

**IMPLEMENTASI KRIPTOGRAFI
UNTUK KEAMANAN DATA MENGGUNAKAN
METODE *WORD AUTO KEY ENCRYPTION* (WAKE)**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Oleh :

Nama : ARDHITYA DWI PRAKOSA

NIM : 10 523 215

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2014

**IMPLEMENTASI KRIPTOGRAFI
UNTUK KEAMANAN DATA MENGGUNAKAN
METODE *WORD AUTO KEY ENCRYPTION* (WAKE)**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Oleh :

**Nama : ARDHITYA DWI PRAKOSA
NIM : 10 523 215**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2014

LEMBAR PENGESAHAN PEMBIMBING

IMPLEMENTASI KRIPTOGRAFI

UNTUK KEAMANAN DATA MENGGUNAKAN

METODE *WORD AUTO KEY ENCRYPTION* (WAKE)

TUGAS AKHIR

Disusun oleh :

Nama : ARDHITYA DWI PRAKOSA

NIM : 10 523 215



Yogyakarta, Februari 2014

Dosen Pembimbing

Ahmad Luthfi, S.Kom, M.Kom

LEMBAR PENGESAHAN PENGUJI

IMPLEMENTASI KRIPTOGRAFI

UNTUK KEAMANAN DATA MENGGUNAKAN

METODE *WORD AUTO KEY ENCRYPTION* (WAKE)

TUGAS AKHIR

Disusun oleh :

Nama : ARDHITYA DWI PRAKOSA

NIM : 10 523 215

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Februari 2014

Tim Penguji,

Ahmad Luthfi, S.Kom., M.Kom

Ketua

Syarif Hidayat, S.Kom., MIT.

Anggota I

Ari Sujarwo, S.Kom., MIT (Hons)

Anggota II

Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom.

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini,

Nama : ARDHITYA DWI PRAKOSA

NIM : 10 523 215

Tugas Akhir dengan judul :

IMPLEMENTASI KRIPTOGRAFI UNTUK KEAMANAN DATA MENGGUNAKAN METODE *WORD AUTO KEY ENCRYPTION (WAKE)*

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya akan siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, Februari 2014

Yang Membuat Pernyataan,

(ARDHITYA DWI PRAKOSA)

HALAMAN PERSEMBAHAN

Dengan segala ketulusan dan kerendahan hati, sebuah karya kecil ini kami persembahkan kepada :

1. **ALLAH Subhanahu Wa Ta'ala**

Puji syukur hamba panjatkan atas limpahan Rahmat dan Karunia yang Kau persembahkan ke dalam kehidupan hamba. Tiada lah selain Allah, Tuhan semesta alam. Yang menurunkan rahmat ilmu kepada semesta alam. Terima kasih atas kemudahan dan kelancaran yang telah Kau curahkan dalam menyelesaikan karya kecil ini.

2. **Muhammad Salallahu Alaihi Wassalam**

Junjungan hamba Nabi Muhammad SAW, yang membimbing umatnya dari zaman jahiliah menuju zaman terang benderang. Terima kasih atas pengorbanan yang Kau tempuh demi menurunkan rahmat Islam bagi kami untuk keselamatan Dunia Akhirat. Terima kasih atas pedoman yang Kau lukiskan dalam Kitab Suci Al-Qur'an sebagai pedoman umat-Mu menjalani hidup. Sholawat dan Salam senantiasa hamba panjatkan untuk Kau, keluarga-Mu, dan para Sahabat-Mu Ya Muhammad SAW.

3. **Bapak dan Ibu**

Terima kasih atas dukungan, semangat, dan doanya Bapak Hardiman dan Ibu Endang Trisilawati. Kehangatan kasih sayang kalian yang membakar api semangatku untuk dapat menyelesaikan karya kecil ini.

4. **Seluruh Keluarga Besar**

Terima kasih buat mas Hendra, dek puput, mbak intan, bulik Puji, om Iyo, dek Ditra, dek Rio, dan seterusnya yang tidak bisa saya sebut satu per satu. Terima kasih atas segala doa dan dukungan kalian.

5. **Ira Maya Fathonah**

Kamu lah inspirasiku. Terima kasih atas segala yang telah kamu berikan untukku. Semua doa, dukungan, semangat, dan semua pengorbananmu yang tak akan sanggup aku ungkapkan dengan kata-kata. Kamu lah yang membangkitkanku saat malas hinggap dalam pikiranku.

6. **Bapak Ahmad Luthfi, S.Kom, M.Kom**

Terima kasih untuk bapak Luthfi atas segala kepercayaan, kebaikan, dan kesabaran bapak dalam membimbing saya untuk menyelesaikan karya kecil ini. Tidak akan saya lupakan apa yang telah bapak ajarkan dalam masa aktif perkuliahan. Terima kasih atas ilmu yang telah diberikan kepada saya, semoga Allah SWT membalas semua kebaikan bapak dan selalu melimpahkan Rahmat kepada bapak Luthfi.

7. **Penghuni D'Team Lo (Dedemit Team Location)**

Terima kasih untuk mas Hendra, bang Unik, mas Dodo, mas Eko, om Tyo, mas Putut, Ami, Slamet, Iwan atas segala dukungan kalian. Kebersamaan kita berlaku selamanya. Semua kebaikan kalian tidak akan pernah terlupakan.

8. **Sahabat-Sahabat Yang Selalu Mendukungku**

Terima kasih untuk mbak Hesti, mbak Jessy, dek Idjah, dan Salman atas segala doa dan dukungan kalian. Terima kasih telah setia menemani dan memberikan semangat disaat rasa malas menghampiriku. Kebaikan dan kebersamaan kita akan selalu ku kenang.

9. **Teman-Temanku**

Terima kasih kepada Anang, Fajar, Irek, Metta, Dwiki, Fany, Maman, Arif, Aisyah, Susan, dan lain-lain yang tidak bisa saya sebut satu per satu. Terima kasih atas segala dukungan kalian. Semoga kita semua dapat menjadi orang yang berguna bagi bangsa dan almamater kita Universitas Islam Indonesia.

MOTTO

“Sesungguhnya dalam penciptaan langit dan bumi dan pergantian siang dan malam terdapat (tanda-tanda kebesaran Allah) bagi orang-orang yang memiliki akal ”

(QS. Al Imran: 190)

“Ingatlah, sesungguhnya wali-wali Allah itu, tidak ada kekhawatiran terhadap mereka dan tidak pula mereka bersedih hati (yaitu) orang-orang yang beriman dan mereka selalu bertakwa”

(QS. Yunus:62-63)

“Sesungguhnya orang yang paling mulia diantara kamu disisi Allah ialah orang yang paling bertakwa di antara kamu ”

(QS. Al Hujarat:13)

“Sesungguhnya kami telah menciptakan manusia dalam bentuk yang sebaik-baiknya, kemudian kami kembalikan dia ke tempat yang serendah-rendahnya, kecuali orang-orang yang beriman dan mengerjakan amal shalih; maka bagi mereka pahala yang tiada putus-putusnya”

(QS.Al Bayyinah:4-6)

Hidup berawal dari mimpi, jangan takut bermimpi karena dengan mimpi kamu akan dapat mengetahui tujuan hidupmu

(Ardhitya Dwi Prakosa)

Kesuksesan adalah pola pikir, atur pola pikirmu untuk sukses maka kamu akan menemukan jalan kesuksesanmu

(Ardhitya Dwi Prakosa)

KATA PENGANTAR



Assalamualaikum Wr. Wb.

Segala puji hanya milik Allah Yang Maha Perkasa Lagi Maha Pengampun. Dia Yang Mengulurkan tangan-Nya di siang hari agar bertaubat siapa yang berlaku maksiat dimalam hari dan yang mengulurkan tangan-Nya dimalam hari agar bertaubat orang yang melakukan dosa di siang hari.

Aku memuji-Nya atas karunia agung keutamaan, nikmat, kemuliaan dan kebaikan-Nya. Shalawat dan salam teruntuk Nabi dan Rasul-Nya yang terpilih, Muhammad SAW. Juga semoga tersampaikan kepada para keluarga dan para sahabatnya sekalian hingga hari kiamat.

Alhamdulillah, dengan Rahmat Allah SWT penulis dapat menyelesaikan laporan tugas akhir yang berjudul **Implementasi Kriptografi Untuk Keamanan Data Dengan Metode Word Auto Key Encryption (WAKE)** dengan sebaik-baiknya penulis lakukan.

Laporan tugas akhir ini disusun sebagai salah satu syarat guna memperoleh gelar Sarjana Teknik Informatika pada Universitas Islam Indonesia. Dan juga sebagai sarana untuk mempraktekkan secara langsung ilmu dan teori yang telah diperoleh selama menjalani masa studi di Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Laporan tugas akhir ini berisikan perancangan, implementasi dan pengujian perangkat lunak kriptografi dengan metode WAKE. Sehingga laporan tugas akhir ini dapat dijadikan acuan dalam mempelajari cara kerja dan proses-proses dalam metode WAKE.

Dalam penyusunan laporan tugas akhir ini, penulis banyak mendapatkan dukungan dan bantuan baik materiil maupun spirituil dari berbagai pihak. Oleh karenanya dalam kesempatan kali ini, penulis dengan segala kerendahan hati ingin mengucapkan terima kasih kepada:

1. Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya kepada penulis sehingga dapat menyelesaikan laporan tugas akhir ini.
2. Nabi Muhammad SAW yang telah membuka pintu Cahaya untuk dapat memperbaiki dari jaman Jahiliah menjadi Zaman yang mulia penuh dengan keindahan, bersyukur, atas Nikmat Allah SAW.
3. Orang Tua yang telah memberikan doa dan restu, serta dukungan materiil dan spirituil kepada penulis sehingga dapat menyelesaikan penulisan laporan tugas akhir dengan baik.
4. Saudara dan Saudari penulis yang telah memberikan dukungan kepada penulis sehingga memiliki semangat dalam menyelesaikan laporan tugas akhir ini.
5. Bapak Prof. Dr. Edy Suandi Hamid, M.Ec. selaku Rektor Universitas Islam Indonesia dan seluruh jajaran Rektorat Universitas Islam Indonesia.
6. Bapak Ir. Gumbolo HS., M.Sc. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
7. Bapak Yudi Prayudi, S.Si., M.Kom. selaku Ketua Jurusan Teknik Informatika.
8. Bapak Ahmad Luthfi, S.Kom, M.Kom. selaku dosen pembimbing yang telah memberikan pengarahan, bimbingan, serta masukan selama pelaksanaan tugas akhir dan penulisan laporan.
9. Segenap Dosen Jurusan Teknik Informatika yang telah memberikan ilmu pengetahuan yang sangat berharga selama masa studi.
10. Seluruh dukungan sahabat yang mendukung dalam penyelesaian laporan tugas akhir ini serta kepada seluruh pihak yang memberikan dukungan yang tidak dapat penulis sebutkan satu persatu.

Semoga Allah SWT melimpahkan segala Rahmat dan Hidayah-Nya kepada kalian semua dan membalas semua kebaikan yang kalian berikan kepada penulis.

Penulis menyadari terdapat banyak kekeliruan dan kekurangan dalam penyusunan laporan tugas akhir ini masih. Oleh karenanya penulis ingin

mengucapkan permohonan maaf atas kekeliruan dan kesalahan ini. Penulis sangat mengharapkan saran, kritik dan masukan yang membangun dalam penyempurnaan laporan tugas akhir ini.

Akhir kata semoga laporan tugas akhir ini dapat bermanfaat bagi penulis dan para pembaca.

Wassalamualaikum Wr. Wb.

Yogyakarta, Februari 2014

ARDHITYA DWI PRAKOSA

SARI

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek-aspek pada keamanan data. Kriptografi bertujuan untuk mengamankan data yang bersifat rahasia agar data tersebut tidak diketahui oleh orang yang tidak berkepentingan. Metode WAKE memiliki empat proses utama yaitu proses pembentukan tabel S-Box, proses pembentukan kunci, enkripsi, dan dekripsi. Untuk melakukan implementasi dan penjelasan proses-proses kriptografi WAKE diperlukan sebuah perangkat lunak. Penelitian ini menghasilkan sebuah perangkat lunak kriptografi yang mampu menerima masukan (input) 16 karakter untuk kata kunci, dan 1000 karakter untuk plaintext. Perangkat lunak mengizinkan pengguna untuk memberikan masukan berupa file.txt. Metode ini cukup cepat dalam implementasinya pada perangkat lunak dan terdapat penjelasan setiap prosesnya. Sehingga dapat digunakan untuk mengamankan data dan mengetahui tahapan-tahapan dalam setiap prosesnya.

Kata kunci : kriptografi, *Word Auto Key Encryption*, S-Box.

TAKRIR

<i>Access Control</i>	: Akses control
<i>Activity Diagram</i>	: Diagram aktifitas
<i>And</i>	: Dan
<i>Authentication</i>	: Keaslian
<i>Auto</i>	: Otomatis
<i>Availability</i>	: Tersedia
<i>Cipher</i>	: Sandi
<i>Ciphertext</i>	: Pesan yang telah disandikan
<i>Confidentiality</i>	: Keyakinan
<i>Cryptogram</i>	: Pesan yang disandikan
<i>Cryptography</i>	: Kriptografi
<i>Cryptos</i>	: Rahasia
<i>Encryption</i>	: Enkripsi
<i>False</i>	: Salah
<i>Form</i>	: Halaman
<i>Graphein</i>	: Tulisan
<i>Import</i>	: Mengambil file dari komputer
<i>Input</i>	: Masukan
<i>Integrity</i>	: Kejujuran
<i>Key</i>	: Kunci
<i>Main</i>	: Utama
<i>Nonrepudiation</i>	: Tidak ada perubahan
<i>Or</i>	: Atau
<i>Output</i>	: Keluaran
<i>Plaintext</i>	: Pesan asli
<i>Private</i>	: Pribadi
<i>Public</i>	: Publik
<i>Shift Left</i>	: Menggeser ke kiri

<i>Shift Right</i>	: Menggeser ke kanan
<i>Standard</i>	: Standar
<i>Stream Cipher</i>	: Salah satu metode dalam kriptografi
<i>Tool</i>	: Fitur
<i>True</i>	: Benar
<i>Word</i>	: Huruf

DAFTAR ISI

HALAMAN JUDUL	ii
LEMBAR PENGESAHAN PEMBIMBING.....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PERSEMBAHAN	vi
MOTTO	viii
KATA PENGANTAR.....	ix
SARI.....	xii
TAKARIR.....	xiii
DAFTAR ISI.....	xv
DAFTAR TABEL	xix
DAFTAR GAMBAR.....	xx
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	5
2.1 Pengertian Kriptografi.....	5
2.2 Sistem Kriptografi.....	7
2.2.1 Sistem Kriptografi Simetris	7
2.2.2 Sistem Kriptografi Asimetris	8
2.3 Dasar-dasar Matematika Kriptografi	10
2.3.1 AND	10
2.3.2 OR	10
2.3.3 XOR.....	11

2.3.4 Penjumlahan Modulo	12
2.3.5 Penggeseran Bit.....	12
2.3.6 Sistem Bilangan	13
2.3.6.1 Konversi Bilangan Biner ke Bilangan Heksadesimal ...	13
2.3.6.2 Konversi Bilangan Biner ke Bilangan Desimal	14
2.3.6.3 Konversi Bilangan Desimal ke Bilangan Heksadesimal	14
2.3.6.4 Konversi Bilangan Desimal ke Bilangan Biner	15
2.4 Algoritma Kriptografi Klasik	15
2.5 <i>Word Auto Key Encryption</i> (WAKE).....	16
2.5.1 Proses Pembentukan Tabel S-Box	17
2.5.2 Proses Pembentukan Kunci	18
2.5.3 Proses Enkripsi.....	19
2.5.4 Proses Dekripsi	19
2.6 Penelitian Sebelumnya	20
BAB III METODOLOGI	21
3.1 Metodologi Penelitian	21
3.2 Analisa Kebutuhan Perangkat Lunak.....	22
3.2.1 Analisa Kebutuhan Input.....	22
3.2.2 Analisa Kebutuhan Proses	22
3.2.3 Analisa Kebutuhan Output	22
3.2.4 Analisa Kebutuhan Perangkat Lunak.....	23
3.2.5 Analisa Kebutuhan Perangkat Keras.....	23
3.3 Perancangan Perangkat Lunak.....	23
3.3.1 Proses Pembentukan Tabel S-Box	25
3.3.2 Proses Pembentukan Kunci	27
3.3.3 Proses Enkripsi.....	28
3.3.4 Proses Dekripsi	29
3.4 Perancangan Antarmuka	30
3.4.1 <i>Form Main</i>	31
3.4.2 <i>Form Menu</i> Enkripsi dan Dekripsi	31
3.4.3 <i>Form</i> Proses Enkripsi	32

3.4.4 <i>Form</i> Proses Dekripsi	33
3.4.5 <i>Form</i> Menu Alur Metode WAKE	34
3.4.6 <i>Form</i> Proses S-Box	35
3.4.7 <i>Form</i> Proses Pembentukan Kunci	36
3.4.8 <i>Form Input</i> Proses Enkripsi	37
3.4.9 <i>Form</i> Hasil Tabel S-Box.....	37
3.4.10 <i>Form</i> Hasil Pembentukan Kunci.....	38
3.4.11 <i>Form</i> Hasil Enkripsi	38
3.4.12 <i>Form Input</i> Proses Dekripsi	39
3.4.13 <i>Form</i> Hasil Dekripsi	40
BAB IV HASIL DAN PEMBAHASAN.....	41
4.1 Implementasi	41
4.1.1 Implementasi <i>Form Main</i>	41
4.1.2 Implementasi <i>Form</i> Menu Enkripsi dan Dekripsi.....	42
4.1.3 Implementasi <i>Form</i> Proses Enkripsi	43
4.1.4 Implementasi <i>Form</i> Proses Dekripsi	44
4.1.5 Implementasi <i>Form</i> Menu Alur Metode WAKE	45
4.1.6 Implementasi <i>Form</i> Proses S-Box	46
4.1.7 Implementasi <i>Form</i> Proses Pembentukan Kunci	47
4.1.8 Implementasi <i>Form Input</i> Proses Enkripsi	48
4.1.9 Implementasi <i>Form</i> Hasil Tabel S-Box.....	49
4.1.10 Implementasi <i>Form</i> Hasil Pembentukan Kunci	50
4.1.11 Implementasi <i>Form</i> Hasil Enkripsi	51
4.1.12 Implementasi <i>Form Input</i> Proses Dekripsi	52
4.1.13 Implementasi <i>Form</i> Hasil Dekripsi	53
4.1.14 Implementasi <i>Form</i> Penanganan Kesalahan.....	53
4.2 Pengujian Perangkat Lunak.....	54
4.3 Kelebihan dan Kekurangan Perangkat Lunak	71
4.3.1 Kelebihan Perangkat Lunak.....	72
4.3.2 Kekurangan Perangkat Lunak.....	72

