



**RANCANG BANGUN KEAMANAN PENGIRIMAN PESAN DENGAN TEKNIK  
PERGESERAN BIT SECARA HORIZONTAL PADA CIPHER SUBSTITUSI**

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

**SKRIPSI**

**OLEH:**

**NAMA : EKA ANDY NOVA**  
**NPM : 1414370458**  
**PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS PEMBANGUNAN PANCA BUDI  
MEDAN  
2020**

**LEMBAR PENGESAHAN**

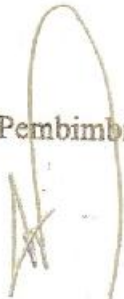
**RANCANG BANGUN KEAMANAN PENGIRIMAN PESAN DENGAN TEKNIK  
PERGESERAN BIT SECARA HORIZONTAL PADA CIPHER SUBSTITUSI**

**Disusun Oleh:**

**NAMA : EKA ANDY NOVA**  
**NPM : 1414370458**  
**PROGRAM STUDI : SISTEM KOMPUTER**

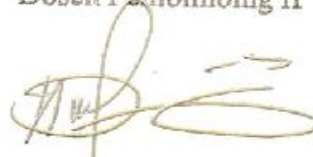
**Skripsi Telah Disetujui oleh Dosen Pembimbing Skripsi  
Pada Tanggal :**

**Dosen Pembimbing I**



**A. P. U. Siahaan, S.Kom., M.Kom.**

**Dosen Pembimbing II**



**Dian Kurnia, S.Kom., M.Kom.**

**Mengetahui:**

**Dekan Fakultas Sains dan Teknologi**



**Hamdani, S.T., M.T.**

**Ketua Program Studi Sistem Komputer**



**Eke Hariyanto, S.Kom., M.Kom.**

## SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Eka Andy Nova  
NPM : 1414370458  
Prodi : Sistem Komputer  
Konsentrasi : Keamanan Jaringan Komputer  
Judul Skripsi : Rancang Bangun Keamanan Pengiriman Pesan Dengan Teknik Pergeseran BIT Secara Horizontal Pada Cipher Subtitusi

Dengan ini menyatakan bahwa :

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

Demikian pernyataan ini saya perbuat dengan sebenar-benarnya, terimakasih.

Medan,

Yang membuat pernyataan



Eka Andy Nova



# UNIVERSITAS PEMBANGUNAN PANCA BUDI FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PO.BOX : 1099 MEDAN

PROGRAM STUDI TEKNIK ELEKTRO	(TERAKREDITASI)
PROGRAM STUDI ARSITEKTUR	(TERAKREDITASI)
PROGRAM STUDI SISTEM KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI TEKNIK KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI AGROTEKNOLOGI	(TERAKREDITASI)
PROGRAM STUDI PETERNAKAN	(TERAKREDITASI)

## PERMOHONAN JUDUL TESIS / SKRIPSI / TUGAS AKHIR\*

Yang bertanda tangan di bawah ini :

Nama Lengkap : EKA ANDY NOVA  
 Tempat/Tgl. Lahir : MARELAN / 14 Oktober 1994  
 Nomor Pokok Mahasiswa : 1414370458  
 Program Studi : Sistem Komputer  
 Konsentrasi : Keamanan Jaringan Komputer  
 Jumlah Kredit yang telah dicapai : 141 SKS, IPK 3.16  
 Nomor Hp : 082366543145  
 Dengan ini mengajukan judul sesuai bidang ilmu sebagai berikut :

### Judul

Rancang Bangun keamanan pengiriman pesan dengan teknik pergeseran bit secara horizontal pada cipher substitusi0

Isian : Diisi Oleh Dosen Jika Ada Perubahan Judul

Yang Tidak Perlu



( Ir. Bhaeti Alamsyah, M.T., Ph.D. )

Medan, ~~03 Januari 2020~~

Pemohon,

( Eka Andy Nova )

Tanggal : .....  
 Disahkan oleh :  
 Dehan  
  
 ( Hamdani, ST, MT )

Tanggal : .....  
 Disetujui oleh :  
 Dosen Pembimbing I :

( Andysah Putera Utama Siahaan, S.Kom, M.Kom )

Tanggal : .....  
 Disetujui oleh :  
 Ka. Prodi Sistem Komputer

( Eko Hariyanto, S.Kom, M.Kom )

Tanggal : .....  
 Disetujui oleh :  
 Dosen Pembimbing II :

( Dian Kurnia, S.Kom, M.Kom )

No. Dokumen: FM-UPBM-18-02

Revisi: 0

Tgl. Eff: 22 Oktober 2018



UNIVERSITAS PEMBANGUNAN PANCA BUDI  
**FAKULTAS SAINS & TEKNOLOGI**

Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571  
 website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id  
 Medan - Indonesia

Universitas : Universitas Pembangunan Panca Budi  
 Fakultas : SAINS & TEKNOLOGI  
 Dosen Pembimbing I : Andyasa P. U. Sihazan, S.Kom., M.Kom., Ph.D  
 Dosen Pembimbing II : Dian Kurnia S.Kom., M.Kom  
 Nama Mahasiswa : EKA ANDY NOVA  
 Jurusan/Program Studi : Sistem Komputer  
 Nomor Pokok Mahasiswa : 1414370458  
 Bidang Pendidikan : Strata 1 (SI)  
 Judul Tugas Akhir/Skripsi : Rancangan Bangun Keamanan Pengiriman Pesan Dengan Teknik Pergeseran Bit Secara Horizontal Pada Cipher Substitusi

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
1/10 2019	Acc Semua Judul		
1/10	Revisi Bab I		
1/11	Revisi Bab II		
1/11	Revisi Bab III		
1/11	Revisi Bab III, IV		
1/12	Revisi Bab IV, V		
1/12	Acc Semuanya Hasil		
7/1 2020	Acc Sidang		
3/10	Acc Jilid		

Medan, 21 November 2019

Diketahui/Ditetujui oleh :

Dekan,



Hamdani, S.T., M.T.



UNIVERSITAS PEMBANGUNAN PANCA BUDI  
**FAKULTAS SAINS & TEKNOLOGI**

Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571  
 website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id  
 Medan - Indonesia

Universitas : Universitas Pembangunan Panca Budi  
 Fakultas : SAINS & TEKNOLOGI  
 Pembimbing I : Andy Sah P.U. Sjahran, S.Kom., M.Kom., Ph.D.  
 Pembimbing II : Dian Kurniawan S.Kom., M.Kom  
 Mahasiswa : EKA ANDY NOVA  
 Jurusan/Program Studi : Sistem Komputer  
 NPM/Pokok Mahasiswa : 1414370458  
 Tingkat Pendidikan : Strata 1 (S1)  
 Tugas Akhir/Skripsi : Rancang Bangun Keamanan Pengiriman Pesan  
 Dengan Teknik Pergeseran Bit Secara Horisontal  
 Pada Cipher Substitusi

WAKTU	PEMBAHASAN MATERI	PARAF	KETERANGAN
30-2019	Ace Bab 1	<i>[Signature]</i>	Ace Seminar proposal
10-2019	perbaiki bab 2 lengkapi daftar pustaka, sesuaikan dengan panduan skripsi	<i>[Signature]</i>	
0-2019	Ace bab 2 lanjut bab 3	<i>[Signature]</i>	
11-2019	Ace bab 3 lanjut bab 4	<i>[Signature]</i>	
11-2019	lengkapi daftar pustaka, Ace bab 4 lanjut kesimpulan dan saran	<i>[Signature]</i>	
11-2019	Ace kesimpulan dan saran	<i>[Signature]</i>	Ace seminar hasil
12-2019	lengkapi bab 5 hasil	<i>[Signature]</i>	Ace sidang
01-2020	lengkapi bab 5 sidang	<i>[Signature]</i>	Ace sidang
10-2020	Ace jilid	<i>[Signature]</i>	Ace jilid

Medan, 04 Oktober 2019

Diketahui/Ditetujui oleh :  
 Dekan,



*Hamdani, S.T., M.T.*

Telah Diperiksa oleh LPMU dengan Plagiarisme...21...%

Medan 23 JANUARI 2020

FM-BPAA-2012-041

Hal : Permohonan Meja Hijau



Medan, 23 Januari 2020  
Kepada Yth : Bapak/Ibu Dekan  
Fakultas SAINS & TEKNOLOGI  
UNPAB Medan



Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama : EKA ANDY NOVA  
Tempat/Tgl. Lahir : Marelan / 14 Oktober 1994  
Nama Orang Tua : MUHAMMAD  
N. P. M : 1414370458  
Fakultas : SAINS & TEKNOLOGI  
Program Studi : Sistem Komputer  
No. HP : 082366543145  
Alamat : Jurong Bay Pass

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul Rancang Bangun keamanan pengiriman pesan dengan teknik pergeseran bit secara horizontal pada cipher substitusi, Selanjutnya saya menyatakan :

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

1. [102] Ujian Meja Hijau	: Rp.	600,000
2. [170] Administrasi Wisuda	: Rp.	1,500,000
3. [202] Bebas Pustaka	: Rp.	400,000
4. [221] Bebas LAB	: Rp.	5,000
Total Biaya	: Rp.	2,100,000
5. Uang kuliah	Rp	1,705,000
	Rp	3,805,000
	Rp	4,955,000

27/20  
y dtz

Periode Wisuda Ke : 64

Ukuran Toga : L



Hormat saya  
*Eka Andy Nova*  
EKA ANDY NOVA  
1414370458

Catatan :

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



## Plagiarism Detector v. 1460 - Originality Report

Analyzed document: 01/23/20 14:51:02

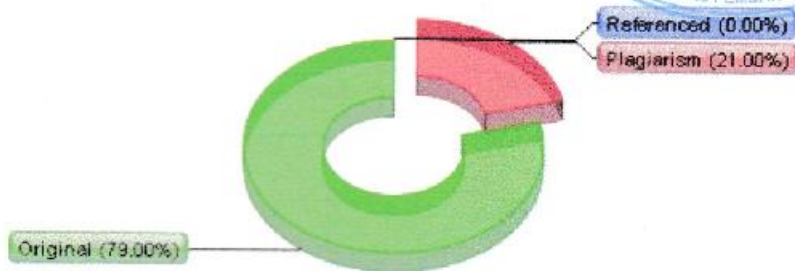
# "EKA ANDY NOVA\_1414370458\_SISTEM KOMPUTER.doc"

Check Type: Internet - via Google and Bing

Licensed to: Universitas Pembangunan Panca Budi\_License03



Relation chart:



Distribution graph:

Comparison Preset: Rewrite. Detected language: Indonesian

Top sources of plagiarism:

% 5	wrds: 391	<a href="http://repository.uinjkt.ac.id/dspace/bitstream/123456789/5251/1/BERNES%20LASE-F...">http://repository.uinjkt.ac.id/dspace/bitstream/123456789/5251/1/BERNES%20LASE-F...</a>
% 4	wrds: 297	<a href="https://repository.bsi.ac.id/index.php/unduh/item/2621/PERANCANGAN-SISTEM-INFORM...">https://repository.bsi.ac.id/index.php/unduh/item/2621/PERANCANGAN-SISTEM-INFORM...</a>
% 4	wrds: 277	<a href="https://kuliah-kami.blogspot.com/2011/12/makalah-perbandingan-visual-basic-c-dan...">https://kuliah-kami.blogspot.com/2011/12/makalah-perbandingan-visual-basic-c-dan...</a>

How other Sources:]

Processed resources details:

93 - Ok / 8 - Failed

How other Sources:]

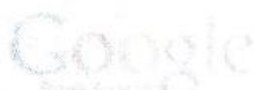
Important notes:

Wikipedia:

Google Books:

Ghostwriting services:

Anti-cheating:



[not detected]

[not detected]

[not detected]

[not detected]

Active References (Urls Extracted from the Document):

URLs detected

Excluded Urls:





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

**KARTU BEBAS PRAKTIKUM**

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

Nama : EKA ANDY NOVA  
N.P.M : 1414370458  
Tingkat/Semester : Akhir  
Fakultas : SAINS & TEKNOLOGI  
Jurusan/Prodi : Sistem Komputer

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

Medan, 23 Januari 2020



## **ABSTRAK**

**EKA ANDY NOVA**

**Rancang Bangun Keamanan Pengiriman Pesan Dengan Teknik Pergeseran  
Bit Secara Horizontal Pada Cipher Substitusi  
2020**

Pesan teks, atau mengirim pesan teks, adalah tindakan menyusun dan mengirim pesan elektronik, biasanya terdiri dari karakter alfabet dan numerik, antara dua atau lebih pengguna perangkat seluler, desktop laptop, atau jenis komputer lain yang kompatibel. Pesan teks digunakan untuk keperluan pribadi, keluarga, bisnis dan sosial. Pengiriman pesan adalah merupakan hal yang harus dijaga keamanannya. Pesan tersebut dapat saja mengandung informasi yang sangat rahasia sehingga tidak boleh diketahui oleh orang lain. Teknik kriptografi adalah salah satu yang dapat mengamankan pesan pada saat pengiriman. Dengan teknik kriptografi, pesan sebelum dikirim harus diubah menjadi pesan tak terbaca atau terenkripsi. Teknik pergeseran bit secara horizontal adalah suatu metode yang bekerja dengan cara menggeser posisi bit pada karakter. Metode ini bekerja sangat baik dalam mentransformasikan pesan ke dalam bentuk pesan terenkripsi. Algoritma ini bekerja dengan cara mengganti atau menggeser posisi biner dari karakter yang akan dikirim. Dengan menerapkan algoritma teknik pergeseran bit secara horizontal maka, tingkat keamanan pada pengiriman pesan dapat ditingkatkan.

**Kata Kunci:** dekripsi, enkripsi, kriptografi, HBR, substitusi

## KATA PENGANTAR

*Assalamuallaikum Wr. Wb.*

Puji syukur ke hadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan judul **“Rancang Bangun Keamanan Pengiriman Pesan Dengan Teknik Pergeseran Bit Secara Horizontal Pada Cipher Substitusi”**.

Skripsi ini ditulis guna memenuhi salah satu syarat pemenuhan kurikulum dalam menyelesaikan pendidikan pada Program Studi S1 Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan. Pada kesempatan ini, penulis menyampaikan rasa terima kasih dan penghargaan yang sebesar-besarnya kepada :

1. Kepada kedua orang tua yang telah mendampingi penulis dalam menyelesaikan skripsi ini dengan baik.
2. Bapak Dr. H. M. Isa Indrawan, SE.,MM., selaku Rektor Universitas Pembangunan Panca Budi Medan.
3. Bapak Hamdani, S.T., M.T., selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. Bapak Eko Hariyanto, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Komputer Universitas Pembangunan Panca Budi Medan.
5. Bapak Andysah Putera Utama Siahaan, S.Kom., M.Kom., selaku Dosen Pembimbing I, Universitas Pembangunan Panca Budi, Medan.
6. Bapak Dian Kurnia S.Kom., M.Kom. selaku Dosen Pembimbing II, Universitas Pembangunan Panca Budi, Medan.
7. Dosen-dosen pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi, Universitas Pembangunan Panca Budi, Medan.
8. Seluruh staf dan karyawan pada Universitas Pembangunan Panca Budi, Medan.
9. Teman-teman seperjuangan yang sudah meluangkan waktu dan memberikan semangat dan motivasi dalam penyelesaian skripsi ini.

Akhir kata penulis sampaikan rasa terima kasih bagi semua pihak yang secara langsung terlibat dalam penyelesaian skripsi ini yang tidak dapat penulis sebutkan satu persatu. Semoga skripsi ini memberikan manfaat bagi penulis khususnya dan bagi kita semua umumnya.

Medan, 14 Oktober 2020  
Penulis

Eka Andy Nova  
1414370458

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR GAMBAR.....</b>	<b>iv</b>
<b>DAFTAR TABEL .....</b>	<b>v</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	4
<b>BAB II LANDASAN TEORI.....</b>	<b>5</b>
2.1 Keamanan Data .....	5
2.1.1 Elemen Keamanan Data.....	5
2.1.2 Pertimbangan Keamanan Data .....	6
2.1.3 Penilaian Risiko Data .....	6
2.1.4 Minimalisasi Data.....	7
2.2 Kriptografi.....	7
2.2.1 Kriptografi Simetris .....	9
2.2.2 Kriptografi Asimetris.....	12
2.3 Horizontal Bit Rotation .....	14
2.4 Cipher Substitusi.....	15
2.5 Sistem Informasi.....	16
2.6 Algoritma .....	17
2.7 Unified Modelling Language (UML) .....	18
2.7.1 Use Case Diagram .....	19
2.7.2 Activity Diagram .....	22
2.7.3 Sequence Diagram.....	24
2.7.4 Class Diagram .....	25
2.7.5 Flowchart .....	26
2.8 Bahasa Pemrograman.....	29
2.8.1 Kode Program .....	31
2.8.2 Intepreter .....	32
2.8.3 Compiler .....	33
2.9 Visual Basic.Net .....	34
<b>BAB III METODE PENELITIAN .....</b>	<b>37</b>
3.1 Kerangka Penelitian.....	37
3.2 Perancangan Penelitian .....	40
3.2.1 Use Case Diagram .....	40
3.2.2 Activity Diagram .....	43
3.2.3 Flowchart Enkripsi .....	45

3.2.4	Flowchart Dekripsi .....	46
3.3	Desain Interface .....	47
3.3.1	Desain Menu Utama .....	47
3.3.2	Desain Kriptografi HBR .....	48
3.3.3	Desain Info .....	49
3.3.4	Desain Tentang Saya .....	50
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>		<b>51</b>
4.1	Spesifikasi Sistem.....	51
4.1.1	Spesifikasi Perangkat Keras.....	52
4.1.2	Spesifikasi Perangkat Lunak .....	52
4.2	Implementasi Interface.....	53
4.2.1	Implementasi Menu Utama.....	53
4.2.2	Implementasi Info.....	54
4.2.3	Implementasi Tentang Saya .....	55
4.2.4	Implementasi Kriptografi HBR.....	55
4.2.5	Hasil Perhitungan Kriptografi HBR .....	56
4.3	Pengujian Perhitungan .....	58
<b>BAB V PENUTUP .....</b>		<b>61</b>
5.1	Kesimpulan.....	61
5.2	Saran .....	61

## **DAFTAR PUSTAKA**

## DAFTAR GAMBAR

Gambar 2.1 Skema proses enkripsi dan dekripsi.....	8
Gambar 2.2 Skema kriptografi simetris .....	11
Gambar 2.3 Skema kriptografi asimetris .....	13
Gambar 2.4 Skema pergeseran bit pada HBR .....	14
Gambar 2.5. Use-case Diagram ATM .....	20
Gambar 2.6 Contoh Sequence Diagram.....	25
Gambar 2.7 Tampilan Toolbox Visual Basic .....	35
Gambar 3.1 Kerangka Penelitian.....	38
Gambar 3.2 Use Case Diagram Enkripsi .....	41
Gambar 3.3 Use Case Diagram Dekripsi .....	42
Gambar 3.4 Activity Diagram Enkripsi .....	43
Gambar 3.5 Activity Diagram Dekripsi .....	44
Gambar 3.6 Flowchart enkripsi algoritma HBR.....	45
Gambar 3.7 Flowchart dekripsi algoritma HBR.....	46
Gambar 3.8 Desain Menu Utama .....	47
Gambar 3.9 Desain Kriptografi HBR .....	48
Gambar 3.10 Desain Info .....	49
Gambar 3.11 Desain Tentang Saya.....	50
Gambar 4.1 Implementasi Menu Utama .....	54
Gambar 4.2 Implementasi Info .....	54
Gambar 4.3 Implementasi Tentang Saya .....	55
Gambar 4.4 Implementasi Kriptografi HBR .....	56
Gambar 4.5 Implementasi enkripsi algoritma HBR .....	57
Gambar 4.6 Implementasi dekripsi algoritma HBR .....	58

## DAFTAR TABEL

Tabel 2.1 Simbol Use Case Diagram.....	21
Tabel 2.2 Simbol Activity Diagram.....	23
Tabel 2.3 Simbol Sequence Diagram.....	24
Tabel 2.4 Simbol Class Diagram.....	25
Tabel 2.5 Simbol Flowchart.....	28
Tabel 2.6 Toolbox Visual Basic.....	36
Tabel 4.1 Spesifikasi perangkat keras.....	52
Tabel 4.2 Spesifikasi perangkat lunak.....	52

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Data merupakan kumpulan informasi yang memiliki sebuah pola. Data harus dijaga dengan baik. Data tidak boleh sampai jatuh ke tangan orang lain yang bukan pemiliknya. Apabila data ini berhasil dicuri, maka akan sangat merugikan pihak pemiliknya apalagi jika data tersebut mengandung unsur-unsur keuangan dan surat-surat yang sangat berharga. Data menjadi semakin berharga karena mengandung informasi rahasia. Selain itu, keterampilan dan peluang orang yang tidak bertanggung jawab dalam hal mengambil berbagai jenis data pribadi berkembang sangat cepat. Pencurian data diakibatkan oleh kelalaian atau kecerobohan seseorang. Hal ini dapat menyebabkan kerugian besar bagi pemiliknya terlebih-lebih jika data tersebut adalah data perusahaan.

