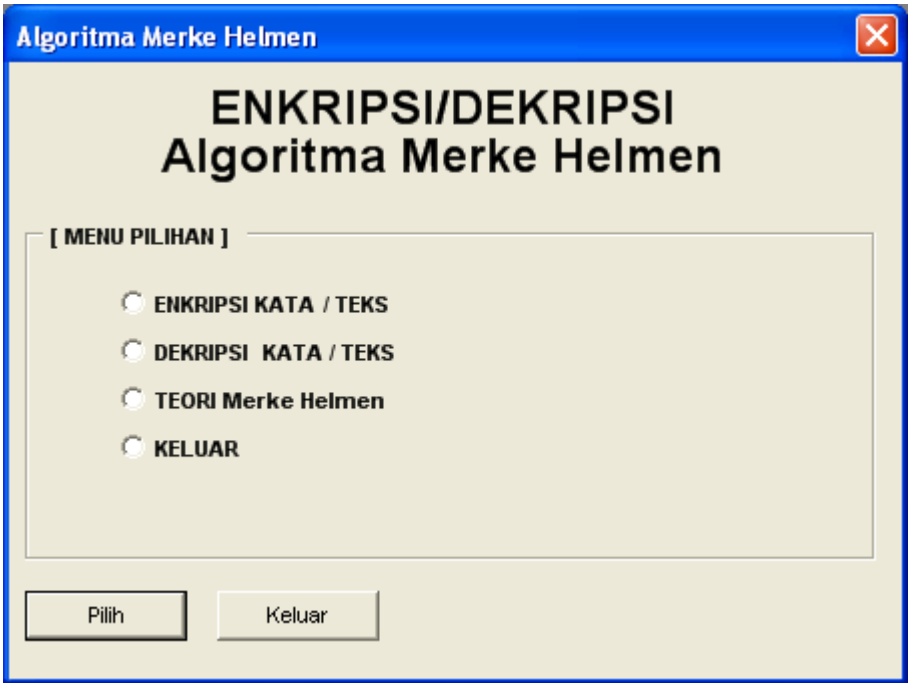
4.2.2 Hasil Pengujian

Setelah mendapatkan hasil tampilan *form interface* implementasi, selanjutnya dilakukan pengujian terhadap penerapan algoritma *Merkle Hellman* melalui *interface* yang telah dirancangtersebut. Adapun tahap-tahap yang dilakukan dalam pengujian adalah sebagai berikut :

1. Pengujian Menu

   Pengujian menu dilakukan dengan tujuan memeriksa interasi antara menu yang telah ada di dalam *form* penyandian *pesan database* dengan hasil proses

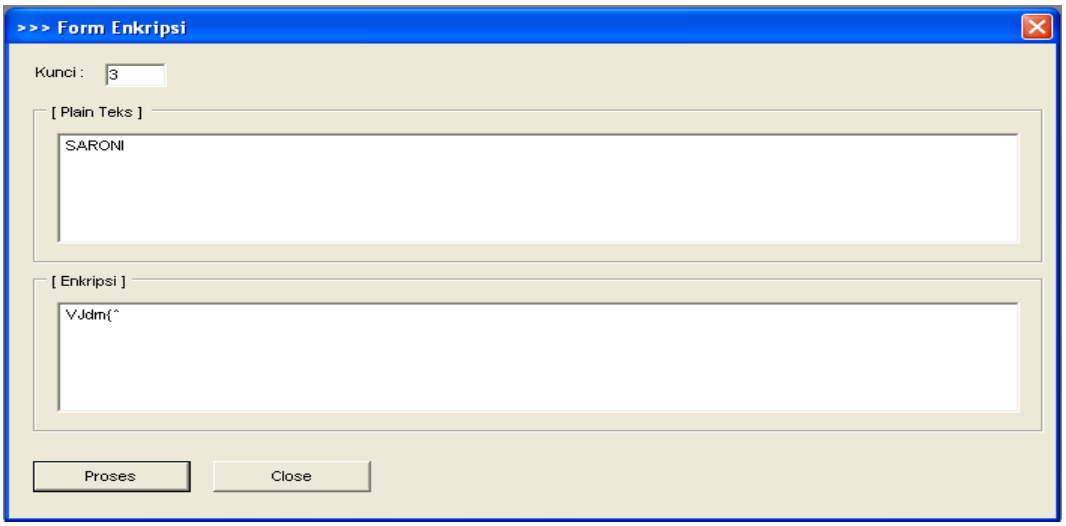
sebagai *output* dari interaksi pemilihan menu. Pengujian pemilihan menu *about* berhasil menyajikan informasi tentang aplikasi ini.