BAB V KESIMPULAN DAN SARAN	73
5.1 Kesimpulan.....	73
5.2 Saran.....	73
DAFTAR PUSTAKA.....	74

DAFTAR TABEL

Tabel 2.1 Aturan Operasi AND	10
Tabel 2.2 Aturan Operasi OR	11
Tabel 2.3 Aturan Operasi XOR	11
Tabel 2.4 Cipher Substitusi	16
Tabel 4.1 Pengujian Manual	56

DAFTAR GAMBAR

Gambar 2.1 Sistem Kriptografi Simetris	8
Gambar 2.2 Sistem Kriptografi Asimetris	8
Gambar 3.1 Metode Waterfall	22
Gambar 3.2 <i>Activity Diagram</i>	24
Gambar 3.3 <i>Flowchart</i> S-Box.....	26
Gambar 3.4 Bagan Pembentukan Kunci.....	27
Gambar 3.5 <i>Flowchart</i> Enkripsi.....	28
Gambar 3.6 <i>Flowchart</i> Dekripsi	29
Gambar 3.7 Perancangan <i>Form Main</i>	31
Gambar 3.8 Perancangan <i>Form</i> Menu Enkripsi dan Dekripsi.....	32
Gambar 3.9 Perancangan <i>Form</i> Proses Enkripsi	33
Gambar 3.10 Perancangan <i>Form</i> Proses Dekripsi	34
Gambar 3.11 Perancangan <i>Form</i> Menu Alur Metode WAKE	35
Gambar 3.12 Perancangan <i>Form</i> Proses S-Box.....	36
Gambar 3.13 Perancangan <i>Form</i> Proses Pembentukan Kunci	36
Gambar 3.14 Perancangan <i>Form Input</i> Proses Enkripsi	37
Gambar 3.15 Perancangan <i>Form</i> Hasil Tabel S-Box.....	38
Gambar 3.16 Perancangan <i>Form</i> Hasil Pembentukan Kunci	38
Gambar 3.17 Perancangan <i>Form</i> Hasil Enkripsi	39
Gambar 3.18 Perancangan <i>Form Input</i> Proses Dekripsi	39
Gambar 3.19 Perancangan <i>Form</i> Hasil Dekripsi	40
Gambar 4.1 <i>Form Main</i>	41
Gambar 4.2 <i>Form</i> Menu Enkripsi dan Dekripsi	42
Gambar 4.3 <i>Form</i> Proses Enkripsi	43
Gambar 4.4 <i>Form</i> Proses Dekripsi.....	44
Gambar 4.5 <i>Form</i> Proses Menu Alur Metode WAKE.....	45
Gambar 4.6 <i>Form</i> Proses S-Box	47
Gambar 4.7 <i>Form</i> Proses Pembentukan Kunci.....	48

Gambar 4.8 <i>Form Input</i> Proses Enkripsi	49
Gambar 4.9 <i>Form Hasil</i> Tabel S-Box	50
Gambar 4.10 <i>Form Hasil</i> Pembentukan Kunci.....	51
Gambar 4.11 <i>Form Hasil</i> Enkripsi	51
Gambar 4.12 <i>Form Input</i> Proses Dekripsi.....	52
Gambar 4.13 <i>Form</i> Proses Dekripsi.....	53
Gambar 4.14 <i>Form</i> Penanganan Kesalahan <i>Input</i> Teks	54
Gambar 4.15 <i>Form</i> Penanganan Kesalahan <i>Input</i> Kunci	54
Gambar 4.16 Pengujian Enkripsi	55
Gambar 4.17 Pengujian Dekripsi	56

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dewasa ini penggunaan komputer untuk transmisi data melalui media komunikasi sudah merupakan hal yang umum dilakukan. Namun demikian, meskipun perkembangan media komunikasi sangat pesat, hal ini membuka peluang bagi pihak-pihak yang tidak bertanggung jawab dengan sengaja melakukan pencurian terhadap pengiriman data melalui saluran komunikasi tersebut. Hal ini yang menjadi dasar munculnya kriptografi. Kriptografi dapat digunakan untuk mengamankan data yang bersifat rahasia agar data tersebut tidak diketahui oleh orang lain yang tidak berkepentingan, yaitu dengan cara mengubah data asli (*plaintext*) menjadi data tersandi (*ciphertext*) yang isinya berbeda dari data aslinya dengan menggunakan sebuah kunci. Data tersandi (*ciphertext*) dapat diubah kembalikan ke dalam bentuk data asli (*plaintext*) dengan menggunakan kunci yang sama. Sehingga hanya orang yang mengetahui kunci yang dapat membuka pesan tersebut.

Cukup banyak metode yang dapat digunakan dalam teknik kriptografi, baik itu klasik maupun modern. Untuk kriptografi klasik diantaranya adalah *Chaesar Cipher*, *Playfair Cipher*, *Shift Cipher*, *Hill Cipher*, *Vigenere Cipher*, *Autokey Cipher*. Sedangkan untuk kriptografi modern diantaranya RSA, Algoritma ElGamal, Algoritma A5, GOSH.

Metode WAKE merupakan salah satu metode *stream cipher* yang telah digunakan secara komersial. WAKE merupakan singkatan dari *Word Auto Key Encryption*. Metode ini ditemukan oleh David Wheeler pada tahun 1993. Metode ini menggunakan kunci 128 bit, dan sebuah tabel 256 x 32 bit. Algoritma kriptografi metode WAKE menggunakan beberapa operasi matematika yaitu XOR, AND, OR dan *Shift Right*. Metode WAKE ini telah digunakan pada

program Dr. Solomon anti virus versi terbaru. Metode WAKE dapat dibagi menjadi beberapa proses yaitu proses pembentukan tabel dan kunci, enkripsi dan dekripsi. Proses penyelesaian metode ini cukup rumit dan sulit untuk dikerjakan secara manual dikarenakan algoritmanya yang cukup panjang dan kompleks. Inti dari metode kriptografi WAKE terletak pada proses pembentukan table S-Box dan proses pembentukan kunci. Tabel S-Box dari metode ini bersifat fleksibel dan berbeda-beda setiap putaran (Dahria, et.al.,2012).

Perangkat lunak kriptografi WAKE ini digunakan untuk keperluan personal. Meskipun dapat juga dikembangkan lebih lanjut untuk keperluan komersial atau bisnis.

Berdasarkan uraian di atas maka penulis ingin membuat tugas akhir dengan judul **“Implementasi Kriptografi Untuk Keamanan Data Menggunakan Metode Word Auto Key Encryption (WAKE)”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan permasalahan yang akan diselesaikan adalah bagaimana implementasi kriptografi yang dapat menjaga kerahasiaan data menggunakan algoritma WAKE?

1.3 Batasan Masalah

Dalam melaksanakan suatu penelitian diperlukan adanya batasan agar tidak menyimpang dari penelitian yang telah direncanakan sehingga tujuan penelitian dapat tercapai. Batasan masalah yang diperlukan yaitu :

- a. Proses enkripsi dan dekripsi yang digunakan adalah algoritma kriptografi WAKE.
- b. Membangun perangkat lunak menggunakan bahasa pemrograman Visual Basic 6.
- c. Karakter kata kunci yang dimasukkan maksimal 16 karakter.
- d. *Plaintext* maksimal 1000 karakter.
- e. *Plaintext* hanya berupa *input*-an manual dan file (.txt)

- f. Apabila *input* berupa file (.txt) maka akan diambil 1000 karakter terdepan dari isi file tersebut.
- g. Jumlah putaran dalam proses pembentukan kunci maksimal 50 putaran.

1.4 Tujuan Penelitian

Tujuan yang hendak dicapai dari penelitian ini adalah membangun aplikasi yang dapat digunakan untuk enkripsi dan dekripsi data dengan menggunakan metode *Word Auto Key Encryption (WAKE)*.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah:

- a. Memenuhi kebutuhan akan keamanan suatu data, sehingga menjamin kerahasiaan, integritas data, dan otentikasi data itu sendiri.
- b. Dapat menjadi wacana baru dalam hal perbaikan terhadap metode kriptografi dan bisa memberikan wawasan untuk pengembangan penelitian selanjutnya.

1.6 Metodologi Penelitian

1.6.1 Analisa masalah

Metode ini untuk menganalisis masalah-masalah yang terjadi, yang menyebabkan mengapa dilakukan penelitian ini.

1.6.2 Analisa Kebutuhan

Metode untuk menganalisis kebutuhan perangkat lunak antara lain dari spesifikasi perangkat keras, analisis perangkat lunak meliputi *input*, *output*, dan langkah-langkah yang dibutuhkan serta antarmuka yang akan digunakan.

1.6.3 Perancangan

Metode perancangan berisi tentang perancangan dari alur jalannya perangkat lunak dan perancangan antarmukanya.

1.6.4 Implementasi

Metode implementasi menerapkan hasil rancangan yang telah disetujui terhadap aplikasi yang dibuat.

1.6.5. Pengujian

Metode pengujian dilakukan untuk menguji aplikasi yang telah selesai dibuat sesuai rancangan sebelum aplikasi tersebut digunakan

1.7 Sistematika Penulisan

Dalam penyusunan tugas akhir ini, sistematika penulisan dibagi menjadi beberapa bab sebagai berikut:

BAB I Pendahuluan, berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II Landasan Teori, membahas tentang masalah-masalah yang berhubungan dengan perancangan aplikasi kriptografi untuk keamanan data serta teori lainnya yang mendukung pembuatan aplikasi.

BAB III Metodologi, berisi tahapan-tahapan dalam perancangan perangkat lunak yang meliputi analisis kebutuhan perangkat lunak, perancangan alur jalannya perangkat lunak, dan perancangan antarmuka perangkat lunak.

BAB IV Hasil dan Pembahasan, berisi implementasi dari perangkat lunak yang dibangun dan pengujian perangkat lunak terhadap kesesuaian dengan kebutuhan dan hasil yang diharapkan.

BAB V Kesimpulan dan Saran, berisi kesimpulan yang diperoleh dari perancangan aplikasi dan saran yang diberikan untuk pengembangan aplikasi lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Pengertian Kriptografi

Keamanan telah menjadi aspek yang sangat penting dari suatu sistem informasi. Sebuah informasi umumnya hanya ditujukan bagi golongan tertentu. Oleh karena itu sangat penting untuk mencegahnya jatuh kepada pihak-pihak lain yang tidak berkepentingan. Untuk melaksanakan tujuan tersebut dirancang suatu sistem keamanan yang berfungsi melindungi suatu informasi.

Salah satu upaya pengamanan data memiliki beberapa aspek penting. Menurut Stalling, ada beberapa tuntutan yang terkait dengan isu keamanan data yaitu (Stalling, 1996):

a. *Confidentiality*

Menjamin bahwa data-data tersebut hanya bisa diakses oleh pihak-pihak tertentu saja.

b. *Authentication*

Baik pada saat mengirim atau menerima informasi, kedua belah pihak perlu mengetahui bahwa pengirim dari pesan tersebut adalah orang yang sebenarnya seperti yang diklaim.

c. *Integrity*

Tuntutan ini berhubungan dengan jaminan setiap pesan yang dikirim pasti sampai pada penerimanya tanpa ada bagian dari pesan tersebut yang diganti, diduplikasi, dirusak, diubah urutannya, dan ditambahkan.

d. *Nonrepudiation*

Nonrepudiation mencegah pengirim maupun penerima mengingkari bahwa mereka telah mengirimkan atau menerima suatu pesan atau informasi. Jika sebuah pesan dikirim, penerima dapat membuktikan bahwa pesan tersebut memang dikirim oleh pengirim yang tertera. Sebaliknya, jika sebuah pesan diterima, pengirim dapat

membuktikan bahwa pesannya telah diterima oleh pihak yang ditujunya.

e. *Access Control*

Membatasi sumber-sumber data hanya kepada orang-orang tertentu.

f. *Availability*

Jika diperlukan setiap saat semua informasi pada sistem komputer harus tersedia bagi semua pihak yang berhak atas informasi tersebut.

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

Kriptografi berasal dari bahasa Yunani yaitu *cryptos* yang artinya rahasia dan *graphein* yang artinya tulisan. Jadi secara bahasa kriptografi dapat diartikan sebagai ilmu dan seni untuk menjaga keamanan pesan (Munir, 2006). Kriptografi secara luas dapat diartikan sebagai ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek-aspek pada keamanan informasi misalnya kerahasiaan, integritas data, otentikasi pengirim atau penerima data, dan otentikasi data (Jusuf Ir, 2002).

Di dalam kriptografi kita akan sering menemukan berbagai istilah atau terminologi. Beberapa istilah yang harus diketahui yaitu :

a. *Pesan*

Pesan adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plainteks*. Agar pesan tidak dapat dimengerti maknanya oleh pihak lain yang tidak berkepentingan, maka pesan perlu disandikan kebentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut *ciphertext* atau *cryptogram*. *Ciphertext* harus dapat ditransformasikan kembali menjadi *plaintext* semula agar penerima dapat membaca isi pesan yang sebenarnya.

b. Pengirim dan penerima

Komunikasi data melibatkan pertukaran pesan antar dua entitas. Pengirim adalah entitas yang mengirimkan pesan kepada entitas lainnya. Penerima adalah entitas yang menerima pesan. Pengirim tentu menginginkan pesan dapat dikirim secara aman, yaitu pengirim yakin bahwa pihak lain tidak dapat membaca isi pesan yang dikirim. Solusinya adalah dengan cara menyandikan pesan tersebut.

c. Enkripsi dan dekripsi

Enkripsi adalah proses penyandian terhadap pesan asli (*plaintext*) menjadi *ciphertext*. Sedangkan dekripsi adalah proses kebalikan dari enkripsi yaitu proses pengembalian data yang telah tersandikan (*ciphertext*) menjadi data asli (*plaintext*).

d. Kunci (*cipher*)

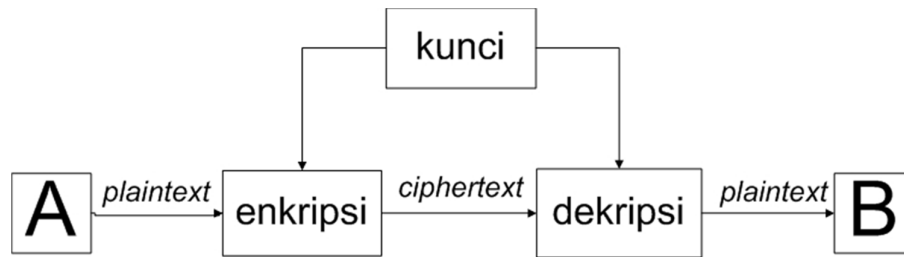
Dalam proses enkripsi dan dekripsi perlu adanya suatu kunci yang menjadi dasar pembentukan dan pengembalian data. Apabila kunci berbeda maka hasil enkripsi dan dekripsi juga akan berbeda.

2.2 Sistem Kriptografi

Berdasarkan jumlah kunci yang digunakan, ada dua jenis sistem kriptografi yaitu sistem kriptografi simetris dan sistem kriptografi asimetris.

2.2.1 Sistem Kriptografi Simetris

Pada sistem kriptografi kunci simetris, kunci untuk enkripsi sama dengan kunci untuk dekripsi. Sistem kriptografi ini mengasumsikan pengirim dan penerima pesan sudah berbagi kunci yang sama sebelum melakukan pertukaran pesan. Aplikasi sistem kriptografi simetris yang utama adalah melindungi kerahasiaan data yang disimpan pada media yang tidak aman. Model sederhana sistem kriptografi simetris dapat dilihat pada gambar 2.1.



Gambar 2.1. Sistem Kriptografi Simetris

Kelebihan dari algoritma kriptografi simetris adalah waktu proses untuk enkripsi dan dekripsi relatif cepat. Hal ini disebabkan karena efisiensi yang terjadi pada pembangkitan kunci. Sedangkan kelemahan dari algoritma ini adalah baik pengirim maupun penerima pesan harus memiliki kunci yang sama, sehingga pengirim harus mencari cara yang aman untuk memberitahukan kunci kepada penerima pesan. Selain itu, pengiriman pesan dengan pengguna yang berbeda dibutuhkan kunci yang berbeda juga. Sehingga jumlah kunci akan berbanding lurus dengan jumlah pengguna (Heriyanto, 1999).

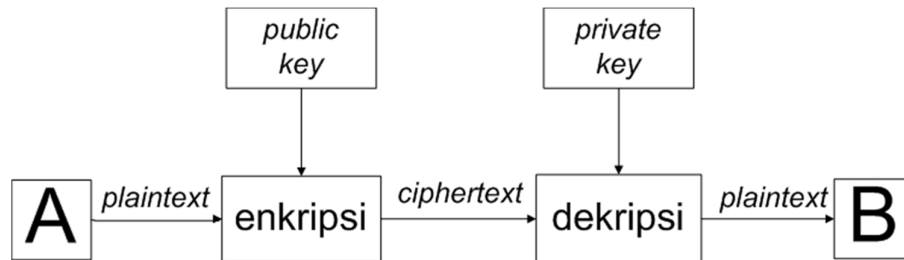
Algoritma enkripsi simetris yang populer dewasa ini adalah DES (*Data Encryption Standard*) dengan panjang kunci 56-bit, IDEA (128-bit), Twofish (sampai dengan 256-bit), Rijndael (sampai dengan 256-bit) dan lain-lain (Stalling, 1996).