Tujuan perlindungan data bukan hanya untuk melindungi data seseorang, tetapi untuk melindungi hak dasar dan kebebasan orang yang berhubungan dengan isi dan tujuan data tersebut. Sementara melindungi data pribadi, dimungkinkan untuk memastikan bahwa hak dan kebebasan orang tidak dilanggar. Sebagai contoh penyebaran data pribadi melalui media sosial. Manipulasi data juga dapat merugikan seseorang dalam kehidupannya. Misalnya, pemrosesan data pribadi yang salah, dapat menyebabkan situasi di mana seseorang diabaikan untuk kesempatan kerja atau, lebih buruk lagi, kehilangan pekerjaan saat ini.



Ada banyak cara untuk mengamankan data, salah satu caranya adalah dengan menerapkan ilmu kriptografi dalam melindungi data tersebut. Kriptografi tidak menghilangkan atau menyembunyikan data, melainkan mengubah data tersebut menjadi bentuk yang lain atau ke dalam bentuk yang tidak dapat difahami oleh manusia. Kriptografi adalah ilmu untuk mengelabui informasi yang dilihat di depan mata. Hal ini bertujuan untuk menghindari dari akses yang tidak sah. Ini adalah teknik menyimpan dan mengirimkan data dalam bentuk tertentu sehingga hanya mereka yang dimaksudkan dapat membaca dan memprosesnya. Kriptografi memungkinkan transaksi elektronik dilakukan secara aman. Sebelum data dipertukarkan, maka data tersebut harus dienkripsi. Karena banyaknya pertukaran data pada jaringan lokal dan global, kriptografi adalah hal sangat penting dalam memastikan keamanan data.

Teknik pergeseran bit secara horizontal atau yang disebut dengan algoritma *Horizontal Bit Rotation (HBR)* merupakan algoritma yang baik dalam melakukan proses enkripsi. Kecepatan algoritma ini sangat baik karena hanya melakukan pergeseran bit pada tiap karakter plaintext. Pergeseran maksimal yang dapat dilakukan adalah sebesar 7 bit. Dengan menerapkan algoritma HBR diharapkan keamanan data lebih dapat ditingkatkan. Berdasarkan latar belakang di atas maka penulis tertarik untuk memilih judul “**Rancang Bangun Keamanan Pengiriman Pesan Dengan Teknik Pergeseran Bit Secara Horizontal Pada Cipher Substitusi**”.

## **1.2 Rumusan Masalah**

Adapun rumusan masalah yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Bagaimana melakukan proses enkripsi dan dekripsi dengan teknik pergeseran bit secara horizontal?
2. Bagaimana menentukan bit yang mengalami perubahan berdasarkan kunci?
3. Bagaimana menentukan kunci pada algoritma pergeseran bit?

## **1.3 Batasan Masalah**

Adapun batasan masalah yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Karakter yang digunakan adalah berdasarkan tabel ASCII.
2. Jumlah pergeseran maksimal adalah bernilai 7.
3. Pesan yang digunakan sebagai plaintext adalah pesan berbasis teks.

## **1.4 Tujuan Penelitian**

Adapun tujuan penelitian yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Bagaimana melakukan proses enkripsi dan dekripsi dengan teknik pergeseran bit secara horizontal?
2. Bagaimana menentukan bit yang mengalami perubahan berdasarkan kunci?
3. Bagaimana menentukan kunci pada algoritma pergeseran bit?

## **1.5 Manfaat Penelitian**

Adapun manfaat penelitian yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Dapat menghindari pencurian data dari pihak yang tidak bertanggung jawab.
2. Dapat mengamankan pesan yang akan dikirim ke pihak penerima.
3. Informasi yang berhasil didapat tidak dengan mudah dapat dipecahkan karena memiliki teknik kriptografi.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Keamanan Data**

Keamanan Data adalah proses melindungi informasi, file, database, dan akun pada jaringan dengan mengadopsi serangkaian kontrol, aplikasi, dan teknik yang mengidentifikasi kepentingan relatif dari set data yang berbeda, sensitivitasnya, persyaratan kepatuhan peraturan dan kemudian menerapkan perlindungan yang sesuai untuk mengamankan sumber daya. Mirip dengan pendekatan lain seperti keamanan perimeter, keamanan file atau keamanan perilaku pengguna, keamanan data bukanlah segalanya, semua-akhir untuk praktik keamanan. Hal ini adalah salah satu metode untuk mengevaluasi dan mengurangi risiko yang datang dengan menyimpan segala jenis data (William Stallings, 2005).

##### **2.1.1 Elemen Keamanan Data**

Elemen inti dari keamanan data adalah kerahasiaan, integritas, dan ketersediaan. Juga dikenal sebagai triad CIA, ini adalah model keamanan dan panduan bagi organisasi untuk menjaga data sensitif mereka terlindungi dari akses yang tidak sah dan pengusiran data.

1. Kerahasiaan memastikan bahwa data hanya diakses oleh individu yang berwenang.
2. Integritas memastikan bahwa informasi dapat diandalkan dan juga akurat.

3. Ketersediaan memastikan bahwa data tersedia dan dapat diakses untuk memenuhi kebutuhan bisnis.

### **2.1.2 Pertimbangan Keamanan Data**

Ada beberapa pertimbangan keamanan data yang harus ada dan dilakukan untuk menjaga informasi:

1. Di mana data sensitif berada? Seseorang tidak akan tahu bagaimana melindungi data jika tidak tahu di mana data sensitif disimpan.
2. Siapa yang memiliki akses ke data? Ketika pengguna memiliki akses yang tidak diperiksa atau ulasan izin yang jarang, itu membuat organisasi berisiko terhadap penyalahgunaan data, pencurian, atau penyalahgunaan. Mengetahui siapa yang memiliki akses ke data perusahaan setiap saat adalah salah satu pertimbangan keamanan data paling vital yang harus dimiliki.
3. Sudahkah seseorang menerapkan pemantauan berkelanjutan dan peringatan real-time pada data? Pemantauan berkelanjutan dan peringatan waktu-nyata adalah penting tidak hanya untuk memenuhi peraturan kepatuhan, tetapi dapat mendeteksi aktivitas file yang tidak biasa, akun yang mencurigakan, dan perilaku komputer sebelum terlambat..

### **2.1.3 Penilaian Risiko Data**

Penilaian risiko data membantu perusahaan mengidentifikasi data sensitif mereka yang terlalu banyak terpapar dan menawarkan langkah yang dapat diandalkan dan berulang untuk memprioritaskan dan memperbaiki risiko keamanan

yang serius. Proses dimulai dengan mengidentifikasi data sensitif yang diakses melalui grup global, data basi, dan / atau izin tidak konsisten. Penilaian risiko merangkum temuan-temuan penting, mengekspos kerentanan data, memberikan penjelasan terperinci tentang setiap kerentanan, dan memasukkan rekomendasi remediasi yang diprioritaskan (Buckbee, 2019).

#### **2.1.4 Minimalisasi Data**

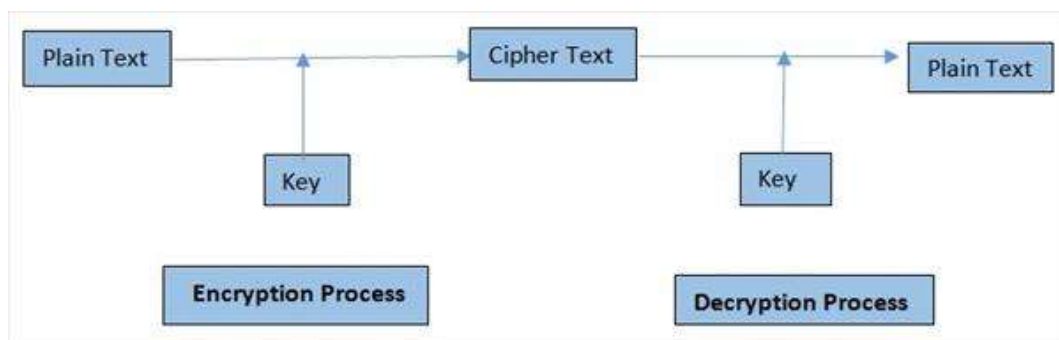
Dekade terakhir manajemen TI telah melihat pergeseran dalam persepsi data. Sebelumnya, memiliki lebih banyak data hampir selalu lebih baik daripada kurang. Seseorang tidak pernah bisa yakin sebelumnya tentang apa yang ingin dilakukan dengannya. Hari ini, data adalah kewajiban. Ancaman dari pelanggaran data yang merusak reputasi, kehilangan jutaan atau denda peraturan yang kaku semua memperkuat pemikiran bahwa mengumpulkan apa pun di luar jumlah minimum data sensitif sangat berbahaya. Untuk itu: ikuti praktik terbaik minimalisasi data dan kaji semua kebutuhan dan prosedur pengumpulan data dari sudut pandang bisnis (Buckbee, 2019).

## **2.2 Kriptografi**

Ilmu enkripsi dan dekripsi data disebut kriptografi. Ini biasanya digunakan untuk memberikan keamanan dan kerahasiaan pada beberapa pesan yang kerahasiaannya perlu disebutkan antara pengirim dan penerima. Ini umumnya dilakukan oleh beberapa algoritma kompleks di mana sejumlah operasi dilakukan pada teks biasa untuk mengubahnya menjadi ciphertext, metode ini dikenal sebagai

enkripsi. Kebalikan dari operasi ini adalah algoritma dekripsi, di mana ciphertext dikonversi kembali menjadi teks biasa. Operasi semacam itu umumnya dilakukan dengan bantuan kunci. Secara sederhana, kunci digunakan untuk mengunci data dan kunci yang sama / kunci berbeda / set kunci digunakan untuk membuka kunci data. Kekuatan algoritma kriptografi umumnya ditentukan oleh kekuatan algoritma dan kerahasiaan kunci.

Contoh: Misalkan Alice ingin mengirim beberapa informasi rahasia tentang kertas ujian IIT kepada Bob. Ia akan mengonversikannya menjadi sandi menggunakan beberapa kunci dan akan dikirim sandi itu kepada Bob. Sekarang Bob sudah mendapatkan cipher tetapi untuk membukanya dia akan membutuhkan kunci yang akan dikirim oleh Alice dalam surat terpisah. Untuk melindungi data dari penyusup, adalah keharusan bahwa ciphertext dan kunci tidak boleh dikirim bersamaan. Gambar berikut ini adalah skema kriptografi:



**Gambar 2.1 Skema proses enkripsi dan dekripsi**

Sumber: (W. Stallings, 2013)

### 2.2.1 Kriptografi Simetris

Enkripsi simetris adalah jenis enkripsi di mana hanya satu kunci (kunci rahasia) yang digunakan untuk mengenkripsi dan mendekripsi informasi elektronik. Entitas yang berkomunikasi melalui enkripsi simetris harus bertukar kunci sehingga dapat digunakan dalam proses dekripsi. Metode enkripsi ini berbeda dari enkripsi asimetris di mana sepasang kunci, satu publik dan satu pribadi, digunakan untuk mengenkripsi dan mendekripsi pesan (Smirnoff & Turner, 2019).

Dengan menggunakan algoritma enkripsi simetris, data dikonversi ke bentuk yang tidak dapat dipahami oleh siapa pun yang tidak memiliki kunci rahasia untuk mendekripsi. Setelah penerima yang dituju yang memiliki kunci memiliki pesan, algoritma membalikkan aksinya sehingga pesan dikembalikan ke bentuk aslinya dan dapat dimengerti. Kunci rahasia yang digunakan pengirim dan penerima dapat berupa kata sandi / kode tertentu atau bisa berupa string acak huruf atau angka yang dihasilkan oleh penghasil angka acak yang aman. Untuk enkripsi tingkat perbankan, kunci simetris harus dibuat menggunakan RNG yang disertifikasi sesuai dengan standar industri, seperti FIPS 140-2. Ada dua jenis algoritma enkripsi simetris:

1. Blok algoritma. Setel panjang bit dienkripsi dalam blok data elektronik dengan penggunaan kunci rahasia tertentu. Saat data sedang dienkripsi, sistem menyimpan data dalam memorinya saat menunggu blok lengkap.
2. Algoritma aliran. Data dienkripsi karena stream bukannya disimpan dalam memori sistem.



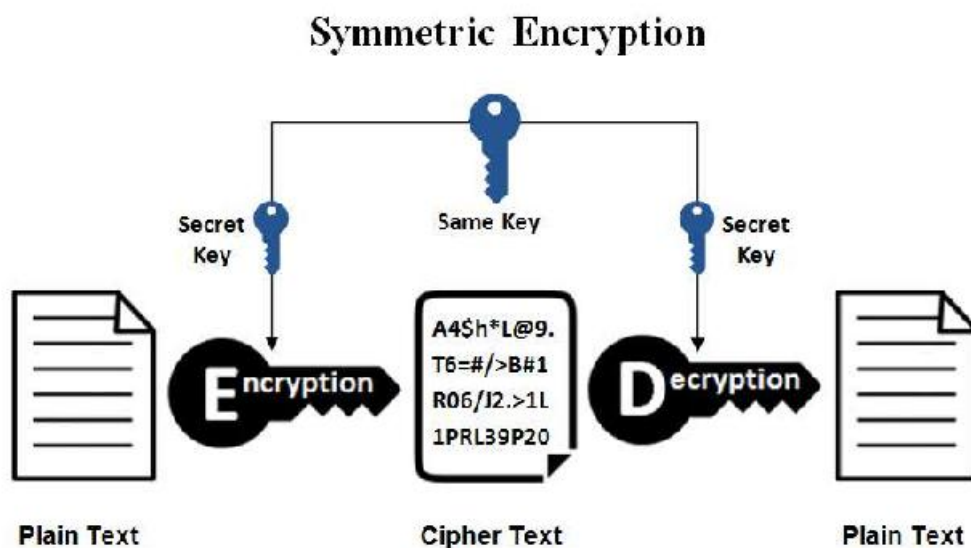
Beberapa contoh algoritma enkripsi simetris meliputi:

1. AES (Standar Enkripsi Lanjutan)
2. DES (Standar Enkripsi Data)
3. IDEA (Algoritma Enkripsi Data Internasional)
4. Blowfish (Pengganti drop-in untuk DES atau IDEA)
5. RC4 (Rivest Cipher 4)
6. RC5 (Rivest Cipher 5)
7. RC6 (Rivest Cipher 6)

AES, DES, IDEA, Blowfish, RC5 dan RC6 adalah cipher blok. RC4 adalah stream cipher.

Sementara enkripsi simetris adalah metode enkripsi yang lebih lama, enkripsi lebih cepat dan lebih efisien daripada enkripsi asimetris, yang memakan banyak biaya pada jaringan karena masalah kinerja dengan ukuran data dan penggunaan CPU yang berat. Karena kinerja yang lebih baik dan kecepatan enkripsi simetris yang lebih cepat (dibandingkan dengan asimetris), kriptografi simetris biasanya digunakan untuk enkripsi massal / mengenkripsi data dalam jumlah besar, mis. untuk enkripsi basis data. Dalam kasus database, kunci rahasia mungkin hanya tersedia untuk database itu sendiri untuk mengenkripsi atau mendekripsi. Beberapa contoh di mana kriptografi simetris digunakan adalah:

1. Aplikasi pembayaran, seperti transaksi kartu di mana PII perlu dilindungi untuk mencegah pencurian identitas atau tuduhan penipuan
2. Validasi untuk mengonfirmasi bahwa pengirim pesan adalah siapa yang ia klaim
3. Pembuatan angka acak atau hashing



**Gambar 2.2 Skema kriptografi simetris**

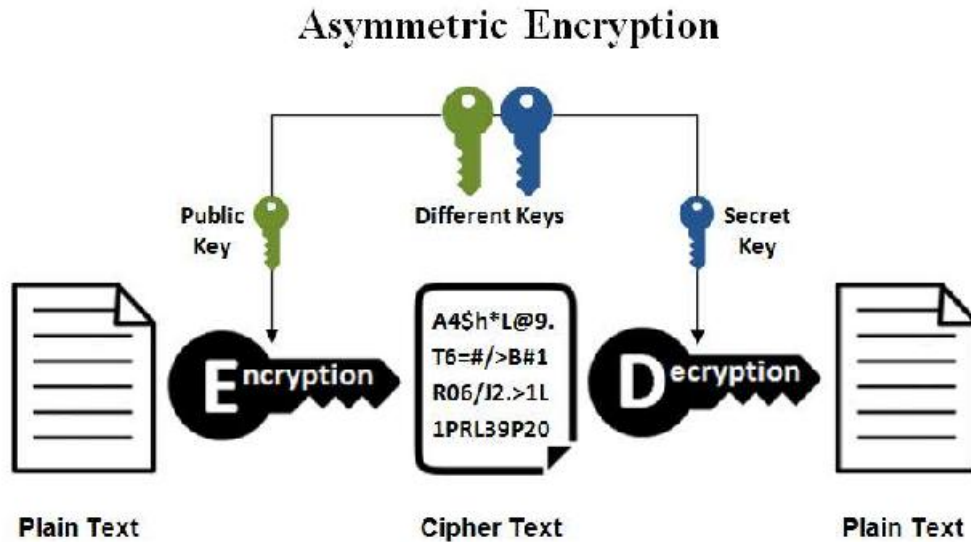
Sumber: (Information, 2019)

Gambar 2.2 menjelaskan jenis enkripsi paling sederhana yang hanya melibatkan satu kunci rahasia untuk mengacak dan menguraikan informasi. Enkripsi simetris adalah teknik lama dan terkenal. Ini menggunakan kunci rahasia yang bisa berupa angka, kata atau serangkaian huruf acak. Ini dicampur dengan teks pesan untuk mengubah konten dengan cara tertentu. Pengirim dan penerima harus mengetahui kunci rahasia yang digunakan untuk mengenkripsi dan mendekripsi semua pesan.

### 2.2.2 Kriptografi Asimetris

Enkripsi asimetris juga dikenal sebagai kriptografi kunci publik, yang merupakan metode yang relatif baru, dibandingkan dengan enkripsi simetris. Enkripsi asimetris menggunakan dua kunci untuk mengenkripsi teks biasa. Kunci rahasia dipertukarkan melalui Internet atau jaringan besar. Ini memastikan bahwa orang jahat tidak menyalahgunakan kunci. Penting untuk dicatat bahwa siapa pun dengan kunci rahasia dapat mendekripsi pesan dan inilah sebabnya enkripsi asimetris menggunakan dua kunci terkait untuk meningkatkan keamanan. Kunci publik tersedia secara bebas untuk siapa saja yang mungkin ingin mengirim Anda pesan. Kunci pribadi kedua dirahasiakan sehingga Anda hanya bisa tahu.

Pesan yang dienkripsi menggunakan kunci publik hanya dapat didekrip menggunakan kunci privat, sementara juga, pesan yang dienkripsi menggunakan kunci privat dapat didekrip menggunakan kunci publik. Keamanan kunci publik tidak diperlukan karena tersedia untuk umum dan dapat dilewatkan melalui internet. Kunci asimetris memiliki kekuatan yang jauh lebih baik dalam memastikan keamanan informasi yang dikirimkan selama komunikasi. Enkripsi asimetris sebagian besar digunakan dalam saluran komunikasi sehari-hari, terutama melalui Internet. Algoritma enkripsi kunci asimetris populer termasuk ElGamal, RSA, DSA, teknik kurva elips, PKCS.