Gambar 4.1 Hasil Pengujian Menu Utama

2. Pengujian *Form* Penyandian Pesan (Enkripsi)

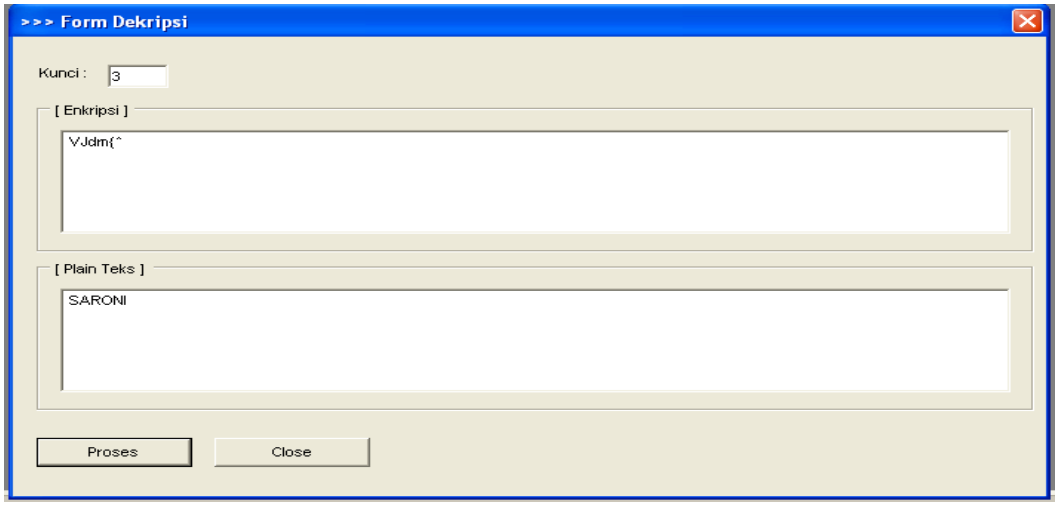
Pada gambar 4.2 dibawah dapat dijelaskan bahwa proses peyandian *pesan* yang telah dipilih berhasil dilakukan, dimana semua *pesan* yang ada pada textbox dirubah dalam bentuk simbol-simbol yang tidak lagi sesuai dengan teks *pesan* aslinya.



Gambar 4.2 Hasil Pengujian Proses Penyandian Pesan

3. Pengujian *Form* Pegembalian Pesan (Dekripsi)

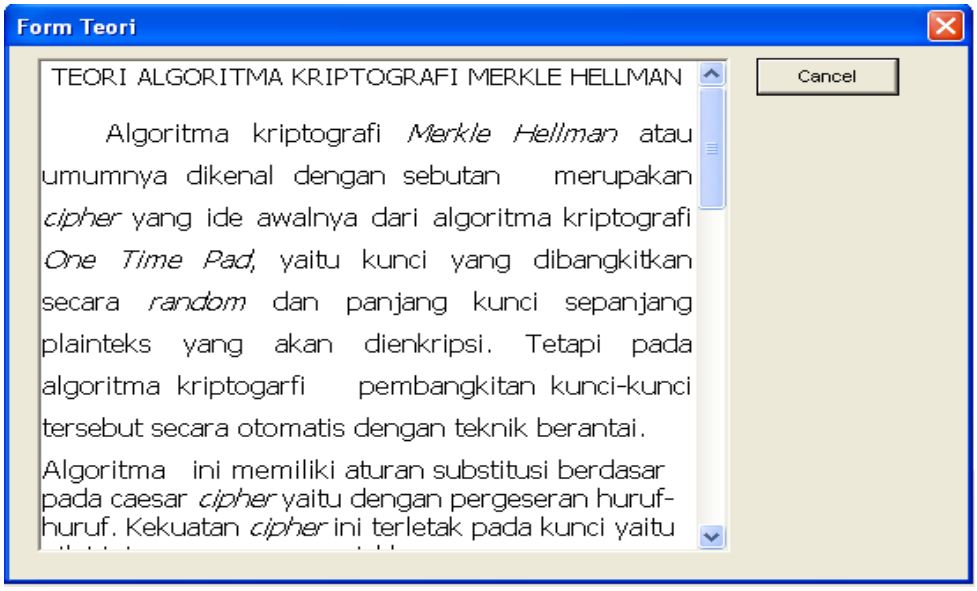
   Pada gambar 4.3 dibawah dapat dijelaskan bahwa proses pengembalian*pesan* yang telah dipilih berhasil dilakukan, dimana semua *pesan* yang ada pada textbox dirubah dalam pada teks *pesan* aslinya.



Gambar 4.3 Hasil Pengujian Proses Pengembalian Pesan

4. Penguraian *Form* Teori

Pada gambar 4.4 dibawah dapat dijelaskan bahwa proses teoriyang telah menjadi bagian algoritma yang digunakan untuk memproses enkripsi dan dekripsi dengan algorima merkle hellman, dimana penjelasan tentang algorima merkle hellman diterangkan di dalam form di bawah ini.