2.2.2 Sistem Kriptografi Asimetris

Sistem kriptografi asimetris biasanya lebih dikenal dengan kriptografi kunci-publik (*public-key cryptography*). Ide kriptografi asimetris ini pertama kali dimunculkan oleh Whitfield Diffie dan Martin Hellman pada tahun 1976. Diffie dan Hellman mempopulerkan sistem ini tanpa menunjukkan algoritmanya.

Pada kriptografi ini, kunci untuk enkripsi tidak rahasia dan dapat diketahui oleh siapapun, sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan. Setiap orang yang berkomunikasi mempunyai sepasang kunci, yaitu kunci *private* dan kunci *public*. Pengirim mengenkripsi pesan dengan menggunakan kunci *public* penerima pesan. Hanya penerima pesan yang dapat mendekripsi pesan karena hanya dia

yang mengetahui kunci *private* miliknya sendiri. Model sederhana sistem kriptografi asimetris dapat dilihat pada gambar 2.2.



Gambar 2.2 Sistem Kriptografi Asimetris

Proses enkripsi *public-key* sederhana melibatkan empat tahap berikut (Stalling, 1996) :

- a. Setiap pengguna di dalam jaringan membuat sepasang kunci untuk digunakan sebagai kunci enkripsi dan dekripsi dari pesan yang akan diterima.
- b. Pengguna mempublikasikan kunci enkripsinya dengan menempatkan kunci publiknya ke tempat umum. Pasangan kunci yang lain tetap dijaga kerahasiaannya.
- c. Jika pengguna A ingin mengirimkan sebuah pesan ke pengguna B, dia akan mengenkripsi pesan tersebut dengan menggunakan kunci publik pengguna B.
- d. Pada saat pengguna B ingin mengirimkan sebuah pesan ke pengguna B, dia akan menggunakan kunci pribadinya sendiri. Tidak ada pihak lain yang bisa mendekripsi pesan itu karena hanya B sendiri yang mengetahui kunci pribadi B.

Sampai saat ini hanya ada beberapa sistem *cryptography* asimetris yang dipublikasikan. Yang paling berhasil sejauh ini adalah algoritma RSA yang memenuhi keenam persyaratan *public-key* di atas. Di samping itu, algoritma enkripsi *public-key* yang lain adalah LUC, DSS, Diffie-Hellman dan lain-lain (Stalling, 1996).

2.3 Dasar-dasar Matematika Kriptografi

Beberapa operasi dasar matematika yang digunakan dalam kriptografi metode WAKE adalah operasi AND, OR, XOR, Penjumlahan Modulo, dan *Shift Right*.

2.3.1 AND

Operasi AND dengan dua *input*-an akan bernilai 1 jika kedua *input*-an bernilai 1. Atau dengan kata lain *output* dari operasi AND akan memiliki nilai 0 apabila salah satu *input*-nya bernilai 0. Lambang dari operasi AND adalah tanda “ $\wedge$ ”. Aturan operasi AND dapat dinyatakan seperti tabel berikut :

Tabel 2.1 Aturan Operasi AND

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Contoh :

```

11001001
10001101
----- ^
10001001

```

2.3.2 OR

Operasi OR dengan dua *input*-an akan bernilai 0 jika kedua *input*-an bernilai 0. Atau dengan kata lain *output* dari operasi OR akan memiliki nilai 1 apabila salah satu *input*-nya bernilai 1. Lambang dari operasi OR adalah tanda “ $\vee$ ”. Aturan operasi OR dapat dinyatakan seperti tabel berikut:

Tabel 2.2 Aturan Operasi OR

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Contoh :

```

11001001
10001101
----- ∨
11001101

```

2.3.3 XOR

XOR adalah operasi *Exclusive-OR* yang dilambangkan dengan tanda “ $\oplus$ ”. Hasil dari operasi XOR akan bernilai 0 apabila kedua *input* bernilai sama. Atau dengan kata lain hasil dari operasi XOR akan bernilai 1 jika kedua *input*-an berbeda. Aturan operasi XOR dapat dirumuskan seperti tabel berikut ini :

Tabel 2.3 Aturan Operasi XOR

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Contoh :

```

11001001
10001101
----- ⊕
01000100

```

2.3.4 Penjumlahan Modulo

Penjumlahan bit modulo yang digunakan dalam metode WAKE tidak jauh berbeda dengan penjumlahan desimal. Perbedaannya hanya terdapat pada pembuangan bit apabila terdapat bit yang berlebihan. Operasi penjumlahan modulo dilakukan dengan cara menjumlahkan dua buah bit bilangan yang sama panjang dan menghasilkan bilangan dengan panjang bit yang sama pula. Jika panjang bit bilangan lebih besar, maka bit bilangan yang berlebihan tersebut akan dibuang (Stalling, 1996).

Contoh :

```

10111101
10010101
----- +
101000100

```

Bit 1 yang berlebihan di depan akan dibuang sehingga hasil yang didapatkan dari proses penjumlahan di atas adalah 01010010.

2.3.5 Pergeseran Bit (Shift)

Operasi pergeseran bit adalah operator penggeseran bit kearah yang ditentukan sebanyak nilai yang telah didefinisikan. Bit kosong yang telah tergeser akan diberikan nilai bit 0. Operasi pergeseran bit terbagi pada dua macam yaitu :

a. Opeator Geser Kanan (*Shift Right*)

Operator ini digunakan untuk melakukan pergeseran bit kearah kanan sebanyak nilai yang didefinisikan. Apabila terdapat operasi “ $x \gg 3$ ” berarti melakukan pergeseran 3 bit ke kanan dari nilai x dan mengisi tempat kosong dengan nilai bit 0.

Contoh operasi *shift right* :

```

11000110 >> 1 : 01100011
11000110 >> 2 : 00110001

```

b. Operator Geser Kiri (*Shift Left*)

Operator ini merupakan kebalikan dari operator *shift right*, yang berarti melakukan pergeseran bit kearah kiri

sebanyak nilai yang telah didefinisikan. Apabila terdapat operasi “ $x \ll 3$ ” berarti melakukan pergeseran 3 bit ke kiri dari nilai x dan mengisi tempat kosong dengan nilai bit 0. Contoh operasi *shift left* :

$11000110 \ll 1 : 10001100$

$11000110 \ll 2 : 00011000$

2.3.6 Sistem Bilangan

Sistem bilangan adalah suatu cara untuk mewakili besaran dari suatu item fisik. Sistem bilangan menggunakan suatu bilangan dasar atau basis (base / radix) yang tertentu. Dalam hubungannya dengan komputer, ada empat jenis sistem bilangan yang dikenal yaitu desimal (basis 10), biner (basis 2), oktel (basis 8) dan hexadecimal (basis 16). Bilangan – bilangan berbasis dapat diubah atau dikonversikan satu sama lain. Proses pengubahan bilangan berbasis yang akan dibahas antara lain :

- a. Konversi bilangan biner ke bilangan heksadesimal
- b. Konversi bilangan biner ke bilangan desimal
- c. Konversi bilangan desimal ke bilangan heksadesimal
- d. Konversi bilangan desimal ke bilangan biner

2.3.6.1 Konversi dari Bilangan Biner ke Bilangan Heksadesimal

Proses konversi bilangan biner ke bilangan heksadesimal dapat dilakukan dengan menggunakan suatu algoritma tertentu. Sebagai contoh, diambil bilangan biner 1100011011, maka proses konversi bilangan biner tersebut ke dalam bentuk bilangan heksadesimal menggunakan algoritma sebagai berikut :

1. Jumlah digit bilangan biner 1100011011 ada sebanyak sepuluh buah dan bukan merupakan kelipatan empat, sehingga harus ditambahkan dua buah bilangan 0 di depan bilangan biner tersebut agar jumlah digit merupakan kelipatan empat.

$1100011011 \rightarrow 001100011011$

2. Pisahkan bilangan biner tersebut ke dalam bentuk kelompok empatan.

$$001100011011 \rightarrow 0011 \mid 0001 \mid 1011$$

3. Konversi masing-masing kelompok empatan tersebut ke dalam bilangan heksa desimal dengan menggunakan tabel sistem biangan.

$$\begin{array}{ccc} 0011 & \mid & 0001 & \mid & 1011 \\ \downarrow & & \downarrow & & \downarrow \\ 3 & & 1 & & B \end{array}$$

4. Sehingga hasil konversi bilangan biner 1100011011 ke dalam bilangan heksadesimal adalah 31B.

2.3.6.2 Konversi dari Bilangan Biner ke Bilangan Desimal

Proses konversi bilangan biner ke dalam bilangan desimal dapat dilakukan dengan menjumlahkan kesetaraan desimal masing-masing posisinya. Sebagai contoh, diambil bilangan biner 1100011011. Maka proses konversi bilangan biner tersebut dapat dilakukan dengan cara seperti berikut ini :

$$\begin{aligned} 1100011011 &= 1 \times 2^9 + 1 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \\ &\quad \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 512 + 256 + 0 + 0 + 0 + 16 + 8 + 0 + 2 + 1 \\ &= 795 \end{aligned}$$

2.3.6.3 Konversi dari Bilangan Desimal ke Bilangan Heksadesimal

Untuk mengubah satu bilangan bulat desimal ke dalam bilangan heksadesimal dapat dilakukan dengan cara membagi secara berurutan bilangan-bilangan desimal dengan 16 dan konversikan angka sisanya ke dalam bentuk heksadesimal dengan urutan terbalik.

$$\begin{array}{lcl} 795 / 16 & = & 49 \text{ sisa } 11 \rightarrow B \\ 49 / 16 & = & 3 \text{ sisa } 1 \rightarrow 1 \\ 3 / 16 & = & 0 \text{ sisa } 3 \rightarrow 3 \end{array} \quad \begin{array}{c} \uparrow \\ | \\ | \end{array}$$

Sehingga hasil konversi bilangan desimal 795 ke dalam bilangan heksadesimal adalah 31B.

2.3.6.4 Konversi dari Bilangan Desimal ke Bilangan Biner

Untuk mengubah satu bilangan bulat desimal ke dalam bilangan biner dapat dilakukan dengan cara membagi secara berurutan bilangan-bilangan desimal tersebut dengan 2 dan catatlah angka sisanya dengan urutan terbalik. Sebagai contoh, diambil bilangan desimal 795. Maka konversi bilangan desimal tersebut dapat dilakukan dengan cara sebagai berikut :

$795 / 2$	$= 397$	sisa 1	$\rightarrow 1$	
$397 / 2$	$= 198$	sisa 1	$\rightarrow 1$	
$198 / 2$	$= 99$	sisa 0	$\rightarrow 0$	
$99 / 2$	$= 49$	sisa 1	$\rightarrow 1$	
$49 / 2$	$= 24$	sisa 1	$\rightarrow 1$	
$24 / 2$	$= 12$	sisa 0	$\rightarrow 0$	
$12 / 2$	$= 6$	sisa 0	$\rightarrow 0$	
$6 / 2$	$= 3$	sisa 0	$\rightarrow 0$	
$3 / 2$	$= 1$	sisa 1	$\rightarrow 1$	
$1 / 2$	$= 0$	sisa 1	$\rightarrow 1$	

Sehingga hasil konversi bilangan desimal 795 ke dalam bilangan biner adalah 1100011011.

2.4 Algoritma Kriptografi Klasik

Kriptografi mempunyai sejarah yang panjang, mulai dari kriptografi Caesar yang berkembang pada zaman sebelum masehi sampai kriptografi modern yang digunakan dalam komunikasi antar komputer pada abad ke-20. Sebelum adanya komputer, kriptografi dilakukan dengan menggunakan pensil dan kertas. Algoritma kriptografi yang dilakukan saat itu dinamakan algoritma kriptografi klasik. Algoritma kriptografi klasik berbasis karakter, yaitu enkripsi dan dekripsi dilakukan pada setiap karakter pesan. Algoritma kriptografi klasik memiliki dua teknik dasar yaitu :

a. Cipher Substitusi

Cipher substitusi adalah sebuah teknik enkripsi yang menggunakan metode pertukaran huruf dengan huruf lainnya atau dengan angka atau symbol tertentu. Setiap unit *plaintext* diganti dengan satu unit *ciphertext*. Satu unit disini berarti satu huruf, pasangan huruf, atau huruf yang dikelompokkan lebih dari dua. Algoritma substitusi tertua yang diketahui adalah Caesar Cipher yang digunakan oleh kaisar Romawi, Julius Caesar untuk menyandikan pesan yang dikirim kepada gubernurnya.

Dalam implementasinya cipher substitusi juga memerlukan kunci. Dalam hal ini kuncinya adalah jumlah pergeseran huruf. Contoh sederhana dari cipher substitusi dapat dilihat pada tabel 2.4 dengan menggunakan kunci $k = 3$.

Tabel 2.4 Cipher Substitusi

Plaintext	M	E	T	O	D	E		W	A	K	E
Ciphertext	P	H	W	R	G	H		Z	D	N	H

Sehingga dari *plaintext* “metode wake” akan terbentuk *ciphertext* “phwrgh zdnh”.

b. Cipher Transposisi

Pada cipher transposisi, huruf-huruf dari *plaintext* diatur ulang tanpa menggantinya, hanya urutannya saja yang diubah. Dengan kata lain algoritma ini melakukan transposisi terhadap rangkaian karakter di dalam teks. Nama lain dari metode ini adalah permutasi atau pengacakan karena transposisi setiap karakter di dalam teks sama dengan mempermutasikan karakter-karakter tersebut (Munir,2006).

2.5 Word Auto Key Encryption (WAKE)

WAKE merupakan singkatan dari *Word Auto Key Encryption* yang ditemukan pertama kali oleh David J. Wheeler pada tahun 1993. Tujuan dari metode ini adalah untuk mendesain sebuah sistem enkripsi dan dekripsi yang memiliki tingkat keamanan yang tinggi. Metode WAKE termasuk ke dalam salah

satu algoritma *stream cipher* yang telah digunakan secara komersial. Salah satu pemanfaatan metode ini adalah program anti virus versi terbaru yang dibuat oleh Dr. Solomon.

Metode WAKE menggunakan kunci 128 bit dan sebuah tabel 256×32 bit. Dalam algoritmanya, metode ini menggunakan operasi AND, OR, XOR, dan *Shift Right*. Inti dari metode WAKE terletak pada proses pembentukan tabel S-Box dan proses pembentukan kuncinya. Tabel S-Box dari metode WAKE bersifat fleksibel dan berbeda-beda untuk setiap putaran (Stalling, 1996).

Secara keseluruhan terdapat empat proses dalam algoritma WAKE. Proses-proses tersebut adalah :

- a. Proses pembentukan tabel S-Box
- b. Proses pembentukan kunci
- c. Proses enkripsi
- d. Proses dekripsi

2.5.1 Proses Pembentukan Tabel S-Box

Terdapat beberapa langkah dalam proses pembentukan tabel S-Box. Langkah-langkah dalam pembuatan tabel S-Box sebagai berikut (Dahria, et.al.,2012) :

1. Inisialisasi nilai TT[0] sampai TT[7] :

TT[0] : 01110010011010101000111100111011

TT[1] : 11100110100110100011101101011100

TT[2] : 1101001111000111000111111100101

TT[3] : 10101011001111000111001111010010

TT[4] : 01001101001110101000111010110011

TT[5] : 00000011100101101101011011101000

TT[6] : 00111101010011000010111101111010

TT[7] : 10011110111000100111110011110011

2. Inisialisasi nilai awal untuk T[0] sampai T[3] :

T[0] = K[0]

T[1] = K[1]

T[2] = K[2]

$$T[3] = K[3]$$

$K[0]$, $K[1]$, $K[2]$, $K[3]$ dihasilkan dari kunci yang dipecah menjadi 4 bagian yang sama panjang.

3. Untuk $T[4]$ sampai $T[255]$, lakukan proses berikut :

$$X = T[n-4] + T[n-1]$$

$$T[n] = X \gg 3 \text{ XOR } TT(X \text{ AND } 7)$$

4. Untuk $T[0]$ sampai $T[22]$, lakukan proses berikut :

$$T[n] = T[n] + T[n+89]$$

5. Atur nilai untuk beberapa variabel di bawah ini :

$$X = T[33]$$

$$Z = T[59] \text{ OR } (00000001000000000000000000000001)$$

$$Z = Z \text{ AND } (11111111011111111111111111111111)$$

$$X = (X \text{ AND } 11111111011111111111111111111111) + Z$$

6. Untuk $T[0]$ sampai $T[255]$, lakukan proses berikut :

$$X = (X \text{ AND } 11111111011111111111111111111111) + Z$$

$$T[n] = T[n] \text{ AND } 00000000111111111111111111111111 \text{ XOR } X$$

7. Inisialisasi nilai untuk beberapa variabel berikut ini :

$$T[256] = T[0]$$

$$X = X \text{ AND } 255$$

8. Untuk $T[0]$ sampai $T[255]$, lakukan proses berikut :

$$\text{Temp} = (T[n \text{ XOR } X] \text{ XOR } X) \text{ AND } 255$$

$$T[n] = T[\text{Temp}]$$

$$T[X] = T[n+1]$$

2.5.2 Proses Pembentukan Kunci

Proses pembentukan kunci pada algoritma WAKE ditentukan berdasarkan jumlah putaran yang dilakukan. Sehingga semakin banyak jumlah putaran dalam pembentukan kunci maka tingkat keamanannya akan semakin tinggi. Fungsi yang digunakan dalam proses pembentukan kunci adalah $M(X,Y) = (X+Y) \gg 8 \text{ XOR } T[(X+Y) \text{ AND } 255]$.

Proses pertama yang dilakukan untuk pembentukan kunci dalam metode WAKE adalah memecah inputan kunci menjadi empat bagian yang nantinya akan dijadikan *input*-an awal pada variabel A_0 , B_0 , C_0 , dan D_0 . Selanjutnya diproses dengan langkah berikut (Dahria, et.al.,2012):

$$A_{i+1} = M(A_i, D_i)$$

$$B_{i+1} = M(B_i, A_{i+1})$$

$$C_{i+1} = M(C_i, B_{i+1})$$

$$D_{i+1} = M(D_i, C_{i+1})$$

2.5.3 Proses Enkripsi

Proses enkripsi dari metode WAKE adalah dengan cara melakukan operasi perhitungan XOR pada *plaintext* dengan kunci untuk menghasilkan *ciphertext*. Rumus dari proses enkripsi pada metode WAKE adalah (Dahria, et.al.,2012):

$$P \oplus K = C$$

Keterangan :

P = *Plaintext*

K = Kunci

C = *Ciphertext*

2.5.4 Proses Dekripsi

Proses dekripsi dari metode WAKE adalah dengan cara melakukan operasi perhitungan XOR pada *ciphertext* dengan kunci untuk menghasilkan *plaintext*. Rumus dari proses dekripsi pada metode WAKE adalah (Dahria, et.al.,2012):

$$C \oplus K = P$$

Keterangan :

P = *Plaintext*

K = Kunci

C = *Ciphertext*

2.6 Penelitian Sebelumnya

Pada penelitian sebelumnya, yang berjudul “Perancangan Perangkat Lunak Pembelajaran Metode Kriptografi *Word Auto Key Encryption* (WAKE)”. Aplikasi ini berisi pembelajaran tentang kriptografi WAKE dan penjelasan langkah-langkah dalam melakukan enkripsi dan dekripsi. Pemrosesan yang dilakukan hanya terbatas pada masukan karakter yang dimasukkan oleh pengguna saja. Panjang kunci harus 16 karakter (Dahria, et.al.,2012).

Pada penelitian ini, penulis akan menggunakan metode *Word Auto Key Encryption* (WAKE), disini penulis akan melakukan pengembangan berupa *input*-an yang dapat langsung dimasukkan oleh pengguna dan dapat dilakukan dengan *import* file berjenis .txt. Panjang kunci akan dibatasi maksimal 16 karakter tetapi dapat kurang dari 16 karakter.

BAB III

METODOLOGI

3.1. Metodologi Penelitian

Dalam merancang dan membangun perangkat lunak kriptografi dengan metode *Word Auto Key Encryption* (WAKE). Metode pembangunan perangkat lunak menggunakan metode pengembangan perangkat lunak secara *waterfall* yang meliputi :

a. Analisa Kebutuhan

Pada tahap ini dilakukan proses pengumpulan data tentang kebutuhan pengguna akan sebuah perangkat lunak kriptografi. Sehingga perangkat lunak dapat dirancang sesuai kebutuhan pengguna.

b. Perancangan

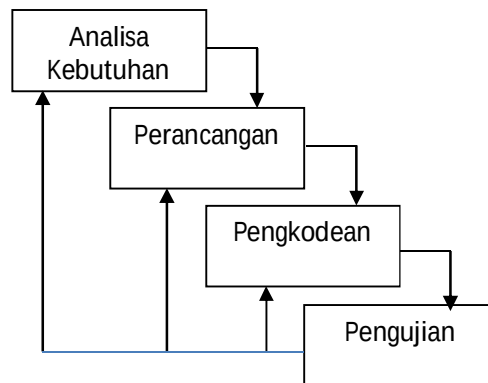
Berdasarkan semua data yang telah dikumpulkan pada tahap sebelumnya, maka pada tahap ini perangkat lunak didesain sedemikian rupa agar memenuhi kebutuhan secara menyeluruh.

c. Pengkodean

Dari desain yang telah dibuat sebelumnya, maka pada tahap ini dilakukan penerjemahan data atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman tertentu.

d. Pengujian

Merupakan tahap pengujian terhadap perangkat lunak yang dibangun.



Gambar 3.1 Metode Waterfall

3.2. Analisa Kebutuhan Perangkat Lunak

3.2.1 Analisa Kebutuhan Input

Dalam perangkat lunak kriptografi WAKE terdapat beberapa kebutuhan input sebagai berikut :

- a. Kunci
- b. *Plaintext*

3.2.2 Analisa Kebutuhan Proses

Dalam perangkat lunak kriptografi WAKE terdapat beberapa proses yang berlangsung yaitu sebagai berikut :

- a. Proses S-box
- b. Proses pembentukan kunci
- c. Proses enkripsi
- d. Proses dekripsi

3.2.3 Analisa Kebutuhan Output

Dalam perangkat lunak kriptografi WAKE terdapat beberapa output yang dapat dihasilkan oleh perangkat lunak diantaranya sebagai berikut :

- a. Pembentukan proses S-box
- b. Pembentukan kunci
- c. Hasil enkripsi
- d. Hasil dekripsi

3.2.4 Analisa Kebutuhan Perangkat Lunak

Dalam proses pembuatan perangkat lunak kriptografi WAKE dibutuhkan beberapa aplikasi pendukung diantaranya sebagai berikut :

- a. Sistem Operasi Windows 7 (32bit)
- b. Visual Basic 6.0
- c. Codejock Xtreme Suite Pro ActiveX v12.1.1
- d. Notepad

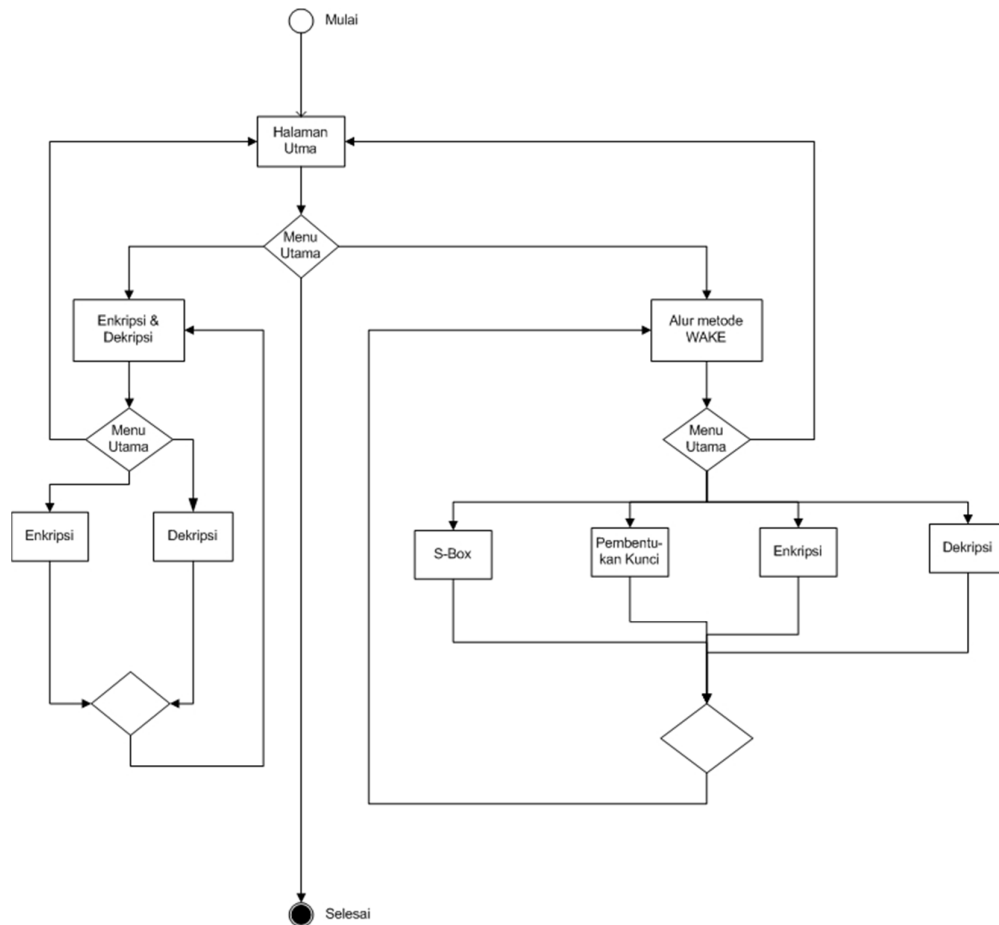
3.2.5 Analisa Kebutuhan Perangkat Keras

Dalam pembuatan perangkat lunak kriptografi WAKE dibutuhkan dukungan beberapa perangkat keras diantaranya sebagai berikut :

- a. VGA NVIDIA GEFORCE with CUDA 512 Mb
- b. RAM 4 GB
- c. Prosessor Intel Core 2 Duo
- d. Hardisk Seagate 512 Gb
- e. Keyboard dan Mouse

3.3. Perancangan Perangkat Lunak

Perangkat lunak kriptografi WAKE memiliki beberapa fasilitas yang dapat memudahkan pengguna untuk melakukan proses enkripsi dan dekripsi data. Untuk lebih jelasnya dapat dilihat pada *activity diagram* perangkat lunak kriptografi WAKE sebagai berikut :



Gambar 3.2 Activity Diagram

Pada saat perangkat lunak pertama kali dijalankan akan muncul menu utama. Dalam menu utama terdapat tiga pilihan yaitu Enkripsi & Dekripsi yang berfungsi untuk melakukan proses enkripsi dan dekripsi data, alur metode WAKE yang berfungsi untuk mengetahui alur enkripsi dan dekripsi data pada setiap prosesnya, dan keluar yang berfungsi untuk menghentikan jalannya perangkat lunak.

Pada menu enkripsi & dekripsi terdapat tiga pilihan menu yaitu proses enkripsi yang berfungsi untuk memulai proses enkripsi data, proses dekripsi yang berfungsi untuk memulai proses dekripsi data, dan menu utama yang berfungsi untuk kembali ke menu utama. Pada data proses enkripsi dan dekripsi dibuka, maka pengguna akan diminta untuk memasukkan teks yang akan dienkripsi atau

didekripsi, kunci, dan putaran kunci. Setelah hasil dari proses enkripsi dan dekripsi terbentuk, hasil tersebut dapat disimpan dalam bentuk file .txt.

Pada menu alur metode wake terdapat beberapa pilihan yaitu :

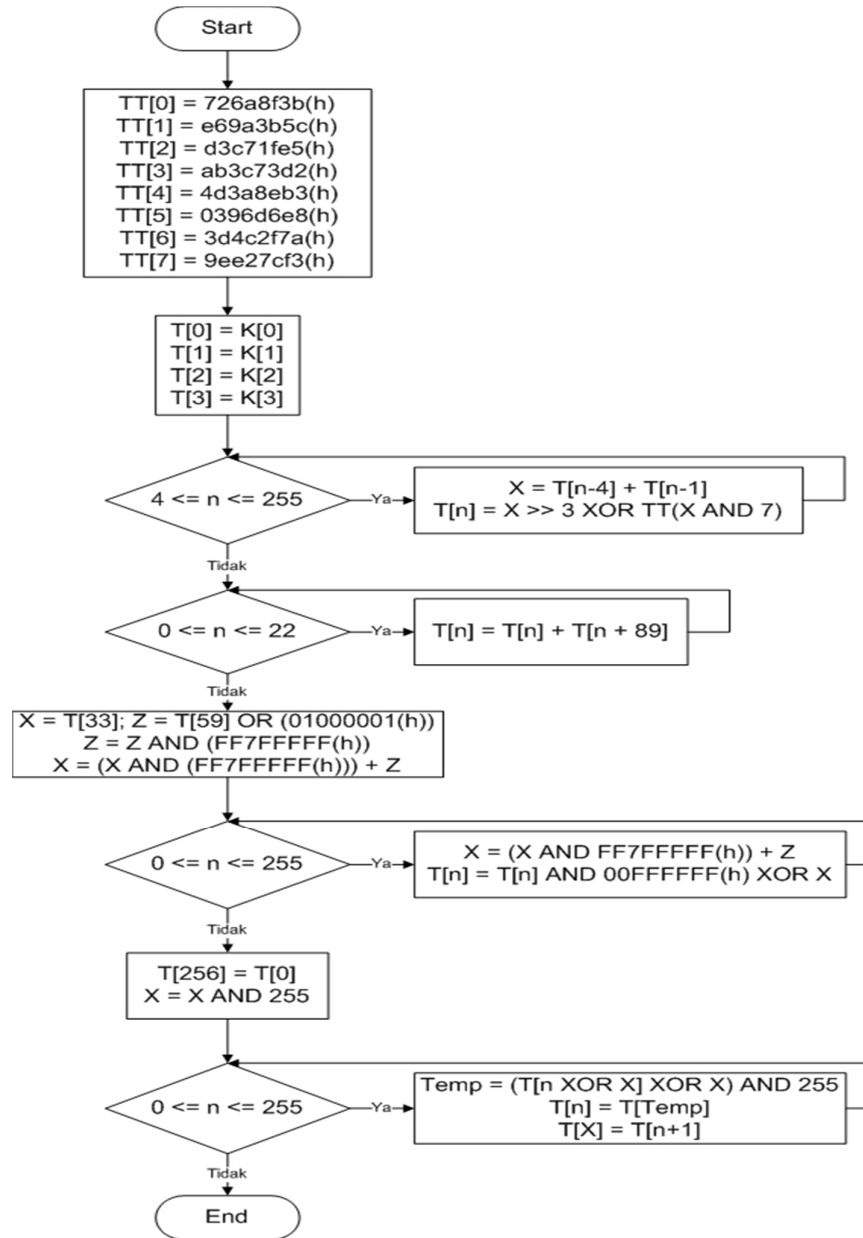
- a. S-Box, yang berfungsi untuk memulai proses pembentukan S-Box. Pada proses ini pengguna akan diminta memasukkan kunci untuk membentuk tabel S-Box.
- b. Pembentukan Kunci, yang berfungsi untuk memulai proses pembentukan kunci. Pada proses ini pengguna akan diminta memasukkan kunci dan putaran kunci yang digunakan untuk membentuk kunci.
- c. Proses Enkripsi, yang berfungsi untuk memulai proses enkripsi data. Pengguna akan diminta memasukkan teks, kunci, dan putaran kunci. Dalam proses ini pengguna juga dapat melihat hasil tabel S-Box dan Kunci yang terbentuk dari kunci yang telah dimasukkan.
- d. Keluar, yang berfungsi untuk keluar dari menu alur metode WAKE dan kembali ke menu utama.

Proses kriptografi metode *Word Auto Key Encryption* (WAKE) terdiri dari empat proses, yaitu :

- a. Proses Pembentukan Tabel S-Box
- b. Proses Pembentukan Kunci
- c. Proses Enkripsi
- d. Proses Dekripsi

3.3.1 Proses Pembentukan Tabel S-Box

Proses pembentukan tabel S-Box terdiri dari delapan proses utama. Dalam prosesnya, pembentukan tabel S-Box memerlukan *input* berupa kunci dengan panjang 128 bit biner atau 16 karakter *ascii*. Untuk mengetahui lebih jelas proses-proses pembentukan tabel S-Box dapat dilihat pada gambar 3.3.



Gambar 3.3 Flowchart S-Box

Pertama kali saat menjalankan proses pembentukan table S-Box maka akan dilakukan inisialisasi nilai $T[0]$ sampai dengan $T[7]$. Setelah itu kunci yang kita masukkan akan diubah ke dalam biner dan dipecah menjadi empat bagian yaitu $K[0]$ sampai $K[3]$. Setelah itu kita inisialisasikan $T[0] = K[0]$, $T[1] = K[1]$, $T[2] = K[2]$, dan $T[3] = K[3]$. Setelah itu untuk $4 \leq n \leq 255$ lakukan proses $X = T[n-4] + T[n-1]$ dan $T[n]$

yang nantinya akan dijadikan inputan awal pada variabel A_0 , B_0 , C_0 , dan D_0 . Selanjutnya diproses dengan langkah berikut :

$$A_{i+1} = M(A_i, D_i)$$

$$B_{i+1} = M(B_i, A_{i+1})$$

$$C_{i+1} = M(C_i, B_{i+1})$$

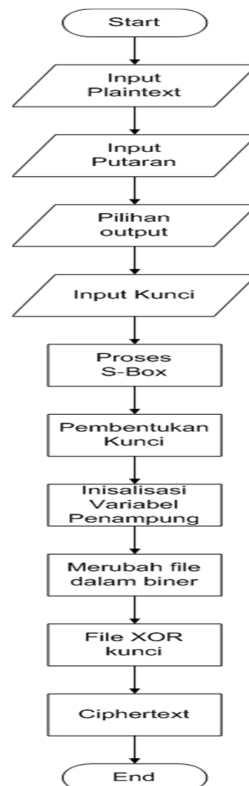
$$D_{i+1} = M(D_i, C_{i+1})$$

Nilai dari D_i merupakan nilai dari kunci K_i .

3.3.3 Proses Enkripsi

Proses enkripsi menggunakan metode WAKE untuk menghasilkan *ciphertext* adalah hasil dari operasi XOR dari *plaintext* dan 32 bit kunci yang dihasilkan dari proses pembentukan kunci atau dapat dirumuskan sebagai berikut :

$$\text{Ciphertext (C)} = \text{Plaintext (P)} \text{ XOR Key (K)}$$



Gambar 3.5 Flowchart Enkripsi

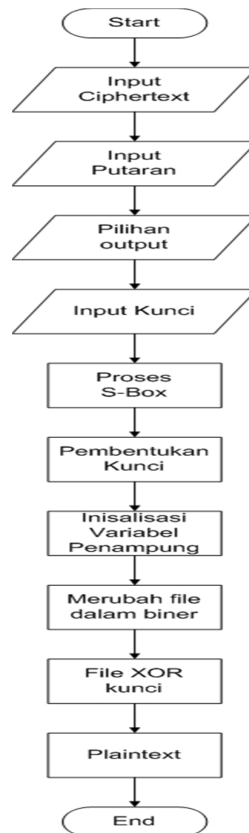
Pada saat proses enkripsi dijalankan, pengguna akan diminta untuk menentukan *plaintext*, jumlah putaran, hasil *output*, dan kunci. Setelah

semua telah diisi maka kunci akan diolah dalam proses pembentukan tabel S-Box. Setelah proses S-Box selesai, maka hasil dari proses S-Box akan digunakan untuk proses pembentukan kunci. Sehingga terbentuklah kunci yang akan digunakan untuk melakukan proses enkripsi. *Plaintext* yang telah dimasukkan tadi akan dikenakan operasi XOR dengan kunci yang telah terbentuk dari proses sebelumnya, sehingga terbentuklah *ciphertext*.

3.3.4 Proses Dekripsi

Proses dekripsi menggunakan metode WAKE untuk menghasilkan *plaintext* adalah hasil dari operasi XOR dari *ciphertext* dan 32 bit kunci yang dihasilkan dari proses pembentukan kunci atau dapat dirumuskan sebagai berikut :

$$Plaintext (P) = Ciphertext (C) \text{ XOR Key } (K)$$



Gambar 3.6 Flowchart Dekripsi

Pada saat proses dekripsi dijalankan, pengguna akan diminta untuk menentukan *ciphertext*, jumlah putaran, hasil *output*, dan kunci. Setelah

semua telah diisi maka kunci akan diolah dalam proses pembentukan tabel S-Box. Setelah proses S-Box selesai, maka hasil dari proses S-Box akan digunakan untuk proses pembentukan kunci. Sehingga terbentuklah kunci yang akan digunakan untuk melakukan proses dekripsi. *Ciphertext* yang telah dimasukkan tadi akan di-XOR-kan dengan kunci yang telah terbentuk dari proses sebelumnya, sehingga terbentuklah *plaintext*.

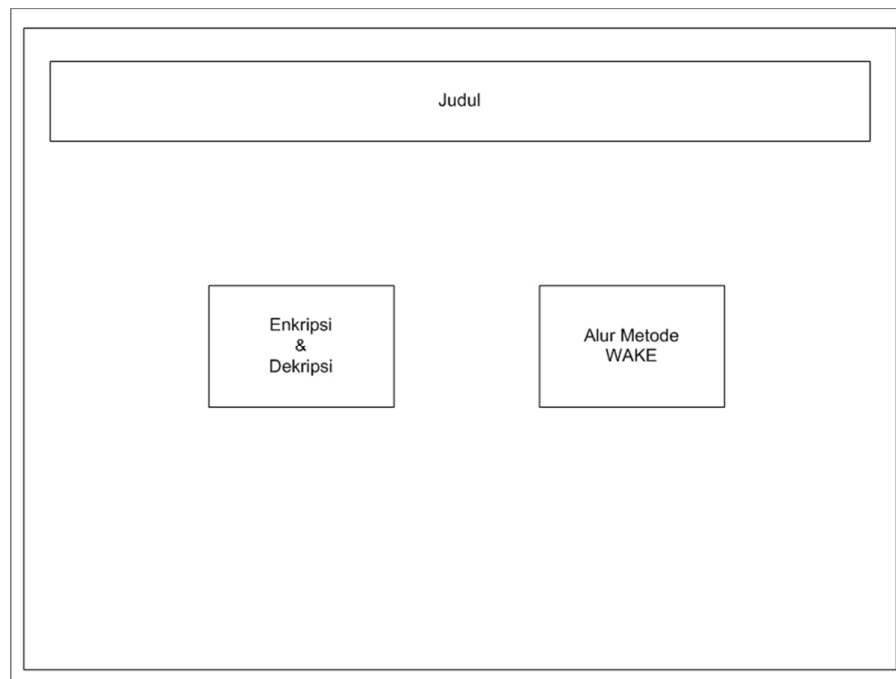
3.4. Perancangan Antarmuka

Perancangan antarmuka perangkat lunak kriptografi WAKE dirancang dengan dasar *user friendly* agar user dapat dengan mudah memahami dan mengoperasikan perangkat lunak tersebut. Perancangan antarmuka perangkat lunak ini menggunakan bahasa pemrograman Microsoft Visual Basic 6.0. Perangkat lunak kriptografi WAKE memiliki beberapa *form*, antara lain :

- a. *Form Main*
- b. *Form Menu* Enkripsi dan Dekripsi
- c. *Form Proses* Enkripsi
- d. *Form Proses* Dekripsi
- e. *Form Menu* Alur Metode WAKE
- f. *Form Proses* S-Box
- g. *Form Proses* Pembentukan Kunci
- h. *Form Input* Proses Enkripsi
- i. *Form Hasil* Tabel S-Box
- j. *Form Hasil* Pembentukan Kunci
- k. *Form Hasil* Enkripsi
- l. *Form Input* Proses Dekripsi
- m. *Form Hasil* Dekripsi

3.4.1 *Form Main*

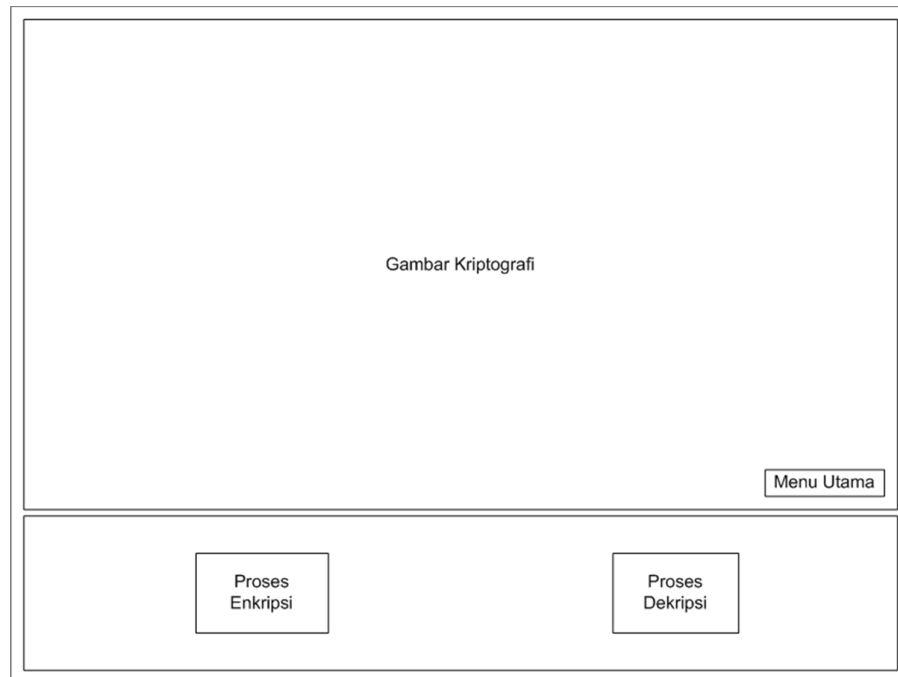
Form Main adalah *form* utama dari perangkat lunak kriptografi WAKE. Saat pertama kali program dibuka maka *form* ini yang akan muncul. Perancangan antarmuka *form main* dapat dilihat pada gambar 3.7.



Gambar 3.7 Perancangan *Form Main*

3.4.2 *Form Menu Enkripsi dan Dekripsi*

Form menu enkripsi dan dekripsi adalah halaman yang muncul saat pengguna melakukan pemanggilan *form* menu enkripsi dan dekripsi dengan menggunakan *tool* enkripsi & dekripsi yang ada pada *form main*. Perancangan antarmuka *form* menu enkripsi dan dekripsi dapat dilihat pada gambar 3.8.



Gambar 3.8 Perancangan *Form* Menu Enkripsi dan Dekripsi

3.4.3 *Form* Proses Enkripsi

Form proses enkripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* proses enkripsi dengan menggunakan *tool* proses enkripsi yang ada pada *form* menu enkripsi dan dekripsi. Pada *form* proses enkripsi, pengguna akan diminta untuk memasukkan *plaintext* dan kunci yang digunakan untuk enkripsi data. Perancangan antarmuka *form* proses enkripsi dapat dilihat pada gambar 3.9.

The diagram illustrates the layout of the 'Form Proses Enkripsi'. It features a main title 'Proses Enkripsi' at the top. Below the title, there are two input fields: 'Input Plaintext' and 'Cari file .txt'. These are followed by a section containing 'Input Kunci' and 'Pilihan Jumlah Putaran'. A 'Proses' button is positioned below these fields. At the bottom of the form, there is a 'Hasil Enkripsi' field, a 'Simpan (.txt)' button, and a 'Keluar' button.

Gambar 3.9 Perancangan *Form* Proses Enkripsi

3.4.4 *Form* Proses Dekripsi

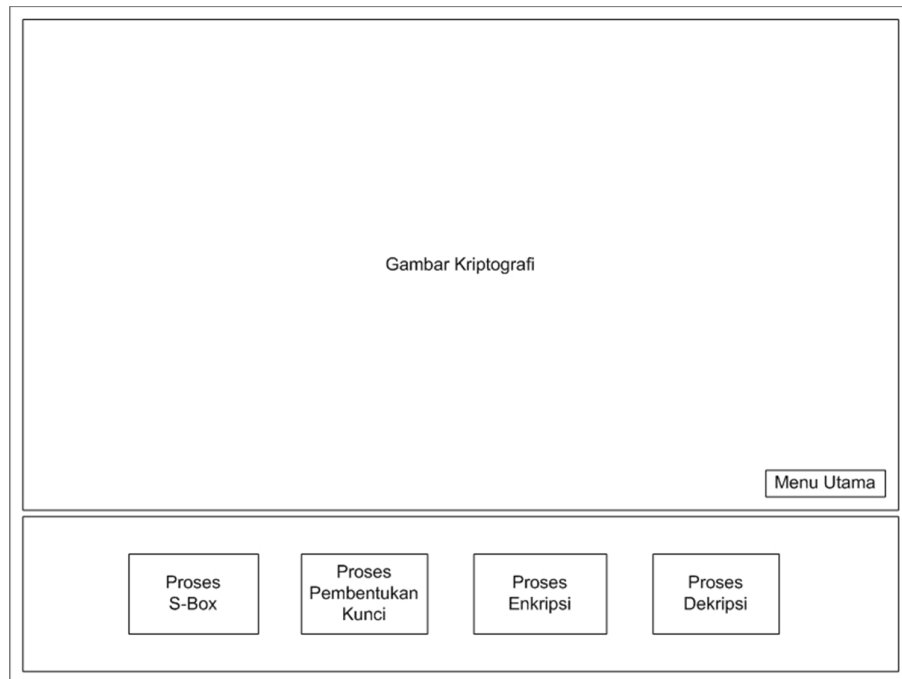
Form proses dekripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* proses dekripsi dengan menggunakan *tool* proses dekripsi yang ada pada *form* menu enkripsi dan dekripsi. Pada *form* proses dekripsi, pengguna akan diminta untuk memasukkan *ciphertext* dan kunci yang digunakan untuk dekripsi data. Perancangan antarmuka *form* proses dekripsi dapat dilihat pada gambar 3.10.

The diagram illustrates the layout of the 'Form Proses Dekripsi' (Decryption Process Form). It is a rectangular window with a title bar at the top containing the text 'Proses Dekripsi'. Below the title bar, the form is organized into several sections. The first section contains two input fields: 'Input Ciphertext' on the left and 'Cari file .txt' on the right. Below these, there is a larger container holding two more input fields: 'Input Kunci' and 'Pilihan Jumlah Putaran'. A 'Proses' button is positioned centrally below these input fields. The bottom section of the form contains a large 'Hasil Dekripsi' field on the left, a 'Simpan (.txt)' button on the right, and a 'Keluar' button at the very bottom right corner.

Gambar 3.10 Perancangan *Form* Proses Dekripsi

3.4.5 *Form* Menu Alur Metode WAKE

Form menu alur metode WAKE adalah halaman yang muncul saat pengguna melakukan pemanggilan *form* menu alur metode WAKE dengan menggunakan *tool* alur metode WAKE yang ada pada *form* main. Perancangan antarmuka *form* menu alur metode WAKE dapat dilihat pada gambar 3.11.



Gambar 3.11 Perancangan *Form* Menu Alur Metode WAKE

3.4.6 *Form* Proses S-Box

Form proses S-Box adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* proses S-Box dengan menggunakan *tool* proses S-Box yang ada pada *form* menu alur metode WAKE. Pada *form* proses S-Box, pengguna akan diminta untuk memasukkan kunci yang akan diproses. Perancangan antarmuka *form* proses S-Box dapat dilihat pada gambar 3.12.

The wireframe for the 'Proses S-Box' form is structured as follows:

- Title Bar:** A rectangular box at the top containing the text 'Proses S-Box'.
- Input Field:** A rectangular box below the title bar containing the text 'Input Kunci'.
- Action Button:** A rectangular box to the right of the input field containing the text 'Proses Kerja S-Box'.
- Output Area:** A large rectangular box below the input field containing the text 'Hasil Pembentukan Tabel S-Box'.
- Exit Button:** A small rectangular box at the bottom right containing the text 'Keluar'.

Gambar 3.12 Perancangan *Form* Proses S-Box

3.4.7 *Form* Proses Pembentukan Kunci

Form proses pembentukan kunci adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* proses pembentukan kunci dengan menggunakan *tool* proses pembentukan kunci yang ada pada *form* menu alur metode WAKE. Pada *form* proses pembentukan kunci, pengguna akan diminta untuk memasukkan kunci yang akan diproses. Perancangan antarmuka *form* proses pembentukan kunci dapat dilihat pada gambar 3.13.

The wireframe for the 'Proses Pembentukan Kunci' form is structured as follows:

- Title Bar:** A rectangular box at the top containing the text 'Proses Pembentukan Kunci'.
- Input Fields:** Two rectangular boxes stacked vertically below the title bar, containing the text 'Input Kunci' and 'Pilihan Jumlah Putaran' respectively.
- Action Button:** A rectangular box to the right of the input fields containing the text 'Proses'.
- Output Area:** A large rectangular box below the input fields containing the text 'Hasil Pembentukan Kunci'.
- Exit Button:** A small rectangular box at the bottom right containing the text 'Keluar'.

Gambar 3.13 Perancangan *Form* Proses Pembentukan Kunci

3.4.8 *Form Input Proses Enkripsi*

Form input proses enkripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form input* proses enkripsi dengan menggunakan *tool* proses enkripsi yang ada pada *form* menu alur metode WAKE. Pada *form input* proses enkripsi, pengguna akan diminta untuk memasukkan *plaintext* dan kunci yang digunakan untuk enkripsi data. Perancangan antarmuka *form* proses enkripsi dapat dilihat pada gambar 3.14.

```

graph TD
    Title[Proses Enkripsi]
    Plaintext[Input Plaintext]
    CariFile[Cari file .txt]
    KunciField[Input Kunci]
    SBox[S-Box]
    PilihanPutaran[Pilihan Jumlah Putaran]
    KunciLabel[Kunci]
    Proses[Proses]
    Keluar[Keluar]

    Title --- Plaintext
    Plaintext --- CariFile
    CariFile --- SBox
    SBox --- KunciField
    KunciField --- PilihanPutaran
    PilihanPutaran --- KunciLabel
    KunciLabel --- Keluar
    Keluar --- Proses
    
```

Gambar 3.14 Perancangan *Form Input* Proses Enkripsi

3.4.9 *Form Hasil Tabel S-Box*

Form hasil tabel S-Box adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* hasil tabel S-Box dengan menggunakan *tool* S-Box yang ada pada *form input* proses enkripsi dan *form input* proses dekripsi. Perancangan antarmuka *form* hasil tabel S-Box dapat dilihat pada gambar 3.15.

Gambar 3.15 Perancangan *Form* Hasil Tabel S-Box

3.4.10 *Form* Hasil Pembentukan Kunci

Form hasil pembentukan kunci adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* hasil pembentukan kunci dengan menggunakan *tool* kunci yang ada pada *form input* proses enkripsi dan *form input* proses dekripsi. Perancangan antarmuka *form* hasil pembentukan kunci dapat dilihat pada gambar 3.16.

Gambar 3.16 Perancangan *Form* Hasil Pembentukan Kunci

3.4.11 *Form* Hasil Enkripsi

Form hasil enkripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* hasil enkripsi dengan menggunakan *tool* proses yang ada pada *form input* proses enkripsi.

Perancangan antarmuka *form* hasil enkripsi dapat dilihat pada gambar 3.17.

The wireframe for the encryption result form (Gambar 3.17) consists of the following elements:

- A title box labeled "Judul" at the top.
- A group box containing three input fields: "Plaintext", "Kunci", and "Putaran".
- A group box at the bottom containing a large output field labeled "Hasil Enkripsi" and a button labeled "Simpan (.txt)".
- A "Keluar" (Exit) button located at the bottom right of the form.

Gambar 3.17 Perancangan *Form* Hasil Enkripsi

3.4.12 *Form Input* Proses Dekripsi

Form input proses dekripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form input* proses dekripsi dengan menggunakan *tool* proses dekripsi yang ada pada *form* menu alur metode WAKE. Pada *form input* proses dekripsi, pengguna akan diminta untuk memasukkan *ciphertext* dan kunci yang digunakan untuk dekripsi data. Perancangan antarmuka *form input* proses dekripsi dapat dilihat pada gambar 3.18.

The wireframe for the decryption input form (Gambar 3.18) consists of the following elements:

- A title box labeled "Proses Dekripsi" at the top.
- A group box on the left containing three input fields: "Input Ciphertext", "Input Kunci", and "Pilihan Jumlah Putaran".
- A column of buttons on the right: "Cari file .txt", "S-Box", "Kunci", and "Keluar".
- A "Proses" button located at the bottom center of the form.

Gambar 3.18 Perancangan *Form Input* Proses Dekripsi

3.4.13 *Form Hasil Dekripsi*

Form hasil dekripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* hasil dekripsi dengan menggunakan *tool* proses yang ada pada *form input* proses dekripsi. Perancangan antarmuka *form* hasil dekripsi dapat dilihat pada gambar 3.19.

The diagram illustrates the layout of the 'Form Hasil Dekripsi' (Decryption Result Form). It is enclosed in a rectangular frame. At the top, there is a wide rectangular box labeled 'Judul'. Below this, a large container box holds three stacked input fields labeled 'Ciphertext', 'Kunci', and 'Putaran'. At the bottom of the frame, there is a horizontal row containing a wide box labeled 'Hasil Dekripsi' and a button labeled 'Simpan (.txt)'. Below the 'Simpan (.txt)' button, centered at the bottom of the frame, is another button labeled 'Keluar'.

Gambar 3.19 Perancangan *Form Hasil Dekripsi*

BAB IV

HASIL DAN PEMBAHASAN

4.1. Implementasi

Tahap implementasi merupakan tahap penerapan dari perancangan sistem ke dalam kondisi sebenarnya, sehingga dapat diketahui bahwa sistem yang dibangun telah sesuai dengan perencanaan. Pada implementasi sistem ini menggunakan bahasa pemrograman Visual Basic 6.0. Berikut ini merupakan implementasi dari perangkat lunak kriptografi dengan metode *Word Auto Key Encryption* (WAKE).

4.1.1. Implementasi *Form Main*

Form Main adalah *form* utama dari perangkat lunak kriptografi WAKE. Saat pertama kali program dibuka maka *form* ini yang akan muncul. Implementasi antarmuka *form main* dapat dilihat pada gambar 4.1.



Gambar 4.1 *Form Main*

Pada *form main* terdapat dua *tools* yaitu Enkripsi & Dekripsi yang digunakan untuk menampilkan *form* menu enkripsi dan dekripsi, dan alur metode WAKE yang digunakan untuk menampilkan *form* menu alur metode WAKE.

4.1.2. Implementasi *Form* Menu Enkripsi dan Dekripsi

Form menu enkripsi dan dekripsi adalah halaman yang muncul saat pengguna melakukan pemanggilan *form* menu enkripsi dan dekripsi dengan menggunakan *tool* enkripsi & dekripsi yang ada pada *form main*. Perancangan antarmuka *form* menu enkripsi dan dekripsi dapat dilihat pada gambar 4.2.



Gambar 4.2 *Form* Menu Enkripsi dan Dekripsi

Pada *form* menu enkripsi dan dekripsi terdapat beberapa *tools* yang memiliki fungsi masing-masing. Beberapa *tools* yang ada pada *form* menu enkripsi dan dekripsi adalah sebagai berikut :

- a. Proses enkripsi data, digunakan untuk menampilkan *form* proses enkripsi.

- b. Proses dekripsi data, digunakan untuk menampilkan *form* proses dekripsi.
- c. Menu Utama, digunakan untuk kembali ke *form main*.

4.1.3. Implementasi *Form* Proses Enkripsi

Form proses enkripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* proses enkripsi dengan menggunakan *tool* proses enkripsi yang ada pada *form* menu enkripsi dan dekripsi. Pada *form* proses enkripsi, pengguna akan diminta untuk memasukkan *plaintext* dan kunci yang digunakan untuk enkripsi data. Perancangan antarmuka *form* proses enkripsi dapat dilihat pada gambar 4.3.

Gambar 4.3 *Form* Proses Enkripsi

Pada *form* proses enkripsi terdapat beberapa *tools* yang memiliki fungsi masing-masing. Beberapa *tools* yang ada pada *form* proses enkripsi adalah sebagai berikut :

- Cari file .txt, digunakan untuk memasukkan *plaintext* berupa file .txt.
- Proses, digunakan untuk memulai proses enkripsi.
- Simpan, digunakan untuk menyimpan hasil enkripsi dalam bentuk file .txt.
- Keluar, digunakan untuk keluar dari *form* proses enkripsi.

4.1.4. Implementasi *Form* Proses Dekripsi

Form proses dekripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* proses dekripsi dengan menggunakan *tool* proses dekripsi yang ada pada *form* menu enkripsi dan dekripsi. Pada *form* proses dekripsi, pengguna akan diminta untuk memasukkan *ciphertext* dan kunci yang digunakan untuk dekripsi data. Perancangan antarmuka *form* proses dekripsi dapat dilihat pada gambar 4.4.

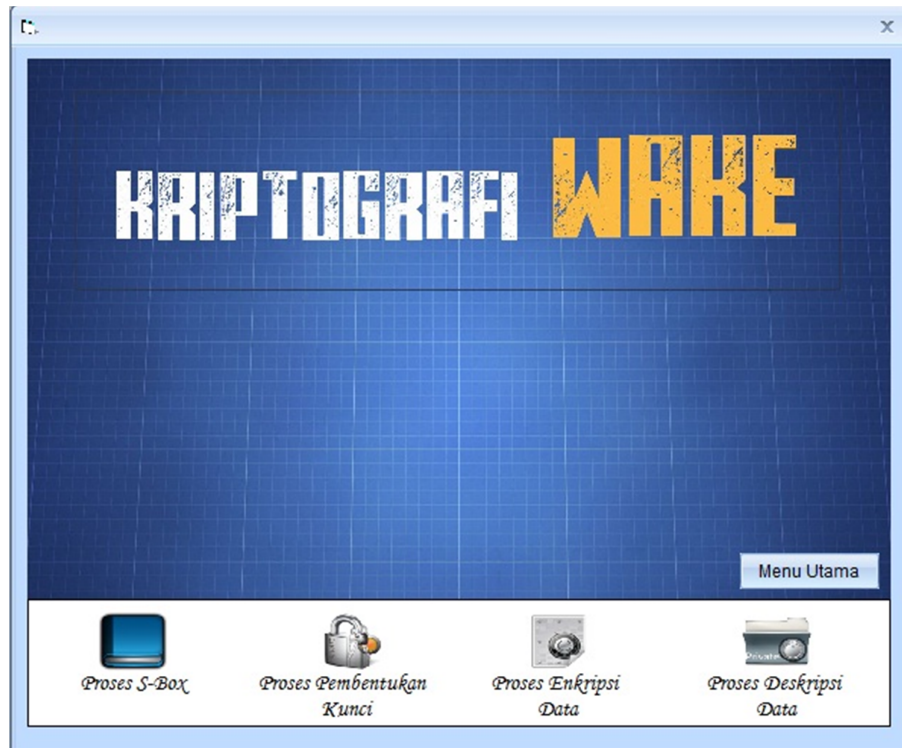
Gambar 4.4 *Form* Proses Dekripsi

Pada *form* proses dekripsi terdapat beberapa *tools* yang memiliki fungsi masing-masing. Beberapa *tools* yang ada pada *form* proses dekripsi adalah sebagai berikut :

- Cari file .txt, digunakan untuk memasukkan *ciphertext* berupa file .txt.
- Proses, digunakan untuk memulai proses dekripsi.
- Simpan, digunakan untuk menyimpan hasil dekripsi dalam bentuk file .txt.
- Keluar, digunakan untuk keluar dari *form* proses dekripsi.

4.1.5. Implementasi *Form* Menu Alur Metode WAKE

Form menu alur metode WAKE adalah halaman yang muncul saat pengguna melakukan pemanggilan *form* menu alur metode WAKE dengan menggunakan *tool* alur metode WAKE yang ada pada *form* main. Perancangan antarmuka *form* menu alur metode WAKE dapat dilihat pada gambar 4.5.



Gambar 4.5 *Form* Menu Alur Metode WAKE

Pada *form* menu alur metode WAKE terdapat beberapa *tools* yang memiliki fungsi masing-masing. Beberapa *tools* yang ada pada *form* menu alur metode WAKE adalah sebagai berikut :

- a. Proses S-Box, digunakan untuk menampilkan *form* proses S-Box.
- b. Proses Pembentukan Kunci, digunakan untuk menampilkan *form* proses pembentukan kunci.
- c. Proses Enkripsi, digunakan untuk menampilkan *form input* proses enkripsi.
- d. Proses Dekripsi, digunakan untuk menampilkan *form input* proses dekripsi.
- e. Menu Utama, digunakan untuk kembali ke *form main*.

4.1.6. Implementasi *Form* Proses S-Box

Form proses S-Box adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* proses S-Box dengan menggunakan *tool* proses S-Box yang ada pada *form* menu alur metode WAKE. Pada *form* proses S-Box, pengguna akan diminta untuk memasukkan kunci yang akan diproses. Perancangan antarmuka *form* proses S-Box dapat dilihat pada gambar 4.6.

Gambar 4.6 *Form* Proses S-Box

Pada *form* proses S-Box terdapat dua *tools* yaitu proses kerja S-Box yang digunakan untuk memulai proses pembentukan tabel S-Box, dan keluar yang digunakan untuk keluar dari *form* proses S-Box.

4.1.7. Implementasi *Form* Proses Pembentukan Kunci

Form proses pembentukan kunci adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* proses pembentukan kunci dengan menggunakan *tool* proses pembentukan kunci yang ada pada *form* menu alur metode WAKE. Pada *form* proses pembentukan kunci, pengguna akan diminta untuk memasukkan kunci yang akan diproses. Perancangan antarmuka *form* proses pembentukan kunci dapat dilihat pada gambar 4.7.

Gambar 4.7 *Form Proses Pembentukan Kunci*

Pada *form* proses pembentukan kunci terdapat dua *tools* yaitu proses yang digunakan untuk memulai proses pembentukan kunci, dan keluar yang digunakan untuk keluar dari *form* proses pembentukan kunci.

4.1.8. Implementasi Form Input Proses Enkripsi

Form input proses enkripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form input* proses enkripsi dengan menggunakan *tool* proses enkripsi yang ada pada *form* menu alur metode WAKE. Pada *form input* proses enkripsi, pengguna akan diminta untuk memasukkan *plaintext* dan kunci yang digunakan untuk enkripsi data. Perancangan antarmuka *form* proses enkripsi dapat dilihat pada gambar 4.8.

Gambar 4.8 *Form Input Proses Enkripsi*

Pada *form input* proses enkripsi terdapat beberapa *tools* yang memiliki fungsi masing-masing. Beberapa *tools* yang ada pada *form* menu alur metode WAKE adalah sebagai berikut :

- a. Cari File .txt, digunakan untuk memasukkan *plaintext* berupa file .txt.
- b. S-Box, digunakan untuk menampilkan hasil proses S-Box dari kunci yang dimasukkan.
- c. Kunci, digunakan untuk menampilkan hasil pembentukan kunci dari kunci yang dimasukkan.
- d. Proses, digunakan untuk menampilkan *form* Hasil Enkripsi.
- e. Keluar, digunakan untuk keluar dari *form input* proses enkripsi.

4.1.9. Implementasi Form Hasil Tabel S-Box

Form hasil tabel S-Box adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* hasil tabel S-Box dengan menggunakan *tool* S-Box yang ada pada *form input* proses enkripsi dan *form input* proses dekripsi. Perancangan antarmuka *form* hasil tabel S-Box dapat dilihat pada gambar 4.9.

Tabel S-Box dari kunci wake :

S-BOX / T[N]	HASIL
T[0]	00011110000000011000111010010110
T[1]	01010001100101100111101110001110
T[2]	11111011111010011000100001111001
T[3]	00111100010010011110000101111101
T[4]	01010001100101100111101110001110
T[5]	11010000100001100101110001011101
T[6]	11000010101010010101101100111101
T[7]	1110100001111011101110110011111
T[8]	10101010110010100001110010100100
T[9]	11111010101001100101010000010111
T[10]	00010000100011100000011101111001
T[11]	01110010010110011010011000000001
T[12]	11010010000111000110111000111101
T[13]	00010100101100000000101001011011
T[14]	11001010111010111110000100111001
T[15]	00000010001110101110011110110001
T[16]	00011111100001001010111100000100
T[17]	11010010000111000110111000111101
T[18]	01110010010110011010011000000001

Keluar

Gambar 4.9 *Form* Hasil Tabel S-Box

Pada *form* hasil tabel S-Box terdapat satu *tool* yaitu *tool* keluar yang digunakan untuk keluar dari halaman *form* hasil tabel S-Box dan kembali ke halaman sebelumnya.

4.1.10. Implementasi *Form* Hasil Pembentukan Kunci

Form hasil pembentukan kunci adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* hasil pembentukan kunci dengan menggunakan *tool* kunci yang ada pada *form input* proses enkripsi dan *form input* proses dekripsi. Perancangan antarmuka *form* hasil pembentukan kunci dapat dilihat pada gambar 4.10.

Gambar 4.10 *Form Hasil Pembentukan Kunci*

Pada *form* hasil pembentukan kunci terdapat satu *tool* yaitu *tool* keluar yang digunakan untuk keluar dari halaman *form* hasil pembentukan kunci dan kembali ke halaman sebelumnya.

4.1.11. Implementasi *Form Hasil Enkripsi*

Form hasil enkripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* hasil enkripsi dengan menggunakan *tool* proses yang ada pada *form input* proses enkripsi. Perancangan antarmuka *form* hasil enkripsi dapat dilihat pada gambar 4.11.

Gambar 4.11 *Form Hasil Enkripsi*

Pada *form* hasil enkripsi terdapat dua *tools* yaitu simpan .txt yang digunakan untuk menyimpan hasil enkripsi ke dalam file .txt, dan keluar

yang digunakan untuk keluar dari halaman *form* hasil enkripsi dan kembali ke halaman sebelumnya.

4.1.12. Implementasi *Form Input* Proses Dekripsi

Form input proses dekripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form input* proses dekripsi dengan menggunakan *tool* proses dekripsi yang ada pada *form* menu alur metode WAKE. Pada *form input* proses dekripsi, pengguna akan diminta untuk memasukkan *ciphertext* dan kunci yang digunakan untuk dekripsi data. Perancangan antarmuka *form input* proses dekripsi dapat dilihat pada gambar 4.12.

Gambar 4.12 *Form Input* Proses Dekripsi

Pada *form input* proses dekripsi terdapat beberapa *tools* yang memiliki fungsi masing-masing. Beberapa *tools* yang ada pada *form input* proses dekripsi adalah sebagai berikut :

- Cari File .txt, digunakan untuk memasukkan *ciphertext* berupa file .txt.
- Lihat Hasil Proses S-Box, digunakan untuk menampilkan tabel S-Box yang terbentuk dari kunci yang dimasukkan.
- Lihat Hasil Pembentukan Kunci, digunakan untuk menampilkan hasil kunci yang terbentuk.

- d. Proses, digunakan untuk menampilkan *form* hasil dekripsi.
- e. Keluar, digunakan untuk keluar dari halaman *form input* proses dekripsi dan kembali ke halaman sebelumnya.

4.1.13. Implementasi *Form Hasil Dekripsi*

Form hasil dekripsi adalah halaman yang muncul pada saat pengguna melakukan pemanggilan *form* hasil dekripsi dengan menggunakan *tool* proses yang ada pada *form input* proses dekripsi. Perancangan antarmuka *form* hasil dekripsi dapat dilihat pada gambar 4.12.

Gambar 4.13 *Form Hasil Dekripsi*

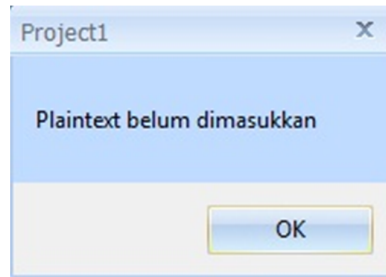
Pada *form* hasil dekripsi terdapat dua *tools* yaitu simpan .txt yang digunakan untuk menyimpan hasil dekripsi ke dalam file .txt, dan keluar yang digunakan untuk keluar dari halaman *form* hasil dekripsi dan kembali ke halaman sebelumnya.

4.1.14. Implementasi Penanganan Kesalahan

Pada perangkat lunak kriptografi WAKE terdapat dua jenis penanganan kesalahan yaitu penanganan kesalahan *input* teks dan penanganan kesalahan *input* kunci.

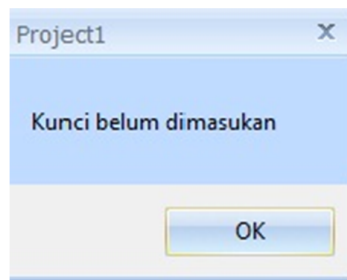
Penanganan kesalahan *input plaintext* adalah halaman yang akan muncul apabila pengguna melakukan proses enkripsi atau dekripsi tanpa

memasukkan teks atau dengan kata lain kolom teks kosong. Implementasi Halaman penanganan kesalahan *input* teks dapat dilihat pada gambar 4.14.



Gambar 4.14 Penanganan Kesalahan *Input* Teks

Penanganan kesalahan *input kunci* adalah halaman yang muncul apabila pengguna menjalankan salah satu proses utama kriptografi WAKE tanpa memasukkan kunci terlebih dahulu atau dengan kata lain kolom kunci kosong. Implementasi halaman penanganan kesalahan *input kunci* dapat dilihat pada gambar 4.15.



Gambar 4.15 Penanganan Kesalahan *Input* Kunci

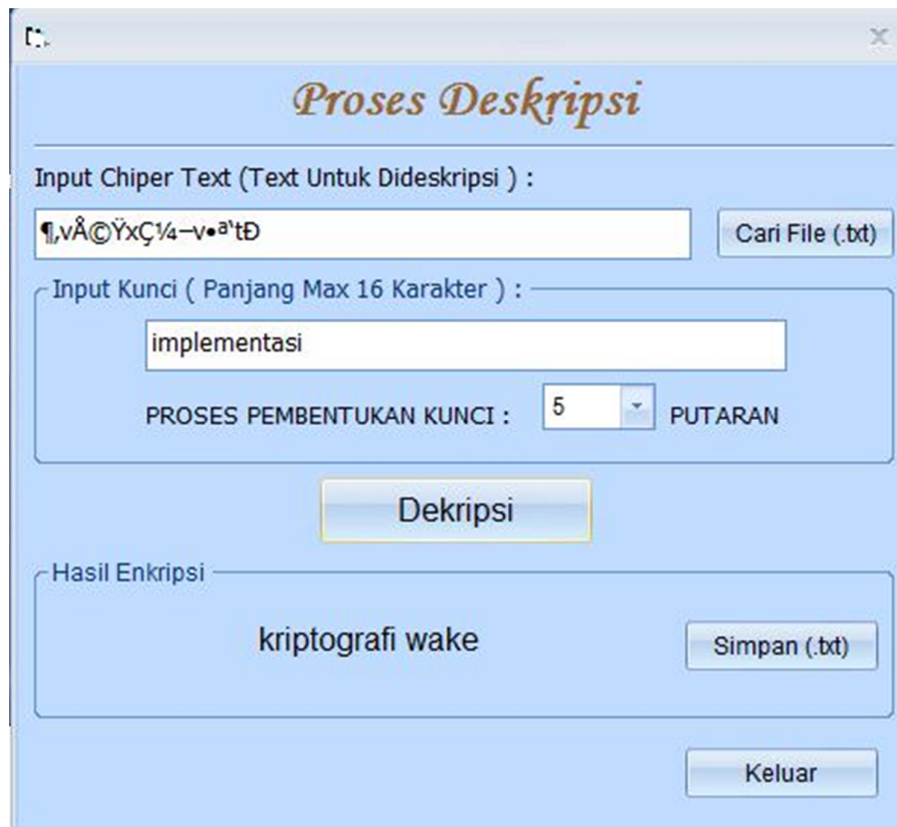
4.2. Pengujian Perangkat Lunak

Pengujian sistem akan dilakukan dengan cara melakukan proses enkripsi dan dekripsi terhadap kata atau kalimat dengan panjang maksimal 1000 karakter. Pengujian sistem ini bertujuan untuk melihat kemampuan sistem dalam melakukan enkripsi dan dekripsi terhadap suatu kata atau kalimat dengan menggunakan kunci tertentu.

Terdapat sebuah kunci yang dimasukkan ke dalam kolom kunci yaitu "implementasi". Pada kolom *plaintext* akan diisi "kriptografi wake". Jumlah putaran yang digunakan adalah lima kali putaran.

Gambar 4.16 Pengujian Enkripsi

Pada gambar 4.16 terdapat hasil pengujian proses enkripsi yang dilakukan terhadap perangkat lunak kriptografi wake. Setelah dilakukan pengujian terhadap proses enkripsi, maka akan muncul hasil enkripsi seperti yang ditunjukkan pada gambar 4.16. Hasil enkripsi akan berubah menjadi *plaintext* seperti semula apabila dilakukan proses dekripsi terhadap hasil enkripsi tersebut dengan kunci dan jumlah putaran yang sama.



Gambar 4.17 Pengujian Dekripsi

Dari gambar 4.17 diketahui bahwa perangkat lunak kriptografi WAKE dapat melakukan proses dekripsi dengan benar, yaitu mengubah hasil enkripsi menjadi *plaintext* yang sama seperti semula. Untuk mengetahui apakah proses enkripsi dan dekripsi yang dilakukan oleh perangkat lunak telah sesuai dengan metode WAKE, maka akan dilakukan proses perhitungan manual seperti yang ditunjukkan oleh tabel 4.1.

Tabel 4.1 Pengujian Enkripsi Manual

=====	
PROSES PEMBENTUKAN TABEL S-BOX	
=====	
LANGKAH 1 - Inisialisasi nilai TT[0] ... TT[7]	

TT[0] =	01110010011010101000111100111011
TT[1] =	11100110100110100011101101011100
TT[2] =	11010011110001110001111111100101
TT[3] =	10101011001111000111001111010010
TT[4] =	01001101001110101000111010110011
TT[5] =	00000011100101101101011011101000
TT[6] =	00111101010011000010111101111010

```

TT[7] = 10011110111000100111110011110011
=====
LANGKAH 2 - Pecah kunci menjadi 4 kelompok besar dan masukkan
pada T[0] ... T[3]
-----
Kunci : 'implementasi0000'
Kode ascii dari : i = 105 = 01101001
Kode ascii dari : m = 109 = 01101101
Kode ascii dari : p = 112 = 01110000
Kode ascii dari : l = 108 = 01101100
Kode ascii dari : e = 101 = 01100101
Kode ascii dari : m = 109 = 01101101
Kode ascii dari : e = 101 = 01100101
Kode ascii dari : n = 110 = 01101110
Kode ascii dari : t = 116 = 01110100
Kode ascii dari : a = 97 = 01100001
Kode ascii dari : s = 115 = 01110011
Kode ascii dari : i = 105 = 01101001
Kode ascii dari : 0 = 48 = 00110000
Kode ascii dari : 0 = 48 = 00110000
Kode ascii dari : 0 = 48 = 00110000
Kode ascii dari : 0 = 48 = 00110000
Kunci (Dalam Biner) :
0110100101101101011100000011011000110010101101101100101011011100
111010001100001011100110110100100110000001100000011000000110000
T[0] = K[0] = 011010010110110101110000001101100
T[1] = K[1] = 01100101011011010110010101101110
T[2] = K[2] = 01110100011000010111001101101001
T[3] = K[3] = 00110000001100000011000000110000
=====
LANGKAH 3 - Untuk n = 4 sampai 255, lakukan prosedur berikut.
-----
n = 4
-> X = T[0] + T[3] = 011010010110110101110000001101100 +
00110000001100000011000000110000 =
10011001100111011010000010011100
-> X >> 3 (Shift Right 3 bit) = 10011001100111011010000010011100
>> 3 = 00010011001100111011010000010011
-> X AND 7 = 10011001100111011010000010011100 AND 7 = 4
-> T[4] = X >> 3 XOR TT[X AND 7] =
00010011001100111011010000010011 XOR TT[4] =
01011110000010010011101010100000

n = 5
-> X = T[1] + T[4] = 01100101011011010110010101101110 +
01011110000010010011101010100000 =
11000011011101101010000000001110
-> X >> 3 (Shift Right 3 bit) = 11000011011101101010000000001110
>> 3 = 00011000011011101101010000000001
-> X AND 7 = 11000011011101101010000000001110 AND 7 = 6
-> T[5] = X >> 3 XOR TT[X AND 7] =
00011000011011101101010000000001 XOR TT[6] =
00100101001000101111101101111011

n = 6

```

```
-> X = T[2] + T[5] = 01110100011000010111001101101001 +
00100101001000101111101101111011 =
10011001100001000110111011100100
-> X >> 3 (Shift Right 3 bit) = 10011001100001000110111011100100
>> 3 = 00010011001100001000110111011100
-> X AND 7 = 10011001100001000110111011100100 AND 7 = 4
-> T[6] = X >> 3 XOR TT[X AND 7] =
00010011001100001000110111011100 XOR TT[4] =
01011110000010100000001101101111
```

```
n = 7
-> X = T[3] + T[6] = 00110000001100000011000000110000 +
01011110000010100000001101101111 =
10001110001110100011001110011111
-> X >> 3 (Shift Right 3 bit) = 10001110001110100011001110011111
>> 3 = 00010001110001110100011001110011
-> X AND 7 = 10001110001110100011001110011111 AND 7 = 7
-> T[7] = X >> 3 XOR TT[X AND 7] =
00010001110001110100011001110011 XOR TT[7] =
10001111001001010011101010000000
```

```
n = 8
-> X = T[4] + T[7] = 01011110000010010011101010100000 +
10001111001001010011101010000000 =
11101101001011100111010100100000
-> X >> 3 (Shift Right 3 bit) = 11101101001011100111010100100000
>> 3 = 00011101101001011100111010100100
-> X AND 7 = 11101101001011100111010100100000 AND 7 = 0
-> T[8] = X >> 3 XOR TT[X AND 7] =
00011101101001011100111010100100 XOR TT[0] =
01101111110011110100000110011111
```

```
n = 9
-> X = T[5] + T[8] = 00100101001000101111101101111011 +
01101111110011110100000110011111 =
10010100111100100011110100011010
-> X >> 3 (Shift Right 3 bit) = 10010100111100100011110100011010
>> 3 = 00010010100111100100011110100011
-> X AND 7 = 10010100111100100011110100011010 AND 7 = 2
-> T[9] = X >> 3 XOR TT[X AND 7] =
000100101001111100100011110100011 XOR TT[2] =
11000001010110010101100001000110
```

```
n = 10
-> X = T[6] + T[9] = 01011110000010100000001101101111 +
11000001010110010101100001000110 =
00011111011000110101101110110101
-> X >> 3 (Shift Right 3 bit) = 00011111011000110101101110110101
>> 3 = 00000011111011000110101101110110
-> X AND 7 = 00011111011000110101101110110101 AND 7 = 5
-> T[10] = X >> 3 XOR TT[X AND 7] =
00000011111011000110101101110110 XOR TT[5] =
00000000011110101011110110011110
```

Dan seterusnya sampai n = 255

```

n = 255
-> X = T[251] + T[254] = 10011001010110001001000011011000 +
1010000001101011111111010110001 =
00111001110001001000111110001001
-> X >> 3 (Shift Right 3 bit) = 00111001110001001000111110001001
>> 3 = 00000111001110001001000111110001
-> X AND 7 = 00111001110001001000111110001001 AND 7 = 1
-> T[255] = X >> 3 XOR TT[X AND 7] =
00000111001110001001000111110001 XOR TT[1] =
11100001101000101010101010101101

=====
LANGKAH 4 - Untuk n = 0 sampai 22, lakukan prosedur berikut.
-----
N = 0
T[0] = T[0] + T[89] = 01101001011011010111000001101100 +
10001010001110100000000110110101 =
11110011101001110111001000100001
N = 1
T[1] = T[1] + T[90] = 01100101011011010110010101101110 +
10100011000011100100001011110111 =
00001000011110111010100001100101
N = 2
T[2] = T[2] + T[91] = 01110100011000010111001101101001 +
00010000000001000001000101101101 =
10000100011001011000010011010110
N = 3
T[3] = T[3] + T[92] = 00110000001100000011000000110000 +
11010110001101011110001011011100 =
00000110011001100001001100001100
N = 4
T[4] = T[4] + T[93] = 01011110000010010011101010100000 +
11101010100101111100011111001110 =
01001000101000010000001001101110
N = 5
T[5] = T[5] + T[94] = 00100101001000101111101101111011 +
00010010001000100001011110110000 =
00110111010001010001001100101011
N = 6
T[6] = T[6] + T[95] = 01011110000010100000001101101111 +
00000111110100100001001111001011 =
01100101110111000001011100111010
N = 7
T[7] = T[7] + T[96] = 10001111001001010011101010000000 +
10000101001000101000001000100111 =
00010100010001111011110010100111
N = 8
T[8] = T[8] + T[97] = 01101111110011110100000110011111 +
00001110011000011001111111010110 =
01111110001100001110000101110101
N = 9
T[9] = T[9] + T[98] = 11000001010110010101100001000110 +
00111001010111000101100110001010 =
11111010101101011011000111010000
N = 10
T[10] = T[10] + T[99] = 00000000011110101011110110011110 +

```

```
00001011101100110001101101000010 =
00001100001011011101100011100000
```

Dan seterusnya sampai $n = 22$

$N = 22$

```
T[22] = T[22] + T[111] = 00110011001001010100010100011011 +
01000101011101100101001011010011 =
01111000100110111001011111101110
```

```
=====
LANGKAH 5 - Set nilai untuk variabel di bawah ini.
-----
```

```
X = 0100011111110000000001011000101
Z = T[59] OR 00000001000000000000000000000001 =
00010001011100001101100011111000 OR
0000000100000000000000000000000001 =
00010001011100001101100011111001
Z = Z AND 11111111011111111111111111111111 =
00010001011100001101100011111001 AND
1111111101111111111111111111111111 =
00010001011100001101100011111001
X = (X AND 11111111011111111111111111111111) + Z =
01000111111100000000001011000101 AND
1111111101111111111111111111111111 =
01000110111000000000001011000101 +
00010001011100001101100011111001 =
01011000111000001101101110111110
```

```
=====
LANGKAH 6 - Untuk  $n = 0$  sampai 255, lakukan prosedur berikut.
-----
```

```
X = (01011000111000001101101110111110 AND
1111111101111111111111111111111111) +
00010001011100001101100011111001 =
0110100110100011011010010110111
T[0] = 11110011101001110111001000100001 AND
0000000011111111111111111111111111 XOR
01101001110100011011010010110111 =
0110100101101101100011010010110
X = (01101001110100011011010010110111 AND
1111111101111111111111111111111111) +
00010001011100001101100011111001 =
01111010110000101000110110110000
T[1] = 00001000011110111010100001100101 AND
0000000011111111111111111111111111 XOR
01111010110000101000110110110000 =
011110101110010010010010111010101
X = (01111010110000101000110110110000 AND
1111111101111111111111111111111111) +
00010001011100001101100011111001 =
10001011101100110110011010101001
T[2] = 10000100011001011000010011010110 AND
0000000011111111111111111111111111 XOR
10001011101100110110011010101001 =
100010111101011011100010011111111
```

```

X = (10001011101100110110011010101001 AND
11111111011111111111111111111111) +
00010001011100001101100011111001 =
1001110010100100001111110100010
T[3] = 000001100110011000100001001100001100 AND
00000000111111111111111111111111 XOR
1001110010100100001111110100010 =
10011100110000100010110010101110
X = (1001110010100100001111110100010 AND
11111111011111111111111111111111) +
00010001011100001101100011111001 =
10101101100101010001100010011011
T[4] = 01001000101000010000001001101110 AND
00000000111111111111111111111111 XOR
10101101100101010001100010011011 =
1010110100110100000110101110101
X = (10101101100101010001100010011011 AND
11111111011111111111111111111111) +
00010001011100001101100011111001 =
10111110100001011111000110010100
T[5] = 00110111010001010001001100101011 AND
00000000111111111111111111111111 XOR
10111110100001011111000110010100 =
1011111011000000111000101011111
X = (10111110100001011111000110010100 AND
11111111011111111111111111111111) +
00010001011100001101100011111001 =
11001111011101101100101010001101
T[6] = 01100101110111000001011100111010 AND
00000000111111111111111111111111 XOR
11001111011101101100101010001101 =
110011111010101011011011011011011
X = (11001111011101101100101010001101 AND
11111111011111111111111111111111) +
00010001011100001101100011111001 =
11100000111001111010001110000110
T[7] = 00010100010001111011110010100111 AND
00000000111111111111111111111111 XOR
11100000111001111010001110000110 =
11100000101000000001111100100001
X = (11100000111001111010001110000110 AND
11111111011111111111111111111111) +
00010001011100001101100011111001 =
11110001110110000111110001111111
T[8] = 01111110001100001110000101110101 AND
00000000111111111111111111111111 XOR
11110001110110000111110001111111 =
11110001111010001001110100001010
X = (11110001110110000111110001111111 AND
11111111011111111111111111111111) +
00010001011100001101100011111001 =
00000010110010010101010101111000
T[9] = 11111010101101011011000111010000 AND
00000000111111111111111111111111 XOR
00000010110010010101010101111000 =
00000010011111001110010010101000

```

```

X = (00000010110010010101010101111000 AND
11111110111111111111111111111111) +
00010001011100001101100011111001 =
00010011101110100010111001110001
T[10] = 00001100001011011101100011100000 AND
00000000111111111111111111111111 XOR
00010011101110100010111001110001 =
00010011100101111111011010010001

```

Dan seterusnya sampai T[255]

```

X = (01000111110010001111101111000101 AND
11111110111111111111111111111111) +
00010001011100001101100011111001 =
01011000101110011101010010111110
T[255] = 111000011010001010101010101101 AND
00000000111111111111111111111111 XOR
01011000101110011101010010111110 =
0101100000011011011111000010011

```

=====

LANGKAH 7 - Set nilai untuk variabel berikut.

```

T[256] = T[0] = 01101001011101101100011010010110
X = X AND 255 = 01011000101110011101010010111110 AND
0000000000000000000000000000000011111111 =
0000000000000000000000000000000010111110

```

=====

LANGKAH 8 - Untuk n = 0 sampai 255, lakukan prosedur berikut.

```

N = 0
Temp = T[190] XOR X AND 255 = 00000111100001111010010110011100 XOR
0000000000000000000000000000000010111110 =
00000111100001111010010100100010 AND 255 =
00000000000000000000000000000000100010
T[0] = T[34] = 10101011111000001001010110101100
T[190] = T[1] = 01111010101110010010010111010101
N = 1
Temp = T[191] XOR X AND 255 = 00011000010000000011110001100111 XOR
0000000000000000000000000000000010111110 =
00011000010000000011110011011001 AND 255 =
0000000000000000000000000000000011011001
T[1] = T[217] = 11010010101110010011011100111011
T[190] = T[2] = 10001011110101101110001001111111
N = 2
Temp = T[188] XOR X AND 255 = 11100101111110110101100010001010 XOR
0000000000000000000000000000000010111110 =
11100101111110110101100000110100 AND 255 =
00000000000000000000000000000000110100
T[2] = T[52] = 11011101100000110101000001011011
T[190] = T[3] = 10011100110000100010110010101110
N = 3
Temp = T[189] XOR X AND 255 = 11110110100101100101110011110101 XOR
0000000000000000000000000000000010111110 =
11110110100101100101110001001011 AND 255 =

```



```
T[0] = 10101011111000001001010110101100
T[1] = 11010010101110010011011100111011
T[2] = 11011101100000110101000001011011
T[3] = 01100100100111010100000000001011
T[4] = 00111111000000111101000110011110
T[5] = 00101111010111011010001101101011
T[6] = 10111111011111010010010010111110
T[7] = 00010001111100101001110101101011
T[8] = 11110101001100111000010010100010
T[9] = 00110000001011001000100001100000
T[10] = 10011001010111000000101001111100
```

T[256] = 01101001011101101100011010010110

Kunci 'implementasi0000 diubah dalam bentuk biner =
01101001011011010111000001101100011001010110110101100101011011100
111010001100001011100110110100100110000001100000011000000110000

A(0) = 01101001011011010111000001101100
B(0) = 01100101011011010110010101101110
C(0) = 01110100011000010111001101101001
D(0) = 00110000001100000011000000110000

```
(01101001011011010111000001101100 ,
00110000001100000011000000110000) =
(01101001011011010111000001101100 +
00110000001100000011000000110000) >> 8 XOR T
[(01101001011011010111000001101100 +
00110000001100000011000000110000) AND 255(10)] =
10011001100111011010000010011100 >> 8 XOR T[156]
00000000100110011001110110100000 XOR
10011110001110000000011100011010 =
10011110101000011001101011101010
A[1] = 10011110101000011001101010111010
```

```
FungsiM (B[0], A[1]) = Fungsi M
```

```
(01100101011011010110010101101110 ,
10011110101000011001101010111010) =
(01100101011011010110010101101110 +
10011110101000011001101010111010) >> 8 XOR T
[(01100101011011010110010101101110 +
10011110101000011001101010111010) AND 255(10)] =
00000100000011110000000000101000 >> 8 XOR T[40]
00000000000001000000111100000000 XOR
11101000000010001100011001000010 =
11101000000011001100100101000010
B[1] = 11101000000011001100100101000010
```

```
FungsiM (C[0], B[1]) = Fungsi M
```

```
(01110100011000010111001101101001 ,
11101000000011001100100101000010) =
(01110100011000010111001101101001 +
11101000000011001100100101000010) >> 8 XOR T
[(01110100011000010111001101101001 +
11101000000011001100100101000010) AND 255(10)] =
01011100011011100011110010101011 >> 8 XOR T[171]
00000000010111000110111000111100 XOR
00111101111010111000100111000110 =
00111101101101111110011111111010
C[1] = 00111101101101111110011111111010
```

```
FungsiM (D[0], C[1]) = Fungsi M
```

```
(00110000001100000011000000110000 ,
00111101101101111110011111111010) =
(00110000001100000011000000110000 +
00111101101101111110011111111010) >> 8 XOR T
[(00110000001100000011000000110000 +
00111101101101111110011111111010) AND 255(10)] =
01101101111010000001100000101010 >> 8 XOR T[42]
00000000011011011110100000011000 XOR
01100001011111101100101010110111 =
01100001000100110010001010101111
D[1] = 01100001000100110010001010101111
```

```
-----
Putaran ke 2
-----
```

```
FungsiM (A[1], D[1]) = Fungsi M
```

```
(10011110101000011001101010111010 ,
01100001000100110010001010101111) =
(10011110101000011001101010111010 +
01100001000100110010001010101111) >> 8 XOR T
[(10011110101000011001101010111010 +
01100001000100110010001010101111) AND 255(10)] =
11111111101101001011110101101001 >> 8 XOR T[105]
```

```

0000000011111111011010010111101 XOR
00111000100011001101100110110101 =
00111000011100110110110100001000
A[2] = 00111000011100110110110100001000

FungsiM (B[1], A[2]) = Fungsi M

(11101000000011001100100101000010 ,
00111000011100110110110100001000) =
(11101000000011001100100101000010 +
00111000011100110110110100001000) >> 8 XOR T
[(11101000000011001100100101000010 +
00111000011100110110110100001000) AND 255(10)] =
00100000100000000011011001001010 >> 8 XOR T[74]
00000000001000001000000000110110 XOR
01001000010100001001010100001010 =
01001000011100000001010100111100
B[2] = 01001000011100000001010100111100

FungsiM (C[1], B[2]) = Fungsi M

(00111101101101111111001111111010 ,
01001000011100000001010100111100) =
(00111101101101111111001111111010 +
01001000011100000001010100111100) >> 8 XOR T
[(00111101101101111111001111111010 +
01001000011100000001010100111100) AND 255(10)] =
10000110001001111111110100110110 >> 8 XOR T[54]
00000000100001100010011111111101 XOR
00010000010010101011011000001100 =
00010000110011001001000111110001
C[2] = 00010000110011001001000111110001

FungsiM (D[1], C[2]) = Fungsi M

(01100001000100110010001010101111 ,
00010000110011001001000111110001) =
(01100001000100110010001010101111 +
00010000110011001001000111110001) >> 8 XOR T
[(01100001000100110010001010101111 +
00010000110011001001000111110001) AND 255(10)] =
01110001110111111011010010100000 >> 8 XOR T[160]
00000000011100011101111110110100 XOR
01100001011111101100101010110111 =
01100001000011110001010100000011
D[2] = 01100001000011110001010100000011

-----
Putaran ke 3
-----

FungsiM (A[2], D[2]) = Fungsi M

(00111000011100110110110100001000 ,
01100001000011110001010100000011) =
(00111000011100110110110100001000 +

```

```

01100001000011110001010100000011) >> 8 XOR T
[(00111000011100110110110100001000 +
01100001000011110001010100000011) AND 255(10)] =
10011001100000101000001000001011 >> 8 XOR T[11]
00000000100110011000001010000010 XOR
01010100101000110110111011100010 =
01010100001110101110110001100000
A[3] = 01010100001110101110110001100000

```

FungsiM (B[2], A[3]) = Fungsi M

```

(01001000011100000001010100111100 ,
01010100001110101110110001100000) =
(01001000011100000001010100111100 +
01010100001110101110110001100000) >> 8 XOR T
[(01001000011100000001010100111100 +
01010100001110101110110001100000) AND 255(10)] =
10011100101010110000000110011100 >> 8 XOR T[156]
00000000100111001010101100000001 XOR
10011110001110000000011100011010 =
10011110101001001010110000011011
B[3] = 10011110101001001010110000011011

```

FungsiM (C[2], B[3]) = Fungsi M

```

(00010000110011001001000111110001 ,
10011110101001001010110000011011) =
(00010000110011001001000111110001 +
10011110101001001010110000011011) >> 8 XOR T
[(00010000110011001001000111110001 +
10011110101001001010110000011011) AND 255(10)] =
10101111011100010011111000001100 >> 8 XOR T[12]
00000000101011110111000100111110 XOR
00100001101001010001101110100011 =
00100001000010100110101010011101
C[3] = 00100001000010100110101010011101

```

FungsiM (D[2], C[3]) = Fungsi M

```

(01100001000011110001010100000011 ,
00100001000010100110101010011101) =
(01100001000011110001010100000011 +
00100001000010100110101010011101) >> 8 XOR T
[(01100001000011110001010100000011 +
00100001000010100110101010011101) AND 255(10)] =
1000001000011001011111110100000 >> 8 XOR T[160]
00000000100000100001100101111111 XOR
01100001011111101100101010110111 =
01100001111111001101001111001000
D[3] = 01100001111111001101001111001000

```

Putaran ke 4

FungsiM (A[3], D[3]) = Fungsi M

```
(01010100001110101110110001100000 ,
01100001111111001101001111001000) =
(01010100001110101110110001100000 +
01100001111111001101001111001000) >> 8 XOR T
[(01010100001110101110110001100000 +
01100001111111001101001111001000) AND 255(10)] =
10110110001101111100000000101000 >> 8 XOR T[40]
0000000010110110001101111000000 XOR
11101000000010001100011001000010 =
11101000101111101111000110000010
A[4] = 11101000101111101111000110000010
```

FungsiM (B[3], A[4]) = Fungsi M

```
(10011110101001001010110000011011 ,
11101000101111101111000110000010) =
(10011110101001001010110000011011 +
11101000101111101111000110000010) >> 8 XOR T
[(10011110101001001010110000011011 +
11101000101111101111000110000010) AND 255(10)] =
10000111011000111001110110011101 >> 8 XOR T[157]
00000000100001110110001110011101 XOR
11110101001100111000010010100010 =
11110101101101001110011100111111
B[4] = 11110101101101001110011100111111
```

FungsiM (C[3], B[4]) = Fungsi M

```
(00100001000010100110101010011101 ,
11110101101101001110011100111111) =
(00100001000010100110101010011101 +
11110101101101001110011100111111) >> 8 XOR T
[(00100001000010100110101010011101 +
11110101101101001110011100111111) AND 255(10)] =
00010110101111110101000111011100 >> 8 XOR T[220]
000000000000101101011111101010001 XOR
01111011100111011010101100011101 =
01111011100010110001010001001100
C[4] = 01111011100010110001010001001100
```

FungsiM (D[3], C[4]) = Fungsi M

```
(01100001111111001101001111001000 ,
01111011100010110001010001001100) =
(01100001111111001101001111001000 +
01111011100010110001010001001100) >> 8 XOR T
[(01100001111111001101001111001000 +
01111011100010110001010001001100) AND 255(10)] =
11011101100001111110100000010100 >> 8 XOR T[20]
00000000110111011000011111101000 XOR
11110111010010001001010000101100 =
11110111100101010001001111000100
D[4] = 11110111100101010001001111000100
```

 Putaran ke 5

FungsiM (A[4], D[4]) = Fungsi M

```
(11101000101111101111000110000010 ,
11110111100101010001001111000100) =
(11101000101111101111000110000010 +
11110111100101010001001111000100) >> 8 XOR T
[(11101000101111101111000110000010 +
11110111100101010001001111000100) AND 255(10)] =
11100000010101000000010101000110 >> 8 XOR T[70]
00000000111000000101010000000101 XOR
00010000010010101011011000001100 =
00010000101010101110001000001001
A[5] = 00010000101010101110001000001001
```

FungsiM (B[4], A[5]) = Fungsi M

```
(11110101101101001110011100111111 ,
00010000101010101110001000001001) =
(11110101101101001110011100111111 +
00010000101010101110001000001001) >> 8 XOR T
[(11110101101101001110011100111111 +
00010000101010101110001000001001) AND 255(10)] =
0000011001011111100100101001000 >> 8 XOR T[72]
0000000000000110010111111001001 XOR
10101011111000001001010110101100 =
10101011111001101100101001100101
B[5] = 10101011111001101100101001100101
```

FungsiM (C[4], B[5]) = Fungsi M

```
(01111011100010110001010001001100 ,
10101011111001101100101001100101) =
(01111011100010110001010001001100 +
10101011111001101100101001100101) >> 8 XOR T
[(01111011100010110001010001001100 +
10101011111001101100101001100101) AND 255(10)] =
00100111011100011101111010110001 >> 8 XOR T[177]
00000000001001110111000111011110 XOR
01111011100111011010101100011101 =
01111011101110101101101011000011
C[5] = 01111011101110101101101011000011
```

FungsiM (D[4], C[5]) = Fungsi M

```
(11110111100101010001001111000100 ,
01111011101110101101101011000011) =
(11110111100101010001001111000100 +
01111011101110101101101011000011) >> 8 XOR T
[(11110111100101010001001111000100 +
01111011101110101101101011000011) AND 255(10)] =
0111001101001111111011010000111 >> 8 XOR T[135]
00000000011100110100111111101110 XOR
```

```

11011101100000110101000001011011 =
11011101111100000001111110110101
D[5] = 11011101111100000001111110110101

```

```

=====
PROSES ENKRIPSI
=====

```

```

Plain Text : 'kriptografi wake'
Kode Ascii Dari 'k' = 01101011
Kode Ascii Dari 'r' = 01110010
Kode Ascii Dari 'i' = 01101001
Kode Ascii Dari 'p' = 01110000
Kode Ascii Dari 't' = 01110100
Kode Ascii Dari 'o' = 01101111
Kode Ascii Dari 'g' = 01100111
Kode Ascii Dari 'r' = 01110010
Kode Ascii Dari 'a' = 01100001
Kode Ascii Dari 'f' = 01100110
Kode Ascii Dari 'i' = 01101001
Kode Ascii Dari ' ' = 00100000
Kode Ascii Dari 'w' = 01110111
Kode Ascii Dari 'a' = 01100001
Kode Ascii Dari 'k' = 01101011
Kode Ascii Dari 'e' = 01100101
Plain Text (Dalam Biner) =
01101011011100100110100101110000011101000110111101100111011100100
110000101100110011010010010000001110111011000010110101101100101
Kunci Dari Proses Pembentukan Kunci =
11011101111100000001111110110101
Chiper Text = Plain Text XOR Key
01101011 XOR 11011101 = 10110110 = ¶
01110010 XOR 11110000 = 10000010 = ,
01101001 XOR 00011111 = 01110110 = v
01110000 XOR 10110101 = 11000101 = Å
01110100 XOR 11011101 = 10101001 = ©
01101111 XOR 11110000 = 10011111 = Ÿ
01100111 XOR 00011111 = 01111000 = x
01110010 XOR 10110101 = 11000111 = Ç
01100001 XOR 11011101 = 10111100 = ¼
01100110 XOR 11110000 = 10010110 = -
01101001 XOR 00011111 = 01110110 = v
00100000 XOR 10110101 = 10010101 = •
01110111 XOR 11011101 = 10101010 = a
01100001 XOR 11110000 = 10010001 = '
01101011 XOR 00011111 = 01110100 = t
01100101 XOR 10110101 = 11010000 = Ð
Hasil Proses Enkripsi = ¶,vÅ©ŸxÇ¼-v•a'tÐ

```

```

=====
PROSES DEKRIPSI
=====

```

```

Chiper Text : '¶,vÅ©ŸxÇ¼-v•a'tÐ'
Kode Ascii Dari '¶' = 10110110
Kode Ascii Dari ', ' = 10000010

```

```

Kode Ascii Dari 'v' = 01110110
Kode Ascii Dari 'Å' = 11000101
Kode Ascii Dari '©' = 10101001
Kode Ascii Dari 'ÿ' = 10011111
Kode Ascii Dari 'x' = 01111000
Kode Ascii Dari 'Ç' = 11000111
Kode Ascii Dari '¼' = 10111100
Kode Ascii Dari '-' = 10010110
Kode Ascii Dari 'v' = 01110110
Kode Ascii Dari '•' = 10010101
Kode Ascii Dari 'a' = 10101010
Kode Ascii Dari ''' = 10010001
Kode Ascii Dari 't' = 01110100
Kode Ascii Dari 'ð' = 11010000
Chiper Text (Dalam Biner) =
10110110100000100111011011000101101010011001111101111000110001111
011110010010110011101101001010110101010100100010111010011010000
Kunci Dari Proses Pembentukan Kunci =
1101110111110000000111110110101
Plain Text = Chiper Text XOR Key
10110110 XOR 11011101 = 01101011 = k
10000010 XOR 11110000 = 01110010 = r
01110110 XOR 00011111 = 01101001 = i
11000101 XOR 10110101 = 01110000 = p
10101001 XOR 11011101 = 01110100 = t
10011111 XOR 11110000 = 01101111 = o
01111000 XOR 00011111 = 01100111 = g
11000111 XOR 10110101 = 01110010 = r
10111100 XOR 11011101 = 01100001 = a
10010110 XOR 11110000 = 01100110 = f
01110110 XOR 00011111 = 01101001 = i
10010101 XOR 10110101 = 00100000 =
10101010 XOR 11011101 = 01110111 = w
10010001 XOR 11110000 = 01100001 = a
01110100 XOR 00011111 = 01101011 = k
11010000 XOR 10110101 = 01100101 = e
Hasil Proses Deskripsi = kriptografi wake

```

Dari tabel 4.1 diketahui bahwa hasil dari proses enkripsi dan dekripsi perangkat lunak kriptografi WAKE telah sesuai dengan metode kriptografi WAKE. Hal tersebut dapat dilihat dari hasil enkripsi dan dekripsi yang sama yang dilakukan secara manual maupun dilakukan dengan perangkat lunak kriptografi WAKE.

4.3. Kelebihan dan Kekurangan Perangkat Lunak

Perangkat lunak kriptografi dengan metode *Word Auto Key Encryption* (WAKE) memiliki beberapa kelebihan dan kekurangan. Kelebihan dan

kekurangan perangkat lunak kriptografi dengan metode *Word Auto Key Encryption* (WAKE) adalah sebagai berikut :

1.3.1 Kelebihan Perangkat Lunak

Kelebihan dari perangkat lunak kriptografi dengan metode *Word Auto Key Encryption* (WAKE) adalah sebagai berikut :

- a. Pengguna dapat mengatur jumlah putaran dalam proses pembentukan kunci.
- b. *Plaintext* dapat berupa *input*-an manual dan file (.txt).
- c. Kunci dapat kurang dari 16 karakter.
- d. Perangkat lunak dapat melakukan proses S-Box, proses Pembentukan Kunci, proses enkripsi, dan proses dekripsi dengan metode WAKE.

1.3.2 Kekurangan Perangkat Lunak

Kekurangan dari perangkat lunak kriptografi dengan metode *Word Auto Key Encryption* (WAKE) adalah sebagai berikut :

- a. Panjang *plaintext* maksimal 1000 karakter.
- b. Panjang kunci maksimal 16 karakter.
- c. Jumlah putaran dalam proses pembentukan kunci maksimal 50 putaran.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang dapat ditarik dari hasil perancangan dan pengujian perangkat lunak kriptografi dengan metode *Word Auto Key Encryption* (WAKE) adalah sebagai berikut :

- a. Perangkat lunak dapat melakukan proses enkripsi untuk menjaga kerahasiaan data.
- b. Perangkat Lunak melakukan enkripsi dan dekripsi data menggunakan metode *Word Auto Key Encryption* (WAKE).

5.2. Saran

Saran untuk pengembangan perangkat lunak kriptografi dengan metode *Word Auto Key Encryption* (WAKE) adalah sebagai berikut :

- a. Perangkat lunak dapat menggunakan lebih dari satu metode kriptografi.
- b. Perangkat lunak dapat melakukan lebih dari 50 putaran dalam proses pembentukan kunci.
- c. Perangkat lunak dapat membaca *input* berupa file .docx, .xls, dll.

DAFTAR PUSTAKA

- Dahria, M., Rahim, A., & Jaya, H. 2012. *Perangkat Lunak Pembelajaran Kriptografi Metode WAKE*: Program Studi Teknik Informatika STMIK Triguna Dharma.
- David, Wheeler. 1999. *A Bulnk Data Encryption Algorithm*: Computer Laboratory, Cambridge University, UK.
- Heriyanto, Tedi. 1999. *Pengenalan Kriptografi* (On-line) Available at http://www.tedi-h.com/papers/p_kripto.html
- K, Jusuf. Ir. M. T. 2004. *Kriptografi Keamanan Internet dan Jaringan Komunikasi*, Bandung: Penerbit Informatika.
- Munir, Rinaldi. 2006. *Kriptografi*. Bandung:Penerbit Informatika.
- S, Bruce. 1996. *Applied Crytography Second Edition*. New York: John Wiley & Sons, Inc.