**Gambar 2.3 Skema kriptografi asimetris**

Sumber: (Information, 2019)

Gambar 2.3 menjelaskan cara kerja dari kriptografi asimetris. Algoritma enkripsi asimetris menggunakan pasangan kunci yang terkait secara matematis untuk enkripsi dan dekripsi; satu adalah kunci publik dan yang lainnya adalah kunci pribadi. Jika kunci publik digunakan untuk enkripsi, kunci pribadi terkait digunakan untuk dekripsi dan jika kunci pribadi digunakan untuk enkripsi, kunci publik terkait digunakan untuk dekripsi (Rouse et al., 2019).

Hanya pengguna atau komputer yang menghasilkan pasangan kunci yang memiliki kunci pribadi. Kunci publik dapat didistribusikan kepada siapa saja yang ingin mengirim data terenkripsi ke pemegang kunci pribadi. Tidak mungkin menentukan kunci privat dengan kunci publik.

Dua peserta dalam alur kerja enkripsi asimetris adalah pengirim dan penerima. Pertama, pengirim memperoleh kunci publik penerima. Kemudian plaintext dienkripsi dengan algoritma enkripsi asimetris menggunakan kunci publik

penerima, membuat ciphertext. Ciphertext kemudian dikirim ke penerima, yang mendekripsi ciphertext dengan kunci pribadinya sehingga ia dapat mengakses plaintext pengirim. Karena fungsi enkripsi satu arah, satu pengirim tidak dapat membaca pesan pengirim lain, walaupun masing-masing memiliki kunci publik penerima.

### 2.3 Horizontal Bit Rotation

Horizontal Bit Rotation (HBR) adalah salah satu dari ilmu kriptografi yang paling awal dan paling sederhana. Plaintext yang diberikan dienkripsi dalam bentuk biner ke dalam ciphertext dengan menggeser setiap bit dari plaintext yang diberikan oleh sebesar  $n$ . Pergeseran bit dilakukan dengan cara memutar bit-bit pada karakter sehingga bertukar posisi. Nilai maksimal yang dapat dilakukan adalah sebesar 7 dan nilai minimal pergeseran adalah sebesar 1.

<b>Indeks</b>		0	1	2	3	4	5	6	7
		0	1	0	0	0	0	1	1
<b>Kunci</b>	3								
<b>Arah</b>	Kanan								
<b>HBR</b>		0	1	1	0	1	0	0	0

**Gambar 2.4 Skema pergeseran bit pada HBR**

Sumber: (Mollin, 2001)

Gambar 2.4 adalah skema pertukaran bit pada tiap karakter. Setiap bit akan digeser posisinya hingga berada di posisi baru sesuai dengan nilai kunci.

## 2.4 Cipher Substitusi

*Cipher* substitusi adalah *cipher* dengan cara mensubstitusikan huruf dengan huruf yang lain sesuai dengan yang ditetapkan. Dalam kriptografi, subtitle *cipher* adalah metode mengenkripsi dimana unit plaintext diganti dengan ciphertext, sesuai dengan sistem tetap; "unit" dapat berupa huruf tunggal (yang paling umum), pasangan huruf, kembar tiga huruf, campuran di atas, dan sebagainya. Jenis-jenis *cipher* substitusi:

### 1. *Cipher* Abjad-Tunggal

*Monogram Monoalphabetic Cipher* adalah *cipher* yang mengganti setiap huruf pada *Plaintext* dengan huruf yang bersesuaian, sehingga apabila terdapat 26 huruf, maka akan terdapat  $26! = 403.291.461.126.605.635.584.000.000$  kemungkinan susunan huruf. Salah satu bentuk *Monogram Monoalphabetic Cipher* adalah *cipher* yang digunakan oleh kaisar Romawi, Julius Caesar (dinamakan juga *Caesar Cipher*), untuk menyandikan pesan yang ia kirim kepada para gubernurnya. Yaitu dengan mengganti (menyulih atau mensubstitusikan) setiap karakter dengan karakter lain dalam susunan abjad (alfabet).

### 2. *Cipher* Abjad-Majemuk

Pensubsitusian setiap huruf menggunakan kunci yang berbeda. *Cipher* abjad-majemuk terdiri dari beberapa *cipher* abjad tunggal yang berbeda-beda. Kebanyakan *cipher* abjad-majemuk adalah *cipher* substitusi periodik. Contoh *Cipher* Abjad-Majemuk adalah *Vigenere Cipher*.

## 2.5 Sistem Informasi

Sistem informasi, seperangkat komponen terintegrasi untuk mengumpulkan, menyimpan, dan memproses data dan untuk menyediakan informasi, pengetahuan, dan produk digital. Perusahaan bisnis dan organisasi lain bergantung pada sistem informasi untuk melaksanakan dan mengelola operasi mereka, berinteraksi dengan pelanggan dan pemasok mereka, dan bersaing di pasar. Sistem informasi digunakan untuk menjalankan rantai pasokan antar organisasi dan pasar elektronik. Misalnya, perusahaan menggunakan sistem informasi untuk memproses akun keuangan, untuk mengelola sumber daya manusia mereka, dan untuk menjangkau pelanggan potensial mereka dengan promosi online. Banyak perusahaan besar dibangun sepenuhnya di sekitar sistem informasi. Ini termasuk eBay, pasar lelang besar; Amazon, mal elektronik yang berkembang dan penyedia layanan cloud computing; Alibaba, e-marketplace bisnis-ke-bisnis; dan Google, perusahaan mesin pencari yang memperoleh sebagian besar pendapatannya dari iklan kata kunci di pencarian Internet. Pemerintah menggunakan sistem informasi untuk menyediakan layanan yang hemat biaya bagi warga negara. Barang digital — seperti buku elektronik, produk video, dan perangkat lunak — dan layanan online, seperti game dan jejaring sosial, dikirimkan dengan sistem informasi. Individu mengandalkan sistem informasi, umumnya berbasis Internet, untuk melakukan banyak kehidupan pribadi mereka: untuk bersosialisasi, belajar, berbelanja, perbankan, dan hiburan (Zwass, 2019).

## 2.6 Algoritma

Algoritma dalam matematika adalah prosedur, deskripsi serangkaian langkah yang dapat digunakan untuk menyelesaikan perhitungan matematis: tetapi mereka jauh lebih umum daripada hari ini. Algoritma digunakan dalam banyak cabang ilmu pengetahuan (dan kehidupan sehari-hari dalam hal ini), tetapi mungkin contoh yang paling umum adalah prosedur langkah-demi-langkah yang digunakan dalam pembagian panjang.

Algoritma adalah metode selangkah demi selangkah untuk menyelesaikan suatu masalah. Ini biasanya digunakan untuk pemrosesan data, perhitungan dan operasi komputer dan matematika terkait lainnya. Algoritma juga digunakan untuk memanipulasi data dengan berbagai cara, seperti memasukkan item data baru, mencari item tertentu atau menyortir item. Algoritma adalah serangkaian instruksi terperinci untuk melakukan operasi atau memecahkan masalah. Dalam pendekatan non-teknis, kami menggunakan algoritma dalam tugas sehari-hari, seperti resep untuk membuat kue atau buku pegangan *do-it-yourself*.

Secara teknis, komputer menggunakan algoritma untuk mendaftar instruksi terperinci untuk melakukan operasi. Misalnya, untuk menghitung gaji karyawan, komputer menggunakan algoritma. Untuk menyelesaikan tugas ini, data yang sesuai harus dimasukkan ke dalam sistem. Dalam hal efisiensi, berbagai algoritma dapat menyelesaikan operasi atau penyelesaian masalah dengan mudah dan cepat.



## 2.7 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang memungkinkan pengembang menentukan, memvisualisasikan, membuat, dan mendokumentasikan artefak sistem perangkat lunak (Technopedia, 2019). Dengan demikian, UML membuat artefak ini dapat diskalakan, aman, dan kuat dalam eksekusi. UML adalah aspek penting yang terlibat dalam pengembangan perangkat lunak berorientasi objek. Ini menggunakan notasi grafis untuk membuat model visual dari sistem perangkat lunak. Arsitektur UML didasarkan pada fasilitas meta-objek, yang mendefinisikan dasar untuk membuat bahasa pemodelan. Mereka cukup tepat untuk menghasilkan seluruh aplikasi. UML yang sepenuhnya dapat dieksekusi dapat digunakan untuk berbagai platform menggunakan teknologi yang berbeda dan dapat digunakan dengan semua proses sepanjang siklus pengembangan perangkat lunak. UML dirancang untuk memungkinkan pengguna mengembangkan bahasa pemodelan visual yang ekspresif, siap pakai. Selain itu, mendukung konsep pengembangan tingkat tinggi seperti kerangka kerja, pola, dan kolaborasi (Wasserkrug et al., 2019).

Penggunaan model ini bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain diluarnya (Sukmawati & Priyadi, 2019).

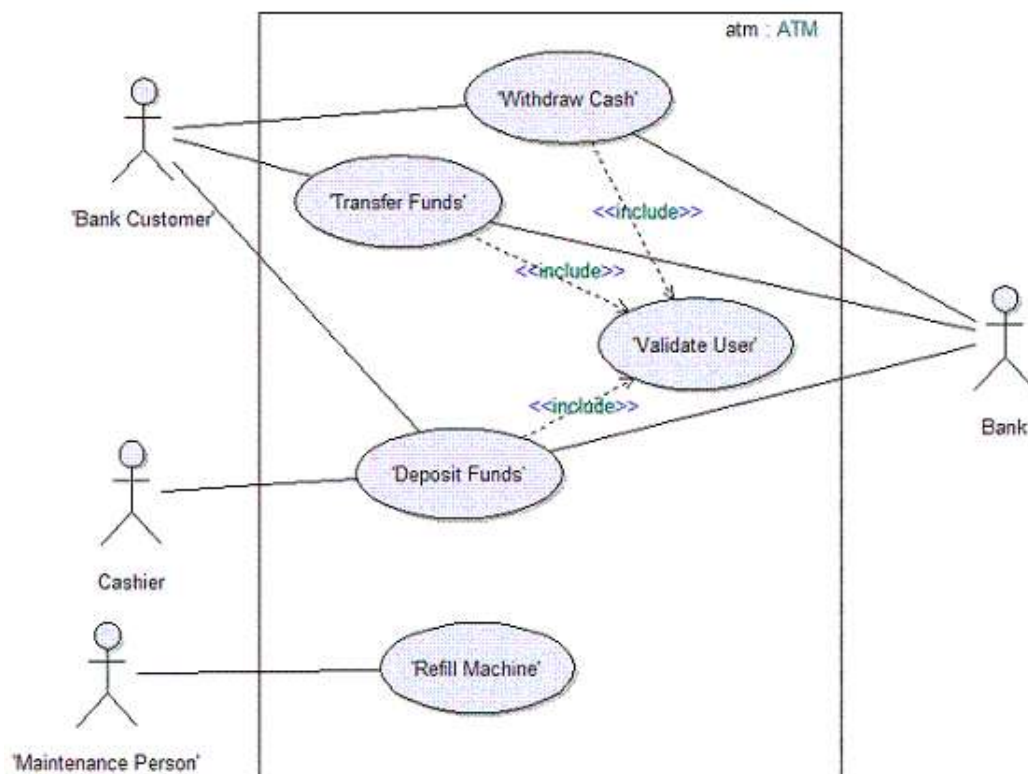
### 2.7.1 Use Case Diagram

*Use Case Diagram* adalah model tentang bagaimana berbagai jenis pengguna berinteraksi dengan sistem untuk memecahkan masalah. Dengan demikian, ini menggambarkan tujuan pengguna, interaksi antara pengguna dan sistem, dan perilaku sistem yang diperlukan dalam memenuhi tujuan-tujuan ini. Model use-case terdiri dari sejumlah elemen model. Elemen model yang paling penting adalah kasus penggunaan, aktor dan hubungan di antara mereka. Diagram use-case digunakan untuk menggambarkan secara grafis subset dari model untuk menyederhanakan komunikasi. Biasanya akan ada beberapa diagram kasus penggunaan yang terkait dengan model yang diberikan, masing-masing menunjukkan subset elemen model yang relevan untuk tujuan tertentu. Elemen model yang sama dapat ditampilkan pada beberapa diagram use-case, tetapi setiap instance harus konsisten. Jika alat digunakan untuk mempertahankan model use-case, kendala konsistensi ini otomatis sehingga setiap perubahan pada elemen model (mengubah nama misalnya) akan secara otomatis tercermin dalam setiap diagram *use-case* yang menunjukkan elemen itu (UTM, 2019).

Model use-case dapat berisi paket yang digunakan untuk menyusun model untuk menyederhanakan analisis, komunikasi, navigasi, pengembangan, pemeliharaan, dan perencanaan. Faktanya, sebagian besar model use-case adalah tekstual, dengan teks yang ditangkap dalam Spesifikasi *Use-Case* yang terkait dengan setiap elemen model use-case. Spesifikasi ini menjelaskan alur peristiwa *use case*. Model *use-case* berfungsi sebagai utas pemersatu sepanjang pengembangan sistem. Ini digunakan sebagai spesifikasi utama dari persyaratan

fungsional untuk sistem, sebagai dasar untuk analisis dan desain, sebagai input untuk perencanaan iterasi, sebagai dasar mendefinisikan kasus uji dan sebagai dasar untuk dokumentasi pengguna. (Kurniawan, 2018).

*Use case diagram* merupakan suatu diagram yang berisi *use case*, *actor*, serta *relationship* diantaranya. *Use Case Diagram* dapat digunakan untuk kebutuhan apa saja yang diperlukan dalam suatu sistem, sehingga sistem dapat digambarkan dengan jelas bagaimana proses dari sistem tersebut, bagaimana cara aktor menggunakan sistem, serta apa saja yang dapat dilakukan pada suatu sistem.











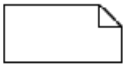
**Gambar 2.5. Use-case Diagram ATM**

Sumber: (Uml-diagrams.org, 2019)

Gambar 2.5 adalah contoh dari penggunaan use-case diagram pada mesin ATM. Use-case memiliki beberapa simbol untuk menyatakan kegiatan dari use-case tersebut. Adapun simbol dari *use case* adalah sebagai berikut:

**Tabel 2.1 Simbol Use Case Diagram**

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		<i>Generalization</i>	Hubungan dimana objek anak berbagi perilaku dan struktur data dari objek yang ada di atasnya.
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi






### 2.7.2 Activity Diagram

*Activity Diagram* (Diagram Aktifitas) menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir

berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir (Ladjamudin, 2017).

*Activity diagram* menurut adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use case*. Diagram ini juga dapat digantikan dengan sejumlah teks.

**Tabel 2.2 Simbol Activity Diagram**



No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk /diawali.
4		<i>Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Sumber: (Kurniawan, 2018)

### 2.7.3 Sequence Diagram

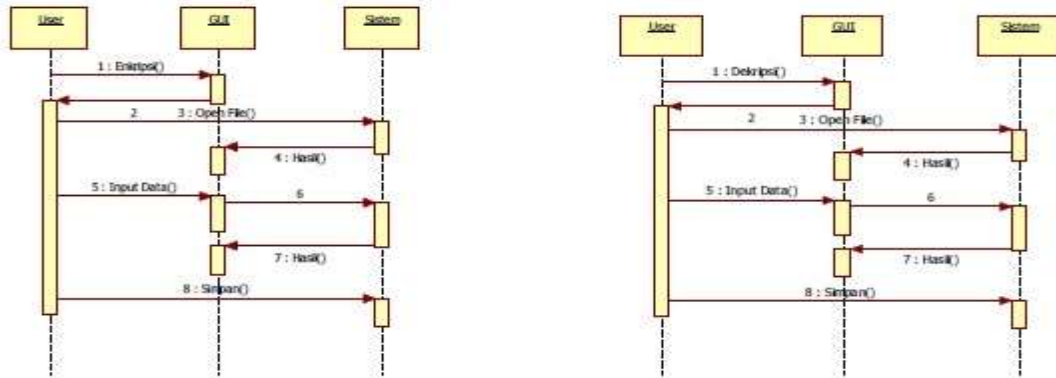
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

**Tabel 2.3 Simbol Sequence Diagram**

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Sumber: (Kurniawan, 2018)

Contoh Sequence Diagram:



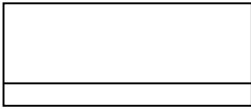

**Gambar 2.6 Contoh Sequence Diagram**

Sumber: (Kurniawan, 2018)

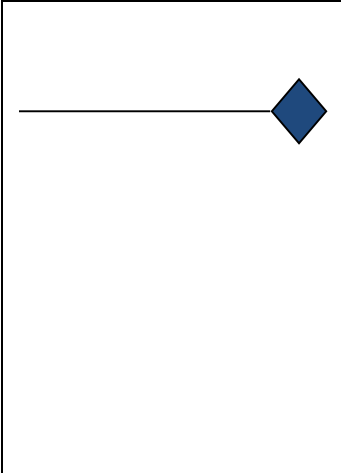
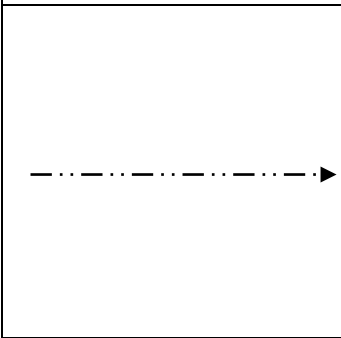
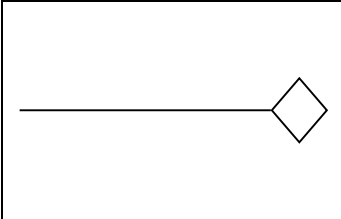
#### 2.7.4 Class Diagram

*Class diagram* digunakan untuk menggambarkan perbedaan yang mendasar antara *class*, hubungan antara *class*, dan di mana *sub-sistem class* tersebut (Jogiyanto, 2016). Simbol yang digunakan dalam *class diagram* adalah sebagai berikut:

**Tabel 2.4 Simbol Class Diagram**

Simbol	Nama	Fungsi
	<i>Class</i>	Menggambarkan <i>Class</i> baru pada diagram.
	<i>Association</i>	Menggambarkan relasi antar asosiasi



	<p><i>Composition</i></p>	<p>Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut.</p>
	<p><i>Dependency</i></p>	<p>Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain.</p>
	<p><i>Aggregation</i></p>	<p><i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi.</p>

Sumber: (Kurniawan, 2018)

### 2.7.5 Flowchart

Flowchart digunakan dalam mendesain dan mendokumentasikan proses atau program sederhana. Seperti jenis diagram lainnya, diagram membantu memvisualisasikan apa yang sedang terjadi dan dengan demikian membantu memahami suatu proses, dan mungkin juga menemukan fitur-fitur yang kurang jelas dalam proses tersebut, seperti kekurangan dan hambatan. Ada berbagai jenis diagram alur: masing-masing jenis memiliki set kotak dan notasi sendiri. Dua jenis kotak yang paling umum dalam diagram alur adalah:

1. Langkah pemrosesan, biasanya disebut aktivitas dan dilambangkan sebagai kotak persegi panjang.
2. Keputusan biasanya dilambangkan sebagai berlian.

Diagram alir digambarkan sebagai "lintas fungsional" ketika bagan dibagi menjadi bagian vertikal atau horizontal yang berbeda, untuk menggambarkan kontrol unit organisasi yang berbeda. Simbol yang muncul di bagian tertentu berada dalam kendali unit organisasi itu. Flowchart lintas fungsional memungkinkan penulis untuk menemukan tanggung jawab untuk melakukan suatu tindakan atau membuat keputusan dengan benar, dan untuk menunjukkan tanggung jawab masing-masing unit organisasi untuk bagian berbeda dari satu proses tunggal.

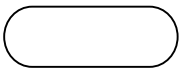
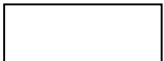
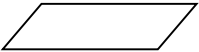
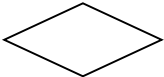
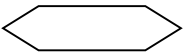
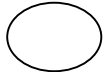

Diagram alir menggambarkan aspek-aspek tertentu dari proses dan biasanya dilengkapi dengan jenis diagram lainnya. Misalnya, Kaoru Ishikawa, mendefinisikan diagram alir sebagai salah satu dari tujuh alat dasar kendali mutu, di sebelah histogram, diagram Pareto, lembar periksa, diagram kontrol, diagram sebab-akibat, dan diagram sebaran. Demikian pula, di UML, notasi pemodelan konsep standar yang digunakan dalam pengembangan perangkat lunak, diagram aktivitas, yang merupakan jenis diagram alur, hanyalah salah satu dari banyak jenis diagram yang berbeda.

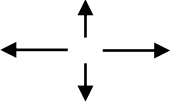

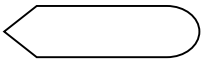


Diagram Nassi-Shneiderman dan Drakon-chart adalah notasi alternatif untuk aliran proses. Nama alternatif umum termasuk diagram alir, diagram alur proses, diagram alur fungsional, peta proses, diagram proses, diagram proses fungsional, model proses bisnis, model proses, diagram alir proses, diagram alur

kerja, diagram alir bisnis. Istilah "diagram alur" dan "diagram alir" digunakan secara bergantian (Nakatsu, 2019).

Struktur grafik yang mendasari diagram alur adalah grafik aliran, yang mengabstraksi jenis simpul, isinya, dan informasi tambahan lainnya. Adapun simbol-simbol flowchart lihat pada tabel sebagai berikut :

**Tabel 2.5 Simbol Flowchart**

NO	SIMBOL	FUNGSI
1.		<b>Terminal</b> , untuk memulai atau mengakhiri suatu program
2.		<b>Proses</b> , suatu simbol yang menunjukkan setiap pengolahan yang dilakukan.
3.		<b>Input-Output</b> , untuk memasukkan menunjukkan hasil dari suatu proses
4.		<b>Decision</b> , suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan
5.		<b>Preparation</b> , suatu symbol yang menyediakan tempat pengolahan
6.		<b>Connector</b> , suatu prosedur penghubung yang akan masuk atau keluar melalui simbol ini dalam lembar yang sama
7.		<b>Off-Page Connector</b> , merupakan symbol masuk atau keluarannya suatu prosedur pada lembaran kertas lainnya

8.		<b>Arus/Flow</b> , dari pada prosedur yang dapat dilakukan atas ke bawah dari bawah ke atas, ke atas dari kiri ke kanan ataupun dari kanan ke kiri
9.		<b>Predefined Process</b> , untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur
10.		Simbol untuk output, yang ditunjukkan ke suatu device, seperti printer, dan sebagainya
11.		Penyimpanan file secara sementara
12.		Menunjukkan input / Output Hardisk (media penyimpanan)

Sumber: (Kurniawan, 2018)

## 2.8 Bahasa Pemrograman

Bahasa pemrograman adalah seperangkat perintah, instruksi, dan penggunaan sintaksis lainnya untuk membuat program perangkat lunak. Bahasa yang digunakan pemrogram untuk menulis kode disebut "bahasa tingkat tinggi." Kode ini dapat dikompilasi menjadi "bahasa tingkat rendah," yang dikenali langsung oleh perangkat keras komputer (Gabbrielli & Martini, 2010).

Bahasa tingkat tinggi dirancang agar mudah dibaca dan dipahami. Ini memungkinkan programmer untuk menulis kode sumber dengan cara alami, menggunakan kata-kata dan simbol yang logis. Misalnya, kata-kata yang dicadangkan seperti fungsi, sementara, jika, dan lainnya digunakan dalam sebagian

besar bahasa pemrograman utama. Simbol seperti <, >, ==, dan != Adalah operator umum. Banyak bahasa tingkat tinggi cukup mirip sehingga pemrogram dapat dengan mudah memahami kode sumber yang ditulis dalam berbagai bahasa.

Contoh bahasa tingkat tinggi termasuk C ++, Java, Perl, dan PHP. Bahasa seperti C ++ dan Java disebut "bahasa yang dikompilasi" karena kode sumber harus dikompilasi terlebih dahulu untuk dapat berjalan. Bahasa seperti Perl dan PHP disebut "bahasa yang ditafsirkan" karena kode sumber dapat dijalankan melalui penerjemah tanpa dikompilasi. Secara umum, bahasa yang dikompilasi digunakan untuk membuat aplikasi perangkat lunak, sementara bahasa yang ditafsirkan digunakan untuk menjalankan skrip, seperti yang digunakan untuk menghasilkan konten untuk situs web dinamis.

Bahasa tingkat rendah meliputi bahasa assembly dan bahasa mesin. Bahasa assembly berisi daftar instruksi dasar dan jauh lebih sulit dibaca daripada bahasa tingkat tinggi. Dalam kasus yang jarang terjadi, seorang programmer dapat memutuskan untuk membuat kode program dasar dalam bahasa assembly untuk memastikan program itu bekerja seefisien mungkin. Assembler dapat digunakan untuk menerjemahkan kode assembly menjadi kode mesin. Kode mesin, atau bahasa mesin, berisi serangkaian kode biner yang dipahami langsung oleh CPU komputer. Tidak perlu dikatakan, bahasa mesin tidak dirancang agar dapat dibaca oleh manusia.

### **2.8.1 Kode Program**

Kode ini hampir seperti menulis paragraf instruksi atau membuat daftar tugas untuk komputer. Tidak seperti kita manusia, daftar tugas dan instruksi yang Anda tulis untuk komputer harus sangat rinci dan ditulis dalam beberapa logika. Dengan kode dan pemrograman, Anda dapat membuat komputer menggambar bentuk yang rumit dan membuat grafik komputer yang kaya, dan kemudian membuat program yang memahami mekanika game dan membantu Anda membuat game yang terasa nyata dengan gravitasi dan tabrakan partikel, dengan program ini yang paling bisa Anda buat semua permainan intens dan imersif. Dengan kode dan pemrograman, Anda dapat membuat dan mengirim konten di seluruh dunia dengan blog Anda dan situs web pribadi dan gaya blog Anda untuk memenuhi gaya Anda. Anda dapat membangun solusi bisnis yang digerakkan oleh teknologi dan menjangkau pelanggan yang lebih luas serta memenuhi kebutuhan yang lebih luas. Selanjutnya, dengan kode dan pemrograman, Anda dapat membuat aplikasi rumah pintar, seperti pengumpan hewan peliharaan otomatis, cermin pintar atau bahkan membuat robot yang dapat membantu menyelesaikan tugas-tugas rumah tangga dan menjadi asisten virtual Anda untuk berbicara dan memahami Anda. Berbeda dengan apa yang dipikirkan banyak orang, ada banyak seni yang terlibat dalam teknik komputer dan ilmu komputer. Anda mungkin tertarik dengan Apa itu pemrograman? posting blog untuk tahu lebih banyak (Gabbrielli & Martini, 2010).

### 2.8.2 Intepreter

Interpreter adalah program komputer yang digunakan untuk secara langsung menjalankan instruksi program yang ditulis menggunakan salah satu dari banyak bahasa pemrograman tingkat tinggi. Interpreter mengubah program tingkat tinggi menjadi bahasa perantara yang kemudian dieksekusi, atau bisa menguraikan kode sumber tingkat tinggi dan kemudian melakukan perintah secara langsung, yang dilakukan baris demi baris atau pernyataan dengan pernyataan.

Bahasa pemrograman diimplementasikan dalam dua cara: interpretasi dan kompilasi. Seperti namanya, seorang juru bahasa mengubah atau menafsirkan kode pemrograman tingkat tinggi menjadi kode yang dapat dipahami oleh mesin (kode mesin) atau menjadi bahasa perantara yang dapat dengan mudah dieksekusi juga. Penerjemah membaca setiap pernyataan kode dan kemudian mengonversi atau mengeksekusi secara langsung. Sebaliknya, assembler atau kompilator mengubah kode sumber tingkat tinggi menjadi kode asli (dikompilasi) yang dapat dieksekusi langsung oleh sistem operasi (Gabbrielli & Martini, 2010).

Dalam kebanyakan kasus, kompilator lebih disukai karena outputnya berjalan jauh lebih cepat dibandingkan dengan interpretasi baris demi baris. Namun, karena interpretasi terjadi per baris atau pernyataan, itu dapat dihentikan di tengah eksekusi untuk memungkinkan modifikasi kode atau debugging. Keduanya memiliki kelebihan dan kekurangan masing-masing dan tidak saling eksklusif; ini berarti bahwa mereka dapat digunakan bersamaan karena sebagian besar lingkungan pengembangan terintegrasi menggunakan kompilasi dan terjemahan untuk beberapa bahasa tingkat tinggi.

Karena penerjemah membaca dan kemudian mengeksekusi kode dalam satu proses tunggal, itu sangat berguna untuk skrip dan program kecil lainnya. Karena itu, biasanya diinstal pada server Web, yang menjalankan banyak skrip yang dapat dieksekusi.

### **2.8.3 Compiler**

Compiler adalah program komputer yang menerjemahkan kode komputer yang ditulis dalam satu bahasa pemrograman (bahasa sumber) ke bahasa lain (bahasa target). Program yang menerjemahkan dari bahasa level rendah ke level lebih tinggi adalah decompiler (Gabbrielli & Martini, 2010). Compiler adalah program perangkat lunak yang mengubah kode pemrograman komputer yang ditulis oleh programmer manusia menjadi kode biner (kode mesin) yang dapat dipahami dan dieksekusi oleh CPU tertentu. Tindakan mengubah kode sumber menjadi kode mesin disebut "kompilasi." Ketika semua kode ditransformasikan pada satu waktu sebelum mencapai platform yang menjalankannya, prosesnya disebut kompilasi "time-of-time (AOT)". Banyak bahasa pemrograman terkenal membutuhkan compiler termasuk:

1. Visual Basic.NET
2. Visual C #
3. Jawa
4. Bahasa campuran
5. C / C ++



## 2.9 Visual Basic.Net

Visual Basic.Net merupakan salah satu *tool development Microsoft* yang dapat digunakan untuk membuat aplikasi di lingkungan kerja berbasis sistem operasi *Windows*. Visual Basic.Net menyediakan *tools* bagi para *developer* untuk membangun aplikasi yang berjalan di *.Net Framework*.

Visual basic merupakan turunan bahasa pemrograman BASIC yang menawarkan pengembangan perangkat lunak computer berbasis grafik dengan cepat. Dengan menggunakan bahasa pemrograman VB, para *programmer* dapat membangun aplikasi dengan menggunakan komponen-komponen yang di sediakan VB.

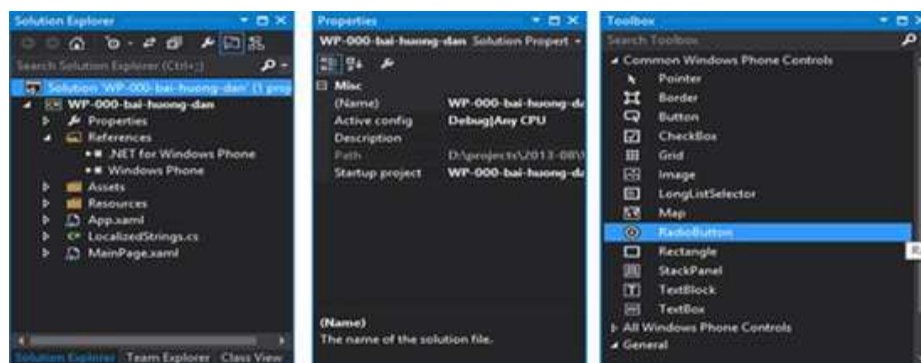
*Microsoft Visual Basic* (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment (IDE) visual* untuk membuat program perangkat lunak berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman (COM), *Visual Basic* merupakan turunan bahasa pemrograman *BASIC* dan menawarkan pengembangan perangkat lunak computer berbasis grafik dengan cepat, beberapa bahasa skrip seperti *Visual Basic for Applications (VBA)* dan *Visual Basic Scripting Edition (VBScript)*, mirip seperti halnya *Visual Basic*, tetapi cara kerjanya yang berbeda.

Para *programmer* dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh *Microsoft Visual Basic*. Program-program yang ditulis dengan *Visual Basic* juga dapat menggunakan *Windows API*, tapi membutuhkan deklarasi fungsi luar tambahan.

Dalam pemrograman untuk bisnis, *Visual Basic* memiliki pangsa pasar yang sangat luas. Dalam sebuah survey yang dilakukan pada tahun 2005, 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk *Visual Basic* yang diikuti oleh C++, JavaScript, dan Java.

Beberapa komponen kerja program *visual basic* 2010 telah ditampilkan sebagai tampilan standard. Masih banyak lagi komponen yang masih tersembunyi sehingga memerlukan perintah tertentu untuk menampilkannya. Kita dapat mengatur komponen di dalam program *visual basic* 2010 sesuai dengan yang kita butuhkan. Berikut ini adalah beberapa komponen kerja dari *visual basic* 2010 adalah.

Toolbox adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain mulai dari tombol *label*, *pointer*, *button*, dan lain-lain. Berikut ini adalah gambaran *toolbox* pada *visual basic* 2010.



**Gambar 2.7 Tampilan Toolbox Visual Basic**

Sumber: (Lee, 2014)

Berikut ini adalah *table* yang berisi nama tombol yang terdapat di dalam *toolbox* beserta fungsinya.

**Tabel 2.6 Toolbox Visual Basic**

Nama tombol	Fungsi
<i>Pointer</i>	Memilih, mengatur ukuran dan memindahkan posisi yang terpasang di bagian <i>form</i> .
<i>Bindingsources</i>	Untuk mengkoneksikan program ke <i>database</i> .
<i>Label</i>	Menampilkan teks, dimana pengguna program tidak bisa mengubah teks tersebut.
<i>Groupbox</i>	Untuk mengelompokkan <i>item</i> yang ada di <i>form</i> .
<i>Checkbox</i>	Membuat kotak periksa, dimana pengguna program dapat memilih sekaligus.
<i>Listbox</i>	Membuat daftar pilihan.
<i>Timer</i>	Membuat kontrol waktu dan interval yang diperlukan.
<i>Image</i>	Menampilkan gambar pada <i>form</i> dalam format <i>bitmap</i> , <i>icon</i> , atau <i>metafile</i> .

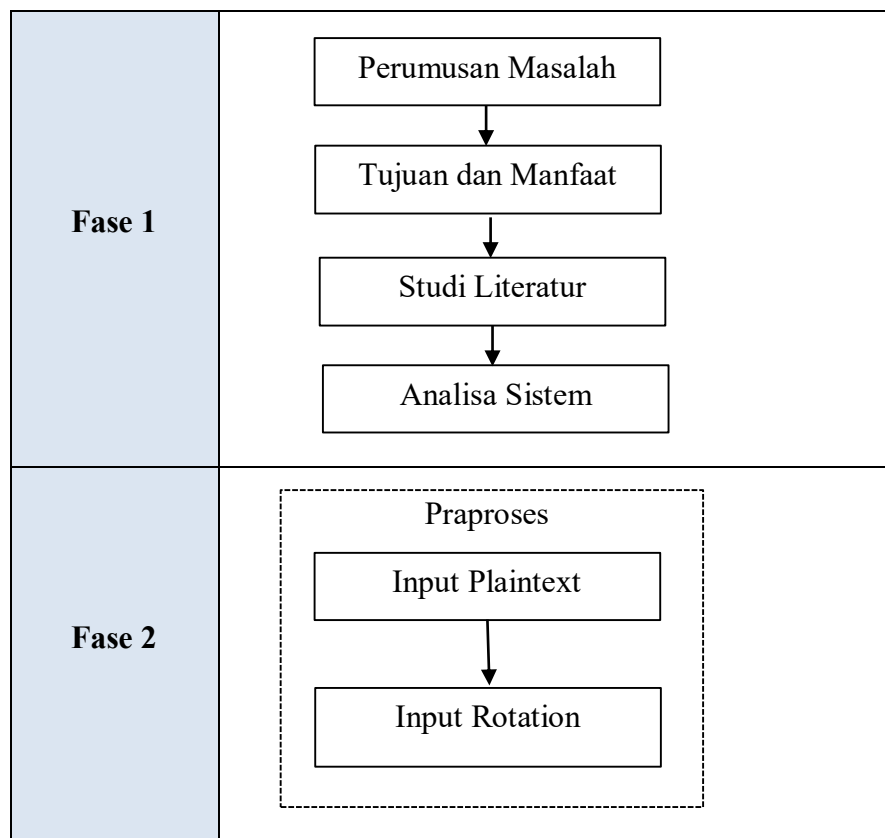
Sumber: (Lee, 2014)

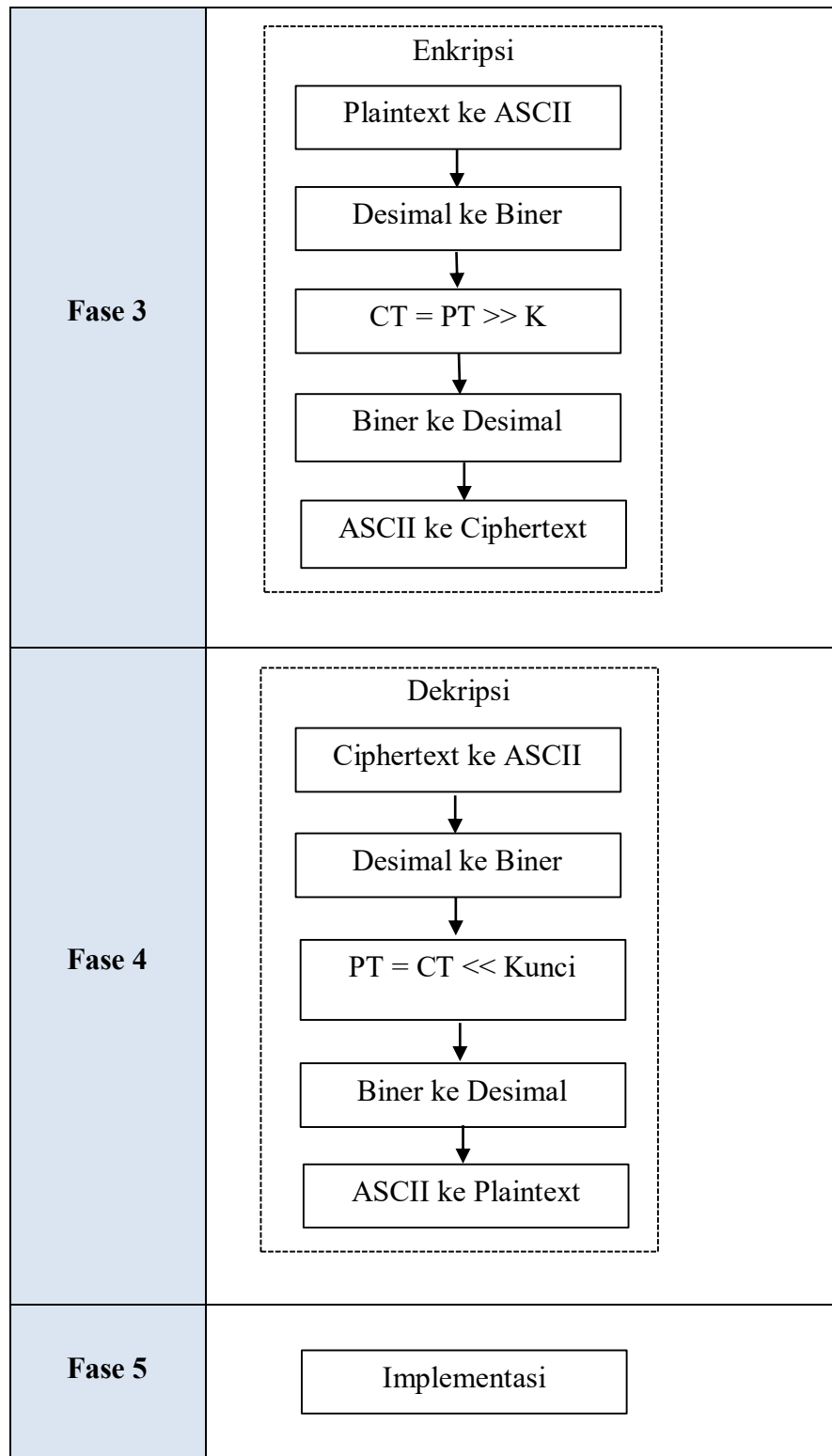
## BAB III

### METODE PENELITIAN

#### 3.1 Kerangka Penelitian

Kerangka penelitian adalah bagan atau pembagian tahapan-tahapanyang dilakukan untuk melakukan klasifikasi penelitian. Kerangka penelitian dilakukan untuk menentukan arah dan gambaran pada pembuatan program aplikasi kriptografi pergeseran bit secara horizontal. Ada lima fase yang dilakukan dalam menyelesaikan peneltian ini. Gambar 3.1 adalah kerangka penelitian yang dilakukan.





**Gambar 3.1 Kerangka Penelitian**

Berikut adalah tahapan penelitian yang dilakukan:

1. Perumusan Masalah

Tahap ini melakukan penentuan topik permasalahan serta bagaimana permasalahan tersebut akan diselesaikan. Identifikasi masalah juga akan dipaparkan di bagian ini.

2. Tujuan dan Manfaat

Tahap ini bertujuan untuk menentukan tujuan dan manfaat dari penelitian ini serta hasil yang diperoleh ketika program aplikasi telah dibuat.

3. Studi Literatur

Studi literatur dilakukan untuk mendapatkan teori-teori tentang ilmu kriptografi tentang pergeseran bit secara horizontal. Studi ini dapat dilakukan melalui buku dan informasi yang ada di internet.

4. Analisa Sistem

Bagian ini menjelaskan proses analisa permasalahan dan bagaimana permasalahan dapat diselesaikan dengan baik. Analisa akan menentukan implementasi dari rancangan yang akan telah ditentukan.

5. Praproses

Tahap ini bertujuan untuk menyiapkan data yang akan diinputkan ke dalam sistem. Data yang disediakan adalah plaintext dan jumlah pergeseran.

6. Enkripsi

Melakukan proses enkripsi dari data yang telah diolah pada praproses. Enkripsi dilakukan berdasarkan jumlah pergeseran bit yang telah ditentukan pada praproses.

### 7. Dekripsi

Melakukan proses dekripsi dari data yang telah menjadi ciphertext yang sebelumnya sudah mengalami proses enkripsi.

### 8. Implementasi

Bagian ini dilakukan pengujian kebenaran hasil program aplikasi. Hasil akan dicocokkan kepada perhitungan manual berdasarkan perhitungan pergeseran bit.

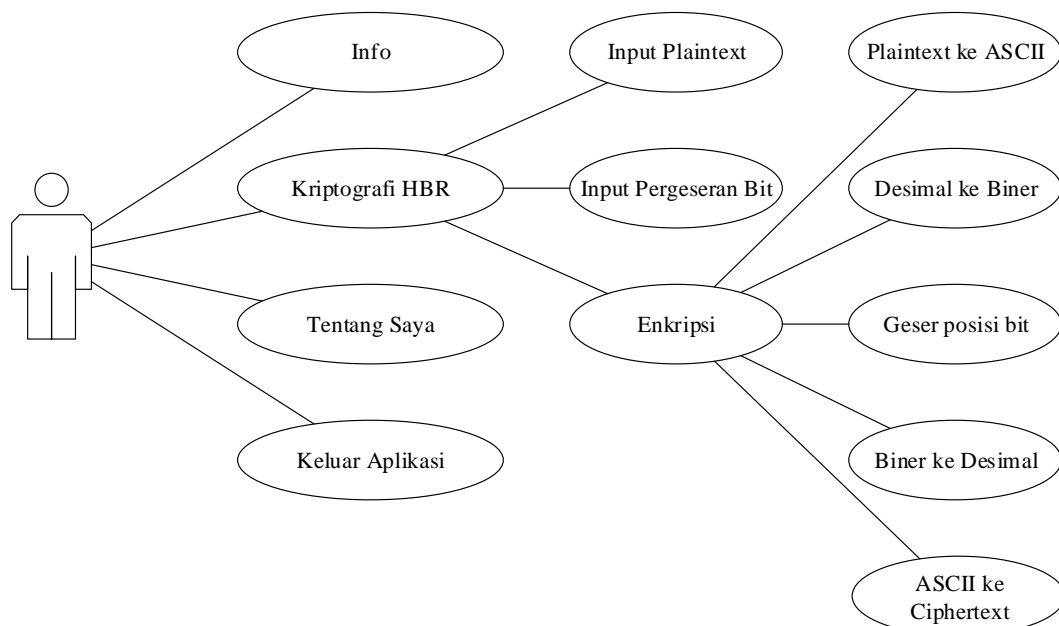
## 3.2 Perancangan Penelitian

Perancangan penelitian disusun agar dapat menentukan hasil implementasi yang berupa program aplikasi. Aplikasi tersebut dibuat dengan menggunakan Microsoft Visual Basic.Net 2010. Perancangan penelitian dijelaskan dalam bentuk *Unified Modelling Language (UML)*. Ada dua buah diagram yang terlibat dalam menentukan perancangan, yaitu *Use Case Diagram* dan *Activity Diagram*.

### 3.2.1 Use Case Diagram

Use case adalah daftar tindakan atau langkah-langkah peristiwa yang biasanya mendefinisikan interaksi antara peran aktor dan sistem untuk mencapai tujuan. Use case adalah teknik yang berguna untuk mengidentifikasi, mengklarifikasi, dan mengatur persyaratan sistem. Sebuah use case terdiri dari serangkaian kemungkinan interaksi antara sistem dan pengguna yang mendefinisikan fitur yang akan diimplementasikan dan resolusi dari setiap kesalahan yang mungkin terjadi.

*Use Case* adalah gambaran fungsi dari sebuah sistem dari segi pengguna. Diagram ini bekerja dengan cara menjelaskan hubungan dan interaksi antar pengguna dengan sistem melalui sebuah alur yang menjelaskan bagaimana sistem tersebut digunakan. Gambar 3.2 adalah perancangan *Use Case* untuk enkripsi program aplikasi pergeseran bit.

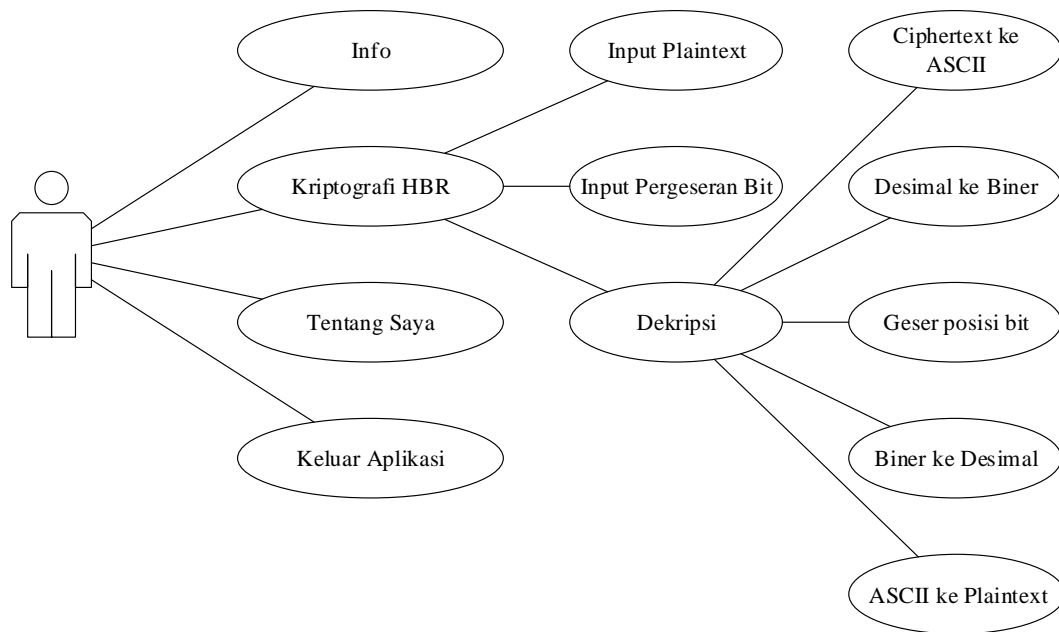


**Gambar 3.2 Use Case Diagram Enkripsi**

Pada alur enkripsi dapat dilihat bahwa proses enkripsi terjadi setelah plaintext dan kunci telah diinputkan. Enkripsi akan mengubah setiap deretan plaintext menjadi karakter pada tabel ASCII sehingga berbentuk bilangan desimal. Bilangan ini kemudian akan dikonversi kembali ke bilangan biner. Bilangan biner akan dilakukan proses rotasi sebesar kunci yang telah ditentukan. Hasil pergeseran akan dikonversi kembali menjadi desimal yang kemudian akan ditransformasikan



kembali ke dalam bentuk ASCII. Deretan ASCII ini akan membentuk sebuah untaian kata yang merupakan ciphertext yang dihasilkan.

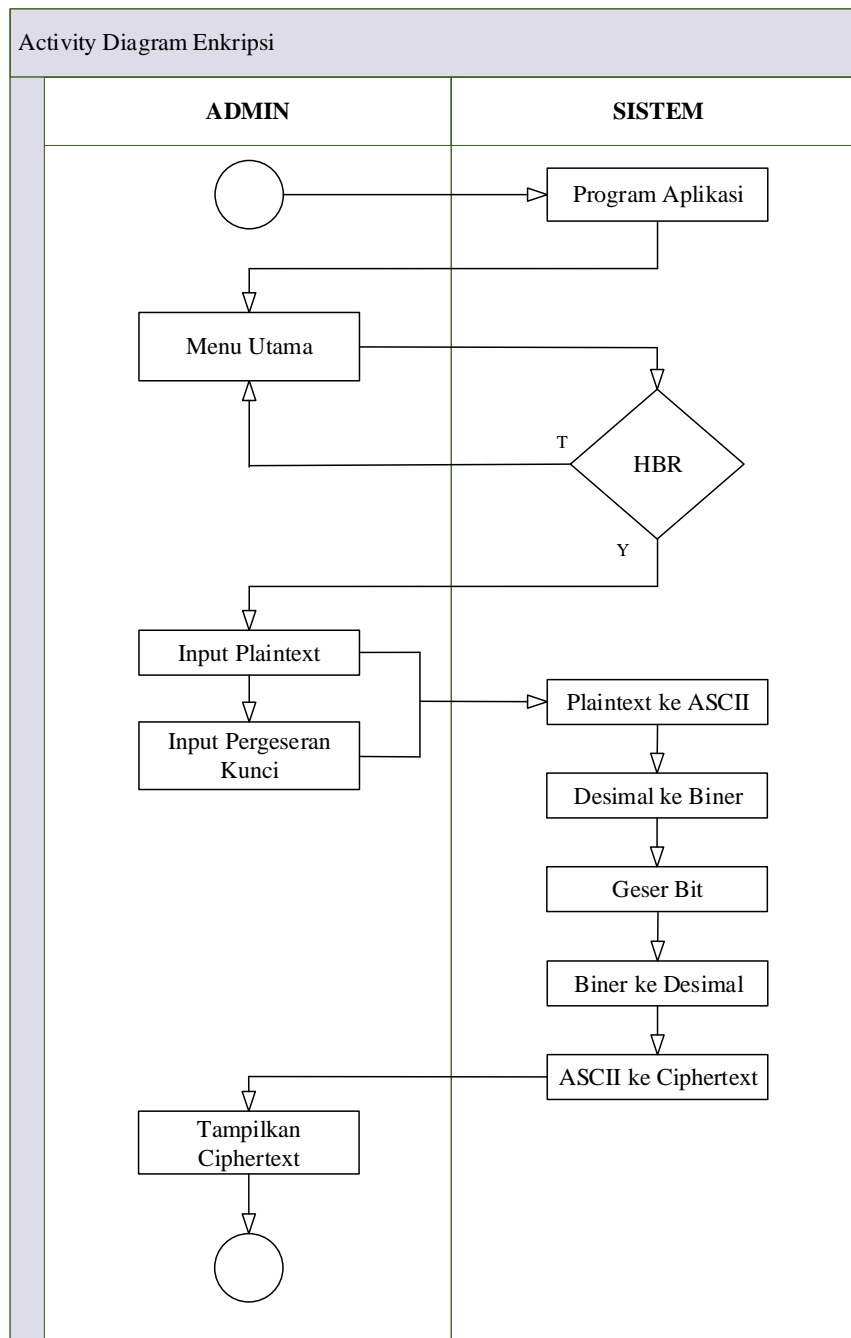


**Gambar 3.3 Use Case Diagram Dekripsi**

Pada alur dekripsi dapat dilihat bahwa proses dekripsi tidak memiliki teknik yang begitu berbeda dengan proses enkripsi. Ciphertext akan diubah menjadi karakter pada tabel ASCII sehingga berbentuk bilangan desimal. Bilangan ini kemudian akan dikonversi kembali ke bilangan biner. Bilangan biner akan dilakukan proses rotasi sebesar kunci yang telah ditentukan. Hasil pergeseran akan dikonversi kembali menjadi desimal yang kemudian akan ditransformasikan kembali ke dalam bentuk ASCII. Deretan ASCII ini akan membentuk sebuah untaian kata yang merupakan plaintext dari ciphertext sebelumnya

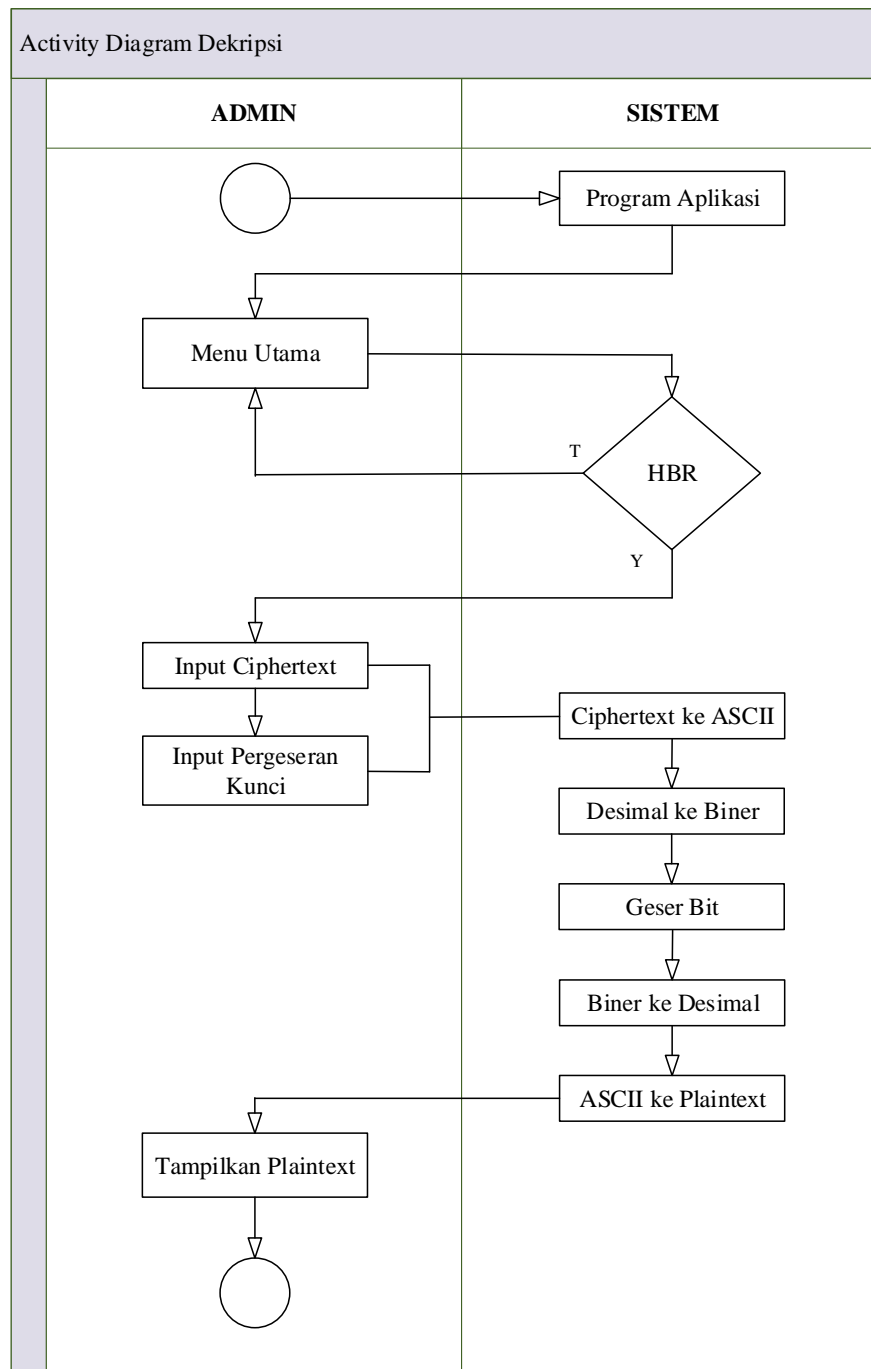
### 3.2.2 Activity Diagram

*Activity diagram* akan menggambarkan alur aktifitas dari sistem. Gambar 3.4 akan menjelaskan *Activity diagram* dari algoritma HBR pada proses enkripsi.



**Gambar 3.4 Activity Diagram Enkripsi**

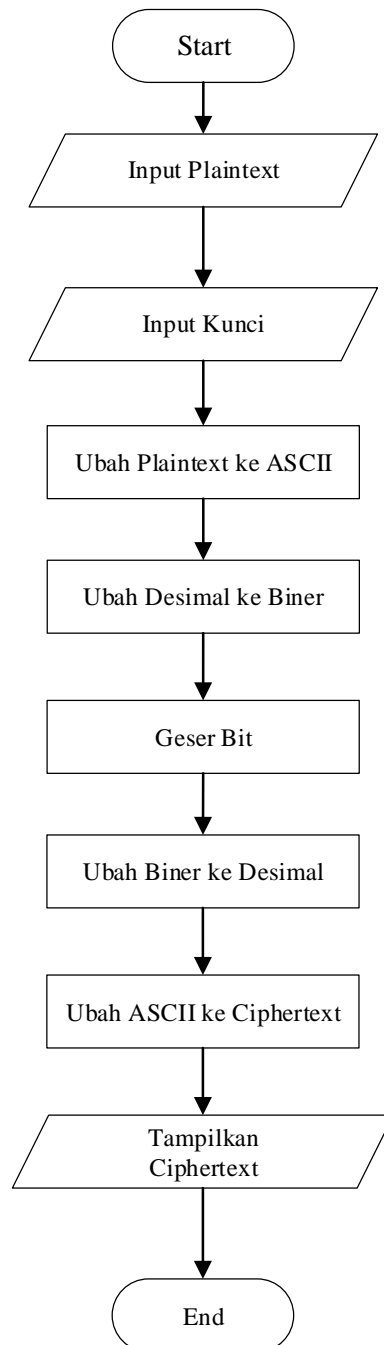
Gambar 3.5 akan menjelaskan *Activity diagram* dari algoritma HBR pada proses dekripsi.



**Gambar 3.5 Activity Diagram Dekripsi**

### 3.2.3 Flowchart Enkripsi

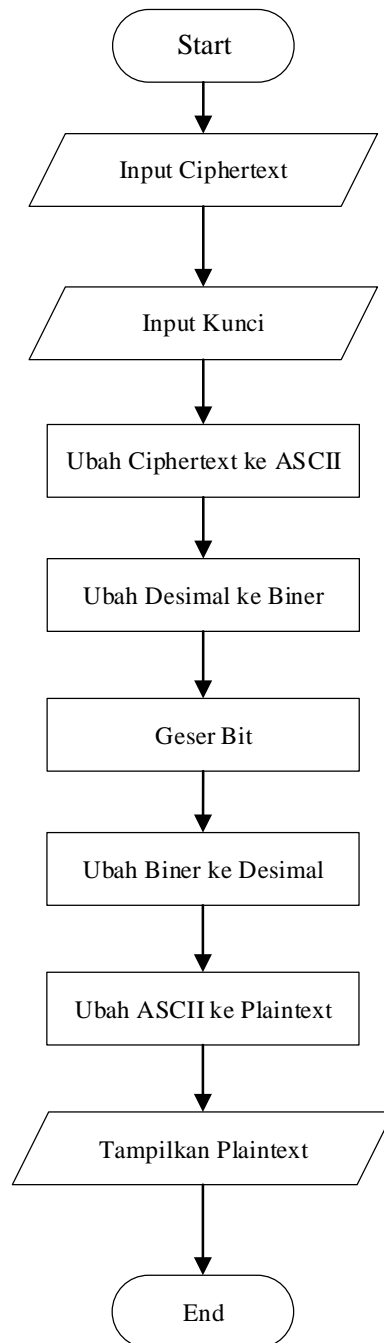
*Flowchart* enkripsi akan menerangkan alur proses enkripsi dengan algoritma HBR. Ini enkripsi dapat dilihat pada gambar 3.6.



**Gambar 3.6 Flowchart enkripsi algoritma HBR**

### 3.2.4 Flowchart Dekripsi

*Flowchart* dekripsi akan menjelaskan alur dari proses dekripsi dengan algoritma HBR. Flowchart ini dapat dilihat pada gambar 3.7.



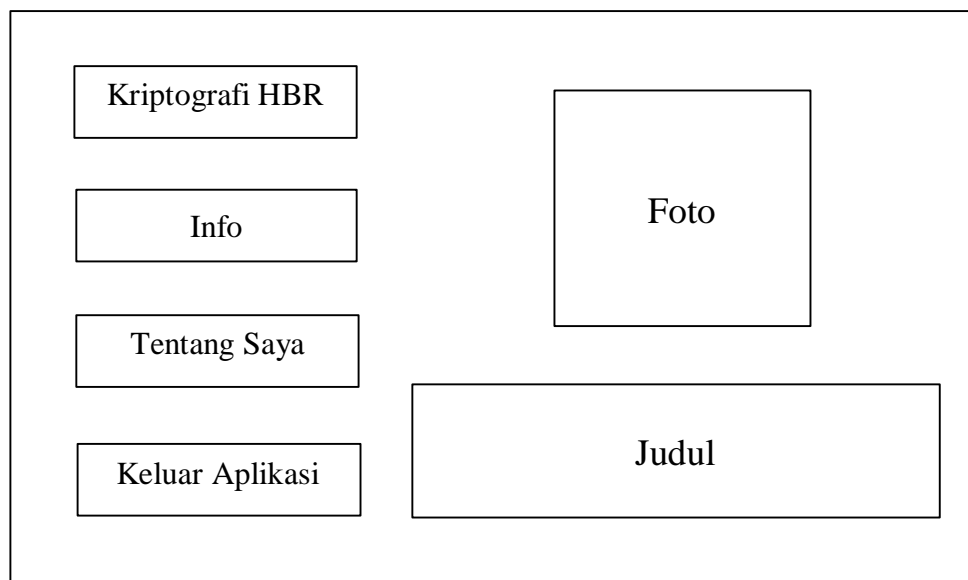
**Gambar 3.7** Flowchart dekripsi algoritma HBR

### 3.3 Desain Interface

Desain *interface* adalah perancangan program aplikasi yang akan dibuat. Kode program akan dibuat menggunakan Microsoft Visual Basic.Net 2010. Desain *interface* ini terbagi menjadi beberapa sub-menu dan memiliki sebuah menu utama yang berfungsi sebagai menjalankan program utama. Beberapa desain berikut ini merupakan desain *interface* dari form pada pembuatan program aplikasi algoritma HBR.

#### 3.3.1 Desain Menu Utama

Desain menu utama adalah rancangan tampilan untuk yang pertama sekali muncul pada saat program aplikasi berjalan. Gambar 3.8 adalah hasil desain Menu Utama.



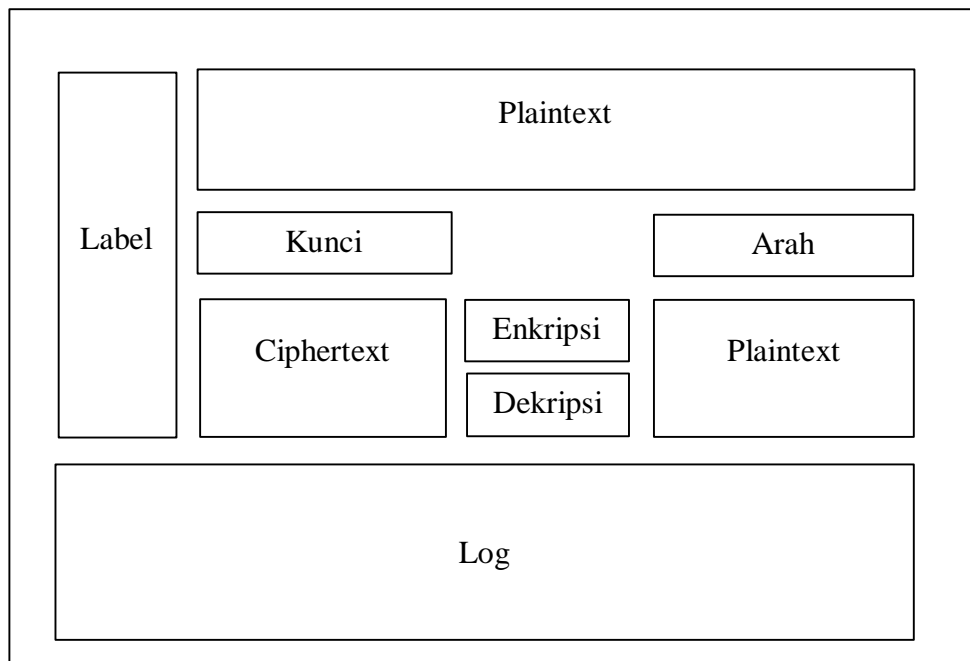
**Gambar 3.8 Desain Menu Utama**

Menu ini memiliki berapa komponen antara lain:

1. Kriptografi HBR
2. Info
3. Tentang Saya
4. Keluar
5. Foto
6. Judul

### 3.3.2 Desain Kriptografi HBR

Desain ini perancangan program aplikasi utama karena menjalankan algoritma HBR. Desain ini terdiri dari input, proses, output dan riwayat perhitungan. Gambar 3.9 adalah tampilan desain algoritma HBR.



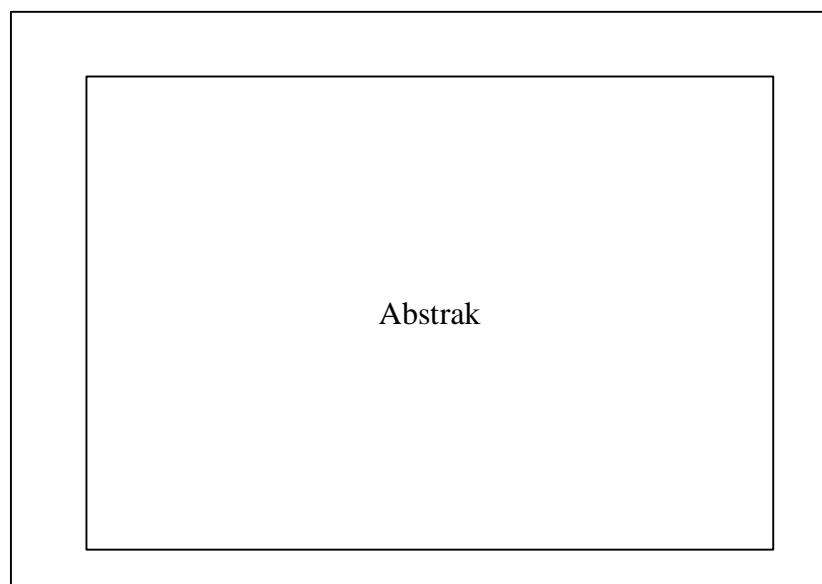
**Gambar 3.9 Desain Kriptografi HBR**

Menu algoritma Vigenere Cipher memiliki beberapa bagian antara lain:

1. Plaintext
2. Ciphertext
3. Kunci
4. Arah
5. Tombol Enkripsi
6. Tombol Dekripsi
7. Log

### 3.3.3 Desain Info

Menu ini menampilkan tentang abstrak yang penulis simpulkan. Desain ini memiliki sebuah objek label untuk menampilkan tulisan abstrak tersebut. Gambar 3.10 adalah hasil perancangan desain Info.

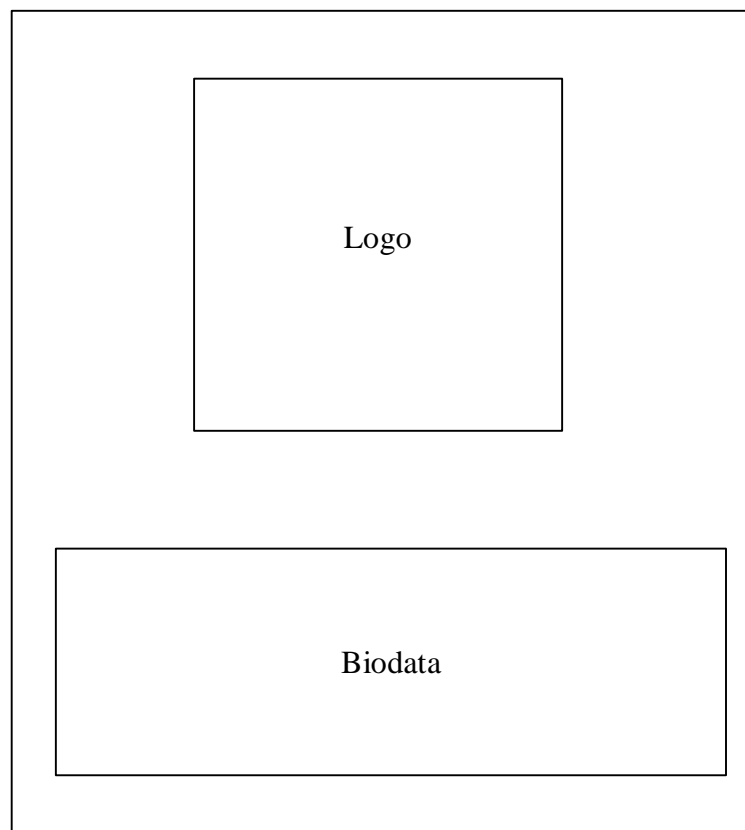


**Gambar 3.10 Desain Info**



### 3.3.4 Desain Tentang Saya

Menu ini menampilkan keterangan tentang penulis. Menu ini terdiri dari logo Universitas Pembangunan Panca Budi dan beberapa keterangan singkat. Menu ini terdiri dari dua objek, yaitu logo dan informasi. Gambar 3.11 adalah hasil desain Tentang Saya.



**Gambar 3.11 Desain Tentang Saya**

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Hasil studi biasanya merujuk pada jawaban langsung untuk pertanyaan penelitian yang dihasilkan dari penelitian. Hasil ini akan didiskusikan pada bagian terakhir untuk mengetahui apakah sistem berjalan dengan baik atau tidak. Perancangan yang telah dilakukan pada bab sebelumnya akan diimplementasikan pada bab ini. Ini menandakan program aplikasi HBR harus dapat diciptakan berdasarkan kaidah-kaidah yang telah ditentukan pada pembahasan sebelumnya. Untuk mendukung pembuatan program, sangat dibutuhkan sekali suatu sistem yang dapat menjalankan program aplikasi tersebut. Ada beberapa kriteria dan syarat yang harus dipenuhi dalam menyediakan suatu sistem.

#### **4.1 Spesifikasi Sistem**

Spesifikasi sistem menjelaskan persyaratan operasional dan kinerja suatu sistem, seperti komputer. Ini dianggap sebagai dokumen tingkat tinggi yang menentukan fungsi global. Spesifikasi sistem membantu untuk menentukan pedoman operasional dan kinerja untuk suatu sistem. Ini dapat menguraikan bagaimana sistem diharapkan untuk melakukan, dan apa yang mungkin termasuk. Spesifikasi utama dapat mencakup definisi antarmuka, aturan desain dokumen, dan area fungsional. Spesifikasi sistem dapat diuraikan selama proses evaluasi dan disepakati selama proses pengujian.

#### 4.1.1 Spesifikasi Perangkat Keras

Penerapan algoritma HBR pada metode kriptografi substitusi membutuhkan perangkat keras untuk menjalankan algoritma tersebut. Perangkat keras ini sebagai sarana pendukung utama. Tabel 4.1 adalah spesifikasi perangkat keras yang dibutuhkan dalam melakukan penelitian ini.

**Tabel 4.1 Spesifikasi perangkat keras**

No.	Nama Komponen	Spesifikasi
1	Processor	Intel Core i5 2.4 GHz
2	RAM	8192 MB
3	Storage	500 GB
4	Display	14 inches

#### 4.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak merupakan kode program untuk menjalankan perangkat keras tersebut. Setiap komputer membutuhkan sistem operasi dan program aplikasi pendukung agar dapat digunakan sesuai dengan tujuan. Kebutuhan perangkat lunak ini bertujuan sebagai sarana non-fisik untuk menghasilkan program aplikasi HBR. Tabel 4.2 adalah spesifikasi perangkat lunak yang digunakan pada penelitian ini.

**Tabel 4.2 Spesifikasi perangkat lunak**

No.	Nama Komponen	Spesifikasi
1	Sistem Operasi	Windows 10 64 Bit
2	IDE Pemrograman	Microsoft Visual Basic.NET 2010
3	Tangkap Gambar	Snipping Tool
4	Data Editor	Microsoft Excel

## 4.2 Implementasi Interface

Menerapkan *interface* berarti benar-benar melakukan pekerjaan untuk mengubah perancangan menjadi program aplikasi. Proses implementasi memiliki langkah-langkah umum dalam menciptakan hasil yang baik. Agar proses implementasi berhasil, beberapa tinjauan dalam membuat program aplikasi seperti kode program harus dilaksanakan dengan baik. Uji coba terhadap interface sangat harus dilakukan agar tidak terjadi kesalahan yang fatal pada saat sistem tersebut dijalankan. Sangat dibutuhkan sekali adanya pemeriksaan lebih dalam tentang program aplikasi yang dibuat. Uji coba harus dilakukan terhadap antarmuka dan terhadap hasil output dari program aplikasi dalam menghitung proses enkripsi dan dekripsi pada kriptografi HBR.

### 4.2.1 Implementasi Menu Utama

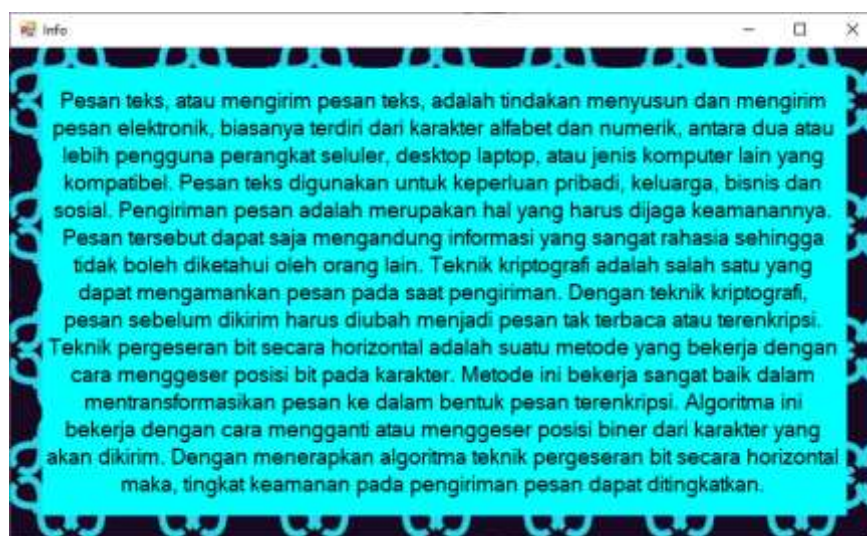
Implementasi Menu Utama adalah tampilan yang pertama sekali muncul pada saat program aplikasi dijalankan. Ada beberapa menu yang akan dimunculkan untuk menentukan pilihan menu berikutnya. Pengguna akan memilih menu mana yang akan mereka tuju. Menu utama bertujuan untuk memberikan pilihan fungsi dari fasilitas-fasilitas yang diberikan pada program aplikasi kriptografi HBR. Menu utama pada penelitian ini terdiri dari tiga buah sub-menu dan satu buah tombol untuk keluar dari aplikasi tersebut. Gambar 4.1 adalah hasil implementasi Menu Utama.



**Gambar 4.1 Implementasi Menu Utama**

#### 4.2.2 Implementasi Info

Halaman info adalah menu yang menampilkan abstrak dari penelitian ini. Halaman ini memiliki sebuah deskripsi tentang cara kerja kriptografi HBR. Gambar 4.2 adalah hasil implementasi dari Info.



**Gambar 4.2 Implementasi Info**

### 4.2.3 Implementasi Tentang Saya

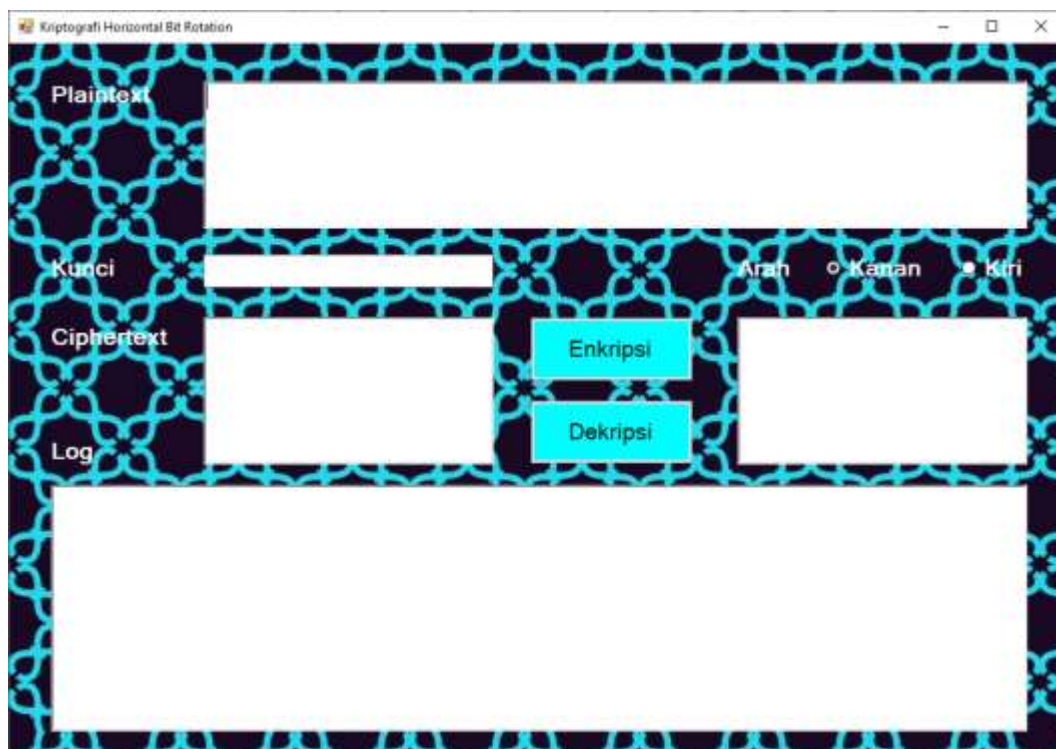
Implementasi Tentang Saya adalah tampilan seputar keterangan mengenai penulis. Halaman ini menampilkan nama, NPM, fakultas, program studi dan universitas. Gambar 4.3 adalah tampilan dari implementasi Tentang Saya.



**Gambar 4.3 Implementasi Tentang Saya**

### 4.2.4 Implementasi Kriptografi HBR

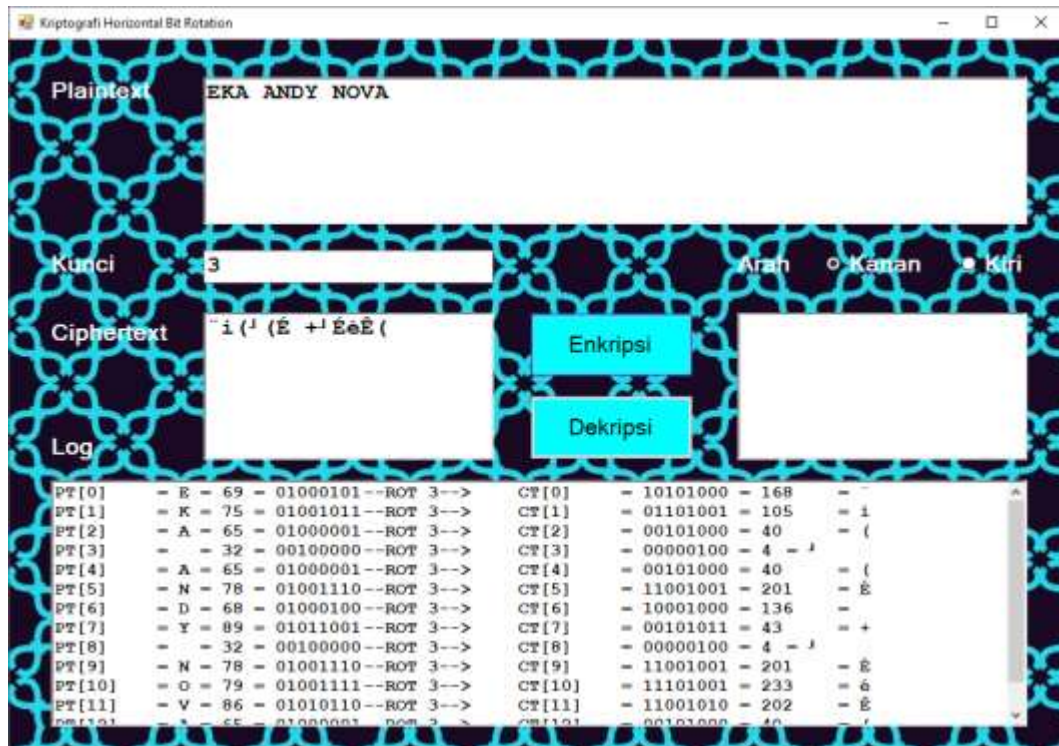
Halaman ini merupakan proses untuk melakukan enkripsi dan dekripsi. Pada halaman ini juga ditampilkan perhitungan lengkap algoritma HBR tersebut. Halaman terdiri dari dua buah plaintext, sebuah kunci dan sebuah ciphertext yang dibentuk dari objek textbox. Sementara untuk proses enkripsi dan dekripsi, halaman ini memiliki beberapa tombol. Gambar 4.4 adalah hasil tampilan dari implementasi Kriptografi HBR.



**Gambar 4.4 Implementasi Kriptografi HBR**

#### **4.2.5 Hasil Perhitungan Kriptografi HBR**

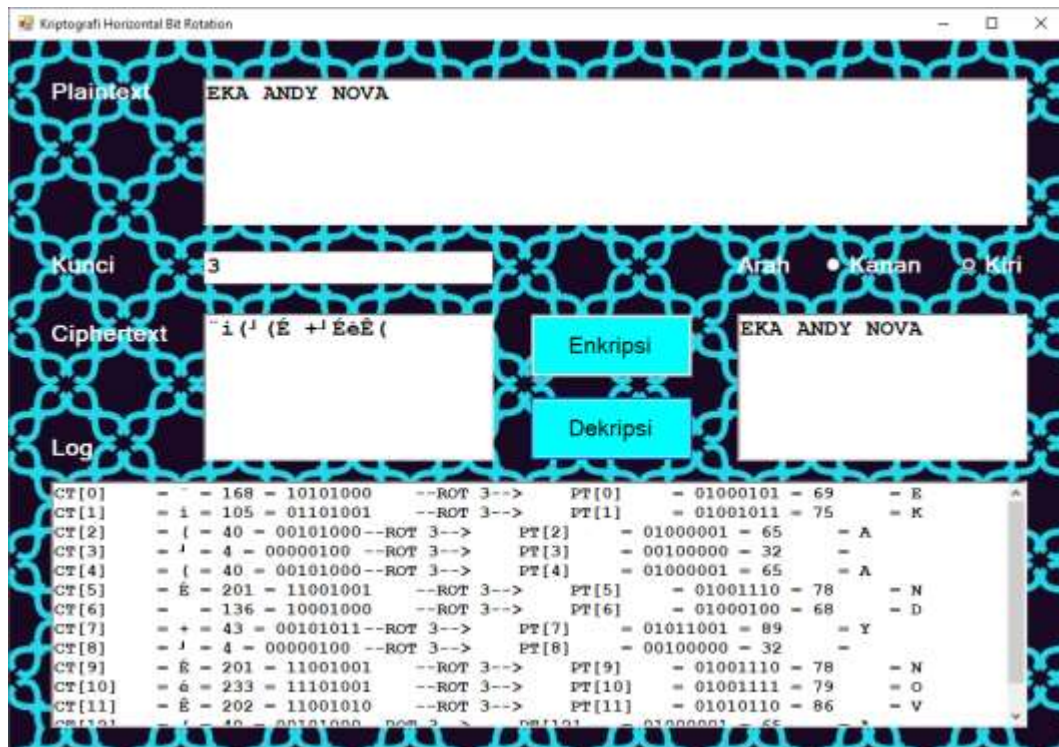
Halaman ini berisi hasil perhitungan algoritma dari proses yang dikerjakan oleh program aplikasi dalam melakukan proses enkripsi dan dekripsi. Plaintext dan Kunci adalah dua bagian yang harus diisi untuk menentukan ciphertext. Kedua nilai dari parameter ini akan diproses sehingga menghasilkan ciphertext. Setiap karakter pada plaintext akan dilakukan pergeseran sebesar kunci untuk mendapatkan ciphertext. Kunci memiliki dua arah, ke kanan dan ke kiri. Ciphertext akan berbeda jika menggunakan kunci yang sama dengan arah yang berbeda. Gambar 4.5 adalah tampilan dari hasil perhitungan proses enkripsi algoritma HBR dengan kunci = 3 dan arah = kanan.



**Gambar 4.5 Implementasi enkripsi algoritma HBR**

Proses dekripsi dilakukan dengan membalikkan arah kunci dari kanan ke kiri atau dari kiri ke kanan. Apabila pada saat proses enkripsi menggunakan kunci = 3 dan arah ke kanan, pada saat dekripsi dapat dilakukan dengan kunci = 5, arah ke kanan atau, kunci = 3, arah ke kiri. Karena sistem pegeseran kunci adalah rotasi, maka kunci = 3 sama dengan kunci  $8 - 3 = 5$ . Proses dekripsi akan melakukan pergeseran bit pada setiap karakter pada sepanjang plaintext. Gambar 4. 6 adalah tampilan dari hasil perhitungan proses dekripsi algoritma HBR dengan kunci = 3 dan arah = kiri atau kunci = 5 dan arah kanan.





Gambar 4.6 Implementasi dekripsi algoritma HBR

### 4.3 Pengujian Perhitungan

Pengujian adalah melakukan uji coba hasil perhitungan proses enkripsi dan dekripsi algoritma HBR. Pengujian dilakukan dengan memberikan input plaintext, kunci dan arah. Hasil program aplikasi harus sesuai dengan hasil perhitungan manual yang diuji pada penjelasan berikut ini. Penjelasan berikut ini adalah penjelasan dan perhitungan lengkap proses dekripsi dan enkripsi pada algoritma HBR.

Plaintext = PERJALANAN INI TERASA SANGAT MENCEKAM

Kunci = 3

Arah = Kanan

### Hasil Enkripsi

PT[0]	= P = 80 = 01010000	--ROT 3-->	CT[0]	= 00001010 = 10	=
PT[1]	= E = 69 = 01000101	--ROT 3-->	CT[1]	= 10101000 = 168	= "
PT[2]	= R = 82 = 01010010	--ROT 3-->	CT[2]	= 01001010 = 74	= J
PT[3]	= J = 74 = 01001010	--ROT 3-->	CT[3]	= 01001001 = 73	= I
PT[4]	= A = 65 = 01000001	--ROT 3-->	CT[4]	= 00101000 = 40	= (
PT[5]	= L = 76 = 01001100	--ROT 3-->	CT[5]	= 10001001 = 137	= %
PT[6]	= A = 65 = 01000001	--ROT 3-->	CT[6]	= 00101000 = 40	= (
PT[7]	= N = 78 = 01001110	--ROT 3-->	CT[7]	= 11001001 = 201	= É
PT[8]	= A = 65 = 01000001	--ROT 3-->	CT[8]	= 00101000 = 40	= (
PT[9]	= N = 78 = 01001110	--ROT 3-->	CT[9]	= 11001001 = 201	= É
PT[10]	= = 32 = 00100000	--ROT 3-->	CT[10]	= 00000100 = 4	= -
PT[11]	= I = 73 = 01001001	--ROT 3-->	CT[11]	= 00101001 = 41	= )
PT[12]	= N = 78 = 01001110	--ROT 3-->	CT[12]	= 11001001 = 201	= É
PT[13]	= I = 73 = 01001001	--ROT 3-->	CT[13]	= 00101001 = 41	= )
PT[14]	= = 32 = 00100000	--ROT 3-->	CT[14]	= 00000100 = 4	= -
PT[15]	= T = 84 = 01010100	--ROT 3-->	CT[15]	= 10001010 = 138	= Š
PT[16]	= E = 69 = 01000101	--ROT 3-->	CT[16]	= 10101000 = 168	= "
PT[17]	= R = 82 = 01010010	--ROT 3-->	CT[17]	= 01001010 = 74	= J
PT[18]	= A = 65 = 01000001	--ROT 3-->	CT[18]	= 00101000 = 40	= (
PT[19]	= S = 83 = 01010011	--ROT 3-->	CT[19]	= 01101010 = 106	= j
PT[20]	= A = 65 = 01000001	--ROT 3-->	CT[20]	= 00101000 = 40	= (
PT[21]	= = 32 = 00100000	--ROT 3-->	CT[21]	= 00000100 = 4	= -
PT[22]	= S = 83 = 01010011	--ROT 3-->	CT[22]	= 01101010 = 106	= j
PT[23]	= A = 65 = 01000001	--ROT 3-->	CT[23]	= 00101000 = 40	= (
PT[24]	= N = 78 = 01001110	--ROT 3-->	CT[24]	= 11001001 = 201	= É
PT[25]	= G = 71 = 01000111	--ROT 3-->	CT[25]	= 11101000 = 232	= è
PT[26]	= A = 65 = 01000001	--ROT 3-->	CT[26]	= 00101000 = 40	= (
PT[27]	= T = 84 = 01010100	--ROT 3-->	CT[27]	= 10001010 = 138	= Š
PT[28]	= = 32 = 00100000	--ROT 3-->	CT[28]	= 00000100 = 4	= -
PT[29]	= M = 77 = 01001101	--ROT 3-->	CT[29]	= 10101001 = 169	= ©
PT[30]	= E = 69 = 01000101	--ROT 3-->	CT[30]	= 10101000 = 168	= "
PT[31]	= N = 78 = 01001110	--ROT 3-->	CT[31]	= 11001001 = 201	= É
PT[32]	= C = 67 = 01000011	--ROT 3-->	CT[32]	= 01101000 = 104	= h
PT[33]	= E = 69 = 01000101	--ROT 3-->	CT[33]	= 10101000 = 168	= "
PT[34]	= K = 75 = 01001011	--ROT 3-->	CT[34]	= 01101001 = 105	= i
PT[35]	= A = 65 = 01000001	--ROT 3-->	CT[35]	= 00101000 = 40	= (
PT[36]	= M = 77 = 01001101	--ROT 3-->	CT[36]	= 10101001 = 169	= ©

Ciphertext = "JI(%o(É(É )É) Š"J(j( j(Éè(Š©·Éh"i(©

Kunci = 3

Arah = Kiri

## Hasil Dekripsi

CT[0]	=	=	10	=	00001010	--ROT 3-->	PT[0]	=	01010000	=	80	=	P	
CT[1]	=	ˆ	=	168	=	10101000	--ROT 3-->	PT[1]	=	01000101	=	69	=	E
CT[2]	=	J	=	74	=	01001010	--ROT 3-->	PT[2]	=	01010010	=	82	=	R
CT[3]	=	I	=	73	=	01001001	--ROT 3-->	PT[3]	=	01001010	=	74	=	J
CT[4]	=	(	=	40	=	00101000	--ROT 3-->	PT[4]	=	01000001	=	65	=	A
CT[5]	=	%	=	137	=	10001001	--ROT 3-->	PT[5]	=	01001100	=	76	=	L
CT[6]	=	(	=	40	=	00101000	--ROT 3-->	PT[6]	=	01000001	=	65	=	A
CT[7]	=	É	=	201	=	11001001	--ROT 3-->	PT[7]	=	01001110	=	78	=	N
CT[8]	=	(	=	40	=	00101000	--ROT 3-->	PT[8]	=	01000001	=	65	=	A
CT[9]	=	É	=	201	=	11001001	--ROT 3-->	PT[9]	=	01001110	=	78	=	N
CT[10]	=	_	=	4	=	00000100	--ROT 3-->	PT[10]	=	00100000	=	32	=	
CT[11]	=	)	=	41	=	00101001	--ROT 3-->	PT[11]	=	01001001	=	73	=	I
CT[12]	=	É	=	201	=	11001001	--ROT 3-->	PT[12]	=	01001110	=	78	=	N
CT[13]	=	)	=	41	=	00101001	--ROT 3-->	PT[13]	=	01001001	=	73	=	I
CT[14]	=	_	=	4	=	00000100	--ROT 3-->	PT[14]	=	00100000	=	32	=	
CT[15]	=	Š	=	138	=	10001010	--ROT 3-->	PT[15]	=	01010100	=	84	=	T
CT[16]	=	ˆ	=	168	=	10101000	--ROT 3-->	PT[16]	=	01000101	=	69	=	E
CT[17]	=	J	=	74	=	01001010	--ROT 3-->	PT[17]	=	01010010	=	82	=	R
CT[18]	=	(	=	40	=	00101000	--ROT 3-->	PT[18]	=	01000001	=	65	=	A
CT[19]	=	j	=	106	=	01101010	--ROT 3-->	PT[19]	=	01010011	=	83	=	S
CT[20]	=	(	=	40	=	00101000	--ROT 3-->	PT[20]	=	01000001	=	65	=	A
CT[21]	=	_	=	4	=	00000100	--ROT 3-->	PT[21]	=	00100000	=	32	=	
CT[22]	=	j	=	106	=	01101010	--ROT 3-->	PT[22]	=	01010011	=	83	=	S
CT[23]	=	(	=	40	=	00101000	--ROT 3-->	PT[23]	=	01000001	=	65	=	A
CT[24]	=	É	=	201	=	11001001	--ROT 3-->	PT[24]	=	01001110	=	78	=	N
CT[25]	=	è	=	232	=	11101000	--ROT 3-->	PT[25]	=	01000111	=	71	=	G
CT[26]	=	(	=	40	=	00101000	--ROT 3-->	PT[26]	=	01000001	=	65	=	A
CT[27]	=	Š	=	138	=	10001010	--ROT 3-->	PT[27]	=	01010100	=	84	=	T
CT[28]	=	_	=	4	=	00000100	--ROT 3-->	PT[28]	=	00100000	=	32	=	
CT[29]	=	©	=	169	=	10101001	--ROT 3-->	PT[29]	=	01001101	=	77	=	M
CT[30]	=	ˆ	=	168	=	10101000	--ROT 3-->	PT[30]	=	01000101	=	69	=	E
CT[31]	=	É	=	201	=	11001001	--ROT 3-->	PT[31]	=	01001110	=	78	=	N
CT[32]	=	h	=	104	=	01101000	--ROT 3-->	PT[32]	=	01000011	=	67	=	C
CT[33]	=	ˆ	=	168	=	10101000	--ROT 3-->	PT[33]	=	01000101	=	69	=	E
CT[34]	=	i	=	105	=	01101001	--ROT 3-->	PT[34]	=	01001011	=	75	=	K
CT[35]	=	(	=	40	=	00101000	--ROT 3-->	PT[35]	=	01000001	=	65	=	A
CT[36]	=	©	=	169	=	10101001	--ROT 3-->	PT[36]	=	01001101	=	77	=	M

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Penulis dapat menarik beberapa kesimpulan berdasarkan hasil pengujian yang dilakukan setelah melakukan penelitian. Adapun kesimpulan yang diperoleh adalah antara lain:

1. Algoritma HBR bekerja dengan cara mengubah posisi bit dari karakter yang telah dikonversi ke dalam bilangan biner.
2. Panjang kunci merupakan pergeseran bit pada karakter ASCII.
3. Proses pergeseran bit hanya dapat dilakukan sebanyak 1 hingga 7 bit.

#### **5.2 Saran**

Penelitian juga memiliki kekurangan. Terdapat beberapa saran yang dapat penulis kemukakan untuk meningkatkan kualitas penelitian ini. Adapun saran tersebut adalah antara lain:

1. Hendaknya algoritma HBR dikombinasikan dengan algoritma lain agar mengurangi peluang pembobolan ciphertext.
2. HBR akan lebih baik apabila jika meningkatkan keamanan kunci sehingga tidak kunci tidak hanya ditentukan dengan angka 1 hingga 7 bit saja.

## DAFTAR PUSTAKA

- Buckbee, M. (2019). *Data Security: Definition, Explanation and Guide*. Varonis. <https://www.varonis.com/blog/data-security/>
- Gabrielli, M., & Martini, S. (2010). *Programming Languages: Principles and Paradigms*. Springer London. <https://doi.org/10.1007/978-1-84882-914-5>
- Information, S. (2019). *Symmetric vs. Asymmetric Encryption – What are differences?* <https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences>
- Jogiyanto, H. M. (2016). *Analisis Dan Desain Sistem Informasi, Pendekatan Terstruktur Teori Dan Praktek Aplikasi Bisnis*. Andi Offset.
- Kurniawan, T. A. (2018). Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 5(1), 77. <https://doi.org/10.25126/jtiik.201851610>
- Ladjudin, A.-B. bin. (2017). *Analisis dan Desain Sistem Informasi*. Graha Ilmu.
- Lee, C. (2014). *Buku Pintar Pemrograman Visual Basic 2010*. Elex Media Komputindo.
- Mollin, R. A. (2001). *An Introduction to Cryptography, Second Edition* (2nd ed.). Chapman & Hall / CRC Press.
- Nakatsu, R. T. (2019). *Reasoning with Diagrams : Decision-Making and Problem-Solving with Diagrams*. John Wiley & Sons.
- Rouse, M., Rosencrance, L., & Cobb, M. (2019). *What is Asymmetric Cryptography?* TechTarget. <https://searchsecurity.techtarget.com/definition/asymmetric-cryptography>
- Smirnoff, P., & Turner, D. M. (2019). *Symmetric Key Encryption - why, where and how it's used in banking*. Cryptomathic. <https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking>
- Stallings, W. (2013). *Cryptography and Network Security: Principles and Practice*. Prentice Hall Press.
- Stallings, William. (2005). *Cryptography and Network Security Principles and Practices* (4th ed.). Prentice Hall.

- Sukmawati, R., & Priyadi, Y. (2019). Perancangan Proses Bisnis Menggunakan UML Berdasarkan Fit/Gap Analysis Pada Modul Inventory Odoo. *INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, 3(2), 104. <https://doi.org/10.29407/intensif.v3i2.12697>
- Technopedia. (2019). *Unified Modeling Language (UML)*. Technopedia. <https://www.techopedia.com/definition/3243/unified-modeling-language-uml>
- Uml-diagrams.org. (2019). *Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors)*. <https://www.uml-diagrams.org/use-case-diagrams.html>
- UTM. (2019). *Concept: Use-Case Model*. Univesidad Technologica de La Mixteca. [http://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/concepts/use\\_case\\_model\\_CD178AF9.html](http://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/concepts/use_case_model_CD178AF9.html)
- Wasserkrug, S., Dalvi, N., Munson, E. V., Gogolla, M., Sirangelo, C., Fischer-Hübner, S., Ives, Z., Velegrakis, Y., Bevan, N., Jensen, C. S., & Snodgrass, R. T. (2019). Unified Modeling Language. In *Encyclopedia of Database Systems* (pp. 3232–3239). Springer US. [https://doi.org/10.1007/978-0-387-39940-9\\_440](https://doi.org/10.1007/978-0-387-39940-9_440)
- Zwass, V. (2019). *Information System*. Britannica. <https://www.britannica.com/topic/information-system>
- Badawi, A. (2018). Evaluasi Pengaruh Modifikasi Three Pass Protocol Terhadap Transmisi Kunci Enkripsi.
- Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." *IT Journal Research and Development* 2.1 (2017): 1-11.
- Bahri, S. (2018). *Metodologi Penelitian Bisnis Lengkap Dengan Teknik Pengolahan Data SPSS*. Penerbit Andi (Anggota Ikapi). Percetakan Andi Offset. Yogyakarta.
- Erika, Winda, Heni Rachmawati, and Ibnu Surya. "Enkripsi Teks Surat Elektronik (E-Mail) Berbasis Algoritma Rivest Shamir Adleman (RSA)." *Jurnal Aksara Komputer Terapan* 1.2 (2012).
- Fitriani, W., Rahim, R., Oktaviana, B., & Siahaan, A. P. U. (2017). Vernam Encrypted Text in End of File Hiding Steganography Technique. *Int. J. Recent Trends Eng. Res*, 3(7), 214-219.
- Hardinata, R. S. (2019). Audit Tata Kelola Teknologi Informasi menggunakan Cobit 5 (Studi Kasus: Universitas Pembangunan Panca Budi Medan). *Jurnal Teknik dan Informatika*, 6(1), 42-45.
- Hariyanto, E., Lubis, S. A., & Sitorus, Z. (2017). Perancangan prototipe helm pengukur kualitas udara. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 1(1).
- Hariyanto, E., & Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. *International Journal of Science and Research (IJSR)*, 5(10), 1363-1365.
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapu durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).

- Iqbal, M., Siahaan, A. P. U., Purba, N. E., & Purwanto, D. (2017). Prim's Algorithm for Optimizing Fiber Optic Trajectory Planning. *Int. J. Sci. Res. Sci. Technol*, 3(6), 504-509.
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.
- Muttaqin, Muhammad. "ANALISA PEMANFAATAN SISTEM INFORMASI E-OFFICE PADA UNIVERSITAS PEMBANGUNAN PANCA BUDI MEDAN DENGAN MENGGUNAKAN METODE UTAUT." *Jurnal Teknik dan Informatika 5.1* (2018): 40-43.
- Ramadhan, Z., Zarlis, M., Efendi, S., & Siahaan, A. P. U. (2018). Perbandingan Algoritma Prim dengan Algoritma Floyd-Warshall dalam Menentukan Rute Terpendek (Shortest Path Problem). *JURIKOM (Jurnal Riset Komputer)*, 5(2), 135-139.
- Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol application for LAN and Wi-Fi communication. *Int. J. Eng. Technol.*, 7(2.13), 345-347.
- Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." *Jurnal Abdi Ilmu 10.2* (2018): 1899-1902.

```
'-----  
' <auto-generated>  
'   This code was generated by a tool.  
'   Runtime Version:4.0.30319.42000  
'  
'   Changes to this file may cause incorrect behavior and will be lost if  
'   the code is regenerated.  
' </auto-generated>  
'-----
```

```
Option Strict On  
Option Explicit On
```

```
Namespace My
```

```
'NOTE: This file is auto-generated; do not modify it directly. To make changes,  
' or if you encounter build errors in this file, go to the Project Designer  
' (go to Project Properties or double-click the My Project node in  
' Solution Explorer), and make changes on the Application tab.  
,
```

```
Partial Friend Class MyApplication
```

```
    <Global.System.Diagnostics.DebuggerStepThroughAttribute()> _  
    Public Sub New()
```

```
MyBase.New(Global.Microsoft.VisualBasic.ApplicationServices.AuthenticationMode.Windows)  
    Me.IsSingleInstance = false  
    Me.EnableVisualStyles = true  
    Me.SaveMySettingsOnExit = true  
    Me.ShutdownStyle =
```

```
Global.Microsoft.VisualBasic.ApplicationServices.ShutdownMode.AfterMainFormCloses  
End Sub
```

```
    <Global.System.Diagnostics.DebuggerStepThroughAttribute()> _  
    Protected Overrides Sub OnCreateMainForm()  
        Me.MainForm = Global.HBR.frmMenu  
    End Sub
```

```
End Class
```

```
End Namespace
```

```
Imports System  
Imports System.Reflection  
Imports System.Runtime.InteropServices
```

```
' General Information about an assembly is controlled through the following  
' set of attributes. Change these attribute values to modify the information  
' associated with an assembly.
```

```
' Review the values of the assembly attributes
```

```
<Assembly: AssemblyTitle("HBR")>  
<Assembly: AssemblyDescription("")>  
<Assembly: AssemblyCompany("")>  
<Assembly: AssemblyProduct("HBR")>
```



```
<Assembly: AssemblyCopyright("Copyright © 2019")>
<Assembly: AssemblyTrademark("")>
```

```
<Assembly: ComVisible(False)>
```

```
'The following GUID is for the ID of the typelib if this project is exposed to COM
```

```
<Assembly: Guid("51d748e1-a6a0-4bda-8479-107bd69552fa")>
```

```
' Version information for an assembly consists of the following four values:
```

```
,
```

```
' Major Version
```

```
' Minor Version
```

```
' Build Number
```

```
' Revision
```

```
,
```

```
' You can specify all the values or you can default the Build and Revision Numbers
```

```
' by using the '*' as shown below:
```

```
' <Assembly: AssemblyVersion("1.0.*")>
```

```
<Assembly: AssemblyVersion("1.0.0.0")>
```

```
<Assembly: AssemblyFileVersion("1.0.0.0")>
```

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
```

```
Partial Class frmHBR
```

```
    Inherits System.Windows.Forms.Form
```

```
'Form overrides dispose to clean up the component list.
```

```
<System.Diagnostics.DebuggerNonUserCode()> _
```

```
Protected Overrides Sub Dispose(ByVal disposing As Boolean)
```

```
    Try
```

```
        If disposing AndAlso components IsNot Nothing Then
            components.Dispose()
```

```
        End If
```

```
    Finally
```

```
        MyBase.Dispose(disposing)
```

```
    End Try
```

```
End Sub
```

```
'Required by the Windows Form Designer
```

```
Private components As System.ComponentModel.IContainer
```

```
'NOTE: The following procedure is required by the Windows Form Designer
```

```
'It can be modified using the Windows Form Designer.
```

```
'Do not modify it using the code editor.
```

```
<System.Diagnostics.DebuggerStepThrough()> _
```

```
Private Sub InitializeComponent()
```

```
    Me.Label1 = New System.Windows.Forms.Label()
```

```
    Me.Label2 = New System.Windows.Forms.Label()
```

```
    Me.Label3 = New System.Windows.Forms.Label()
```

```
    Me.txtPT = New System.Windows.Forms.RichTextBox()
```

```
    Me.txtKunci = New System.Windows.Forms.TextBox()
```

```
    Me.btnCT = New System.Windows.Forms.RichTextBox()
```

```
    Me.btnEnkripsi = New System.Windows.Forms.Button()
```

```
    Me.btnDekripsi = New System.Windows.Forms.Button()
```

```
    Me.txtPT2 = New System.Windows.Forms.RichTextBox()
```

```

Me.Label4 = New System.Windows.Forms.Label()
Me.txtLog = New System.Windows.Forms.RichTextBox()
Me.Label5 = New System.Windows.Forms.Label()
Me.rdbKanan = New System.Windows.Forms.RadioButton()
Me.rdbKiri = New System.Windows.Forms.RadioButton()
Me.SuspendLayout()
'
'Label1
'
Me.Label1.AutoSize = True
Me.Label1.BackColor = System.Drawing.Color.Transparent
Me.Label1.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.Label1.ForeColor = System.Drawing.Color.White
Me.Label1.Location = New System.Drawing.Point(34, 35)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(104, 25)
Me.Label1.TabIndex = 0
Me.Label1.Text = "Plaintext"
'
'Label2
'
Me.Label2.AutoSize = True
Me.Label2.BackColor = System.Drawing.Color.Transparent
Me.Label2.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.Label2.ForeColor = System.Drawing.Color.White
Me.Label2.Location = New System.Drawing.Point(34, 196)
Me.Label2.Name = "Label2"
Me.Label2.Size = New System.Drawing.Size(71, 25)
Me.Label2.TabIndex = 1
Me.Label2.Text = "Kunci"
'
'Label3
'
Me.Label3.AutoSize = True
Me.Label3.BackColor = System.Drawing.Color.Transparent
Me.Label3.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.Label3.ForeColor = System.Drawing.Color.White
Me.Label3.Location = New System.Drawing.Point(34, 260)
Me.Label3.Name = "Label3"
Me.Label3.Size = New System.Drawing.Size(120, 25)
Me.Label3.TabIndex = 2
Me.Label3.Text = "Ciphertext"
'
'txtPT
'
Me.txtPT.Font = New System.Drawing.Font("Courier New", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.txtPT.Location = New System.Drawing.Point(181, 35)
Me.txtPT.Name = "txtPT"
Me.txtPT.Size = New System.Drawing.Size(767, 137)
Me.txtPT.TabIndex = 3
Me.txtPT.Tag = ""
Me.txtPT.Text = "UNIVERSITAS PEMBANGUNAN PANCA BUDI"
'
'txtKunci

```

```

    Me.txtKunci.Font = New System.Drawing.Font("Courier New", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.txtKunci.Location = New System.Drawing.Point(181, 196)
    Me.txtKunci.Name = "txtKunci"
    Me.txtKunci.Size = New System.Drawing.Size(270, 31)
    Me.txtKunci.TabIndex = 4
    Me.txtKunci.Text = "4"
    'txtCT
    Me.txtCT.Font = New System.Drawing.Font("Courier New", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.txtCT.Location = New System.Drawing.Point(181, 254)
    Me.txtCT.Name = "txtCT"
    Me.txtCT.Size = New System.Drawing.Size(270, 137)
    Me.txtCT.TabIndex = 5
    Me.txtCT.Text = ""
    'btnEnkripsi
    Me.btnEnkripsi.BackColor = System.Drawing.Color.Cyan
    Me.btnEnkripsi.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.btnEnkripsi.Location = New System.Drawing.Point(485, 254)
    Me.btnEnkripsi.Name = "btnEnkripsi"
    Me.btnEnkripsi.Size = New System.Drawing.Size(152, 60)
    Me.btnEnkripsi.TabIndex = 6
    Me.btnEnkripsi.Text = "Enkripsi"
    Me.btnEnkripsi.UseVisualStyleBackColor = False
    'btnDekripsi
    Me.btnDekripsi.BackColor = System.Drawing.Color.Cyan
    Me.btnDekripsi.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.btnDekripsi.Location = New System.Drawing.Point(485, 331)
    Me.btnDekripsi.Name = "btnDekripsi"
    Me.btnDekripsi.Size = New System.Drawing.Size(152, 60)
    Me.btnDekripsi.TabIndex = 7
    Me.btnDekripsi.Text = "Dekripsi"
    Me.btnDekripsi.UseVisualStyleBackColor = False
    'txtPT2
    Me.txtPT2.Font = New System.Drawing.Font("Courier New", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.txtPT2.Location = New System.Drawing.Point(678, 254)
    Me.txtPT2.Name = "txtPT2"
    Me.txtPT2.Size = New System.Drawing.Size(270, 137)
    Me.txtPT2.TabIndex = 9
    Me.txtPT2.Text = ""
    'Label14
    Me.Label14.AutoSize = True
    Me.Label14.BackColor = System.Drawing.Color.Transparent

```

```

        Me.Label4.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label4.ForeColor = System.Drawing.Color.White
        Me.Label4.Location = New System.Drawing.Point(34, 366)
        Me.Label4.Name = "Label4"
        Me.Label4.Size = New System.Drawing.Size(51, 25)
        Me.Label4.TabIndex = 8
        Me.Label4.Text = "Log"
    '
    'txtLog
    '
    Me.txtLog.Font = New System.Drawing.Font("Courier New", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.txtLog.Location = New System.Drawing.Point(39, 410)
    Me.txtLog.Name = "txtLog"
    Me.txtLog.Size = New System.Drawing.Size(909, 229)
    Me.txtLog.TabIndex = 10
    Me.txtLog.Text = ""
    '
    'Label15
    '
    Me.Label15.AutoSize = True
    Me.Label15.BackColor = System.Drawing.Color.Transparent
    Me.Label15.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.Label15.ForeColor = System.Drawing.Color.White
    Me.Label15.Location = New System.Drawing.Point(673, 196)
    Me.Label15.Name = "Label15"
    Me.Label15.Size = New System.Drawing.Size(61, 25)
    Me.Label15.TabIndex = 11
    Me.Label15.Text = "Arah"
    '
    'rdbKanan
    '
    Me.rdbKanan.AutoSize = True
    Me.rdbKanan.BackColor = System.Drawing.Color.Transparent
    Me.rdbKanan.Checked = True
    Me.rdbKanan.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.rdbKanan.ForeColor = System.Drawing.Color.White
    Me.rdbKanan.Location = New System.Drawing.Point(761, 194)
    Me.rdbKanan.Name = "rdbKanan"
    Me.rdbKanan.Size = New System.Drawing.Size(97, 29)
    Me.rdbKanan.TabIndex = 13
    Me.rdbKanan.TabStop = True
    Me.rdbKanan.Text = "Kanan"
    Me.rdbKanan.UseVisualStyleBackColor = False
    '
    'rdbKiri
    '
    Me.rdbKiri.AutoSize = True
    Me.rdbKiri.BackColor = System.Drawing.Color.Transparent
    Me.rdbKiri.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.rdbKiri.ForeColor = System.Drawing.Color.White
    Me.rdbKiri.Location = New System.Drawing.Point(887, 194)
    Me.rdbKiri.Name = "rdbKiri"
    Me.rdbKiri.Size = New System.Drawing.Size(65, 29)

```

```

Me.rdbKiri.TabIndex = 14
Me.rdbKiri.Text = "Kiri"
Me.rdbKiri.UseVisualStyleBackColor = False
'
'frmHBR
'
Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.BackColor = System.Drawing.Color.FromArgb(CType(CType(192, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
Me.BackgroundImage = Global.HBR.My.Resources.Resources.Wall
Me.ClientSize = New System.Drawing.Size(984, 661)
Me.Controls.Add(Me.rdbKiri)
Me.Controls.Add(Me.rdbKanan)
Me.Controls.Add(Me.Label5)
Me.Controls.Add(Me.txtLog)
Me.Controls.Add(Me.txtPT2)
Me.Controls.Add(Me.Label4)
Me.Controls.Add(Me.btnDekripsi)
Me.Controls.Add(Me.btnEnkripsi)
Me.Controls.Add(Me.txtCT)
Me.Controls.Add(Me.txtKunci)
Me.Controls.Add(Me.txtPT)
Me.Controls.Add(Me.Label3)
Me.Controls.Add(Me.Label2)
Me.Controls.Add(Me.Label1)
Me.Name = "frmHBR"
Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
Me.Text = "Kriptografi Pergeseran Bit Secara Horizontal"
Me.ResumeLayout(False)
Me.PerformLayout()

```

```
End Sub
```

```

Friend WithEvents Label1 As System.Windows.Forms.Label
Friend WithEvents Label2 As System.Windows.Forms.Label
Friend WithEvents Label3 As System.Windows.Forms.Label
Friend WithEvents txtPT As System.Windows.Forms.RichTextBox
Friend WithEvents txtKunci As System.Windows.Forms.TextBox
Friend WithEvents txtCT As System.Windows.Forms.RichTextBox
Friend WithEvents btnEnkripsi As System.Windows.Forms.Button
Friend WithEvents btnDekripsi As System.Windows.Forms.Button
Friend WithEvents txtPT2 As System.Windows.Forms.RichTextBox
Friend WithEvents Label4 As System.Windows.Forms.Label
Friend WithEvents txtLog As System.Windows.Forms.RichTextBox
Friend WithEvents Label5 As System.Windows.Forms.Label
Friend WithEvents rdbKanan As System.Windows.Forms.RadioButton
Friend WithEvents rdbKiri As System.Windows.Forms.RadioButton

```

```
End Class
```

```

Public Class frmHBR
    Dim Log As String
    Dim PT, CT As String
    Dim PT_Bit(), CT_Bit() As String
    Dim CT_ASCII() As Byte

```

```

Dim Kunci, Arah As Byte

Private Function Rotate(ByVal Str As String, ByVal n As Byte, ByVal Direction As
Byte) As String
    Dim Result As String = ""

    n = n Mod 8

    If Direction = 0 Then
        Result = Str.Remove(0, n) & Str.Substring(0, n)
    ElseIf Direction = 1 Then
        Result = Str.Substring(Str.Length - n, n) & Str.Remove(Str.Length - n, n)
    End If
    Return Result
End Function

Private Sub btnEnkripsi_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEnkripsi.Click
    Log = ""
    txtLog.Text = ""
    PT = txtPT.Text
    CT = ""
    Kunci = txtKunci.Text

    If rdbKiri.Checked Then
        Arah = 0
    ElseIf rdbKanan.Checked Then
        Arah = 1
    End If

    ReDim PT_Bit(PT.Length - 1)
    ReDim CT_Bit(PT.Length - 1)
    ReDim CT_ASCII(PT.Length - 1)

    For i = 0 To PT.Length - 1
        PT_Bit(i) = Convert.ToString(Convert.ToByte(PT(i)), 2)
        For h = 0 To 8 - PT_Bit(i).Length - 1
            PT_Bit(i) = "0" & PT_Bit(i)
        Next

        CT_Bit(i) = Rotate(PT_Bit(i), Kunci, Arah)
        CT_ASCII(i) = Convert.ToByte(CT_Bit(i), 2)
        CT &= Convert.ToChar(CT_ASCII(i))
        Log &= "PT[" & i & "] " & vbTab & "= " & PT(i) & " = " &
Convert.ToByte(PT(i)) & " = " & PT_Bit(i) & vbTab & "--ROT " & Kunci & "-->" & vbTab &
"CT[" & i & "] " & vbTab & "= " & CT_Bit(i) & " = " & Convert.ToByte(CT_Bit(i), 2) &
vbTab & " = " & CT(i) & vbCrLf

    Next

    txtCT.Text = CT
    txtLog.Text = Log
End Sub

Private Sub btnDekripsi_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDekripsi.Click
    Log = ""
    txtLog.Text = ""

```

```

CT = txtCT.Text

PT = ""
Kunci = txtKunci.Text

If rdbKiri.Checked Then
    Arah = 0
ElseIf rdbKanan.Checked Then
    Arah = 1
End If

ReDim PT_Bit(CT_ASCII.Length - 1)
ReDim CT_Bit(CT_ASCII.Length - 1)

For i = 0 To CT_ASCII.Length - 1
    CT_Bit(i) = Convert.ToString(CT_ASCII(i), 2)
    For h = 0 To 8 - CT_Bit(i).Length - 1
        CT_Bit(i) = "0" & CT_Bit(i)
    Next

    PT_Bit(i) = Rotate(CT_Bit(i), Kunci, Arah)
    PT &= Convert.ToChar(Convert.ToByte(PT_Bit(i), 2))
    Log &= "CT[" & i & "]" & vbTab & "= " & CT(i) & " = " & CT_ASCII(i) & " = "
& CT_Bit(i) & vbTab & "--ROT " & Kunci & "-->" & vbTab & "PT[" & i & "]" & vbTab & "= "
& PT_Bit(i) & " = " & Convert.ToByte(PT_Bit(i), 2) & vbTab & " = " & PT(i) & vbCrLf

Next

txtPT2.Text = PT
txtLog.Text = Log
End Sub
End Class

```

```

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated(> _
Partial Class frmInfo
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode(> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.

```

```

<System.Diagnostics.DebuggerStepThrough()> _
Private Sub InitializeComponent()
    Dim resources As System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(frmInfo))
    Me.