Gambar 4.4 Hasil Penguraian Form Teori

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah melakukan analisa terhadap penerapan algoritma *Merkle Hellman* pada penyandian pesan , maka penulis dapat menarik beberapa kesimpulan sebagai berikut :

1. Algoritma *Merkle Hellman* melakukan proses penyandian pada setiap *pesan* dari pesan yang telah dipilih sebanyak dua kali (secara ganda) dimana nilai subtitusi setiap karakter *pesan* tergantung pada nilai hasil perkalian kunci dengan bilangan faktor pengali yang terbentuk.
2. Hasil akhir yang digunakan adalah hasil proses penyandian kedua dimana hasilnya tidak memiliki kemiripan dengan pesan asli. Pembentukan bilangan faktor pengali dalam proses penyandian dapat menggunakan bilangan integer positif, prima positif, ataupun bilangan yang dapat ditentukan sendiri oleh pengguna, sehingga mampu mempersulit pemecahan sandi yang dihasilkan.
3. Perancangan atau Pembentukan kunci dengan algoritma merkle hellman dan pendekatan matematika pada kriptografi dengan menggunakan algoritma merkle hellman .

1.2 Saran

Adapun yang menjadi saran dalam penulisan skripsi ini adalah sebagai berikut :

1. Implementasi algoritma *Merkle Hellman* dapat dikembangkan dalam menyandikan jenis citra *image* maupun *audio* dan *video*.
2. Pengembangan komponen sistem yang mampu melakukan proses penyandian pesan dengan hanya menentukan nama lokasi pesan. Pelebaran jumlah karakter pesan dan kombinasi kunci serta faktor pengali yang acak, sangat baik dalam upaya peningkatan keamanan dan kesulitan pemecahan hasil penyandian. Implementasi algoritma *Merkle Hellman* dapat dikembangkan pada aplikasi yang berbasis *online*
3. Penyandian terhadap pesan hendaknya dikembangkan tidak hanya menggunakan algoritma *merkle Hellman*, tetapi dapat dilakukan dengan mengimplementasikan algoritma penyandian yang lain. Kekuatan dan keamanan penyandian tidak hanya tergantung pada kunci yang digunakan, melainkan bagaimana algoritma yang digunakan dapat menghasilkan sandi-sandi baru yang mampu memberikan tingkat kerumitan pemecahan yang tinggi.

DAFTAR PUSTAKA

Rinaldi Munir, Kriptografi, Keamanan Data, Penerbit Informatika Bandung 2005.

http://www.cix.co.uk/~klockstone/wake.htm, tanggal 11 Juli 2005.

K. Jusuf Ir, M.T., Kriptografi, **Keamanan Internet dan Jaringan Komunikasi**, Penerbit Informatika Bandung, 2002.

http://www.cix.co.uk/~klockstone/hereward.htm, tanggal 11 Juli 2005.

Maxicom, ***Microsoft Visual Studio 2008***, 2010.

http://eprint.iacr.org/2001/065.pdf, tanggal 11 Juli 2005.

S. Bruce, ***Applied Crytography***, Second Edition, John Wiley & Sons, Inc, 1996.

Cryptography FAQ (06/10: Public Key Cryptography). (2006).

http://www.faqs.org/faqs/cryptographyfaq/part06/. Tanggal.akses: 26 Desember 2006 .

Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. Jurnal Teknik dan Informatika, 5(2), 13-19.

Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). Jurnal Media Informatika Budidarma, 2(2).

Rahim, R., Supiyandi, S., Siahaan, A. P. U., Listyorini, T., Utomo, A. P., Triyanto, W. A. & Khairunnisa, K. (2018, June). TOPSIS Method Application for Decision Support System in Internal Control for Selecting Best Employees. In Journal of Physics: Conference Series (Vol. 1028, No. 1, p. 012052). IOP Publishing.

Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. Int. J. Sci. Res. Sci. Technol, 3(6), 470-473.

Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 100-109.

Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.

Khairul, k., ilhamiarsyah, u., wijaya, r. F., & utomo, r. B. (2018, september). Implementasi augmented reality sebagai media promosi penjualan rumah. In *seminar nasional royal (senar)* (vol. 1, no. 1, pp. 429-434).

Siahaan, A. P. U., Aryza, S., Nasution, M. D. T. P., Napitupulu, D., Wijaya, R. F., & Arisandi, D. (2018). Effect of matrix size in affecting noise reduction level of filtering.

Siahaan, MD Lesmana, Melva Sari Panjaitan, and Andysah Putera Utama Siahaan. "MikroTik bandwidth management to gain the users prosperity prevalent." *Int. J. Eng. Trends Technol* 42.5 (2016): 218-222.

Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." IT Journal Research and Development 2.1 (2017): 1-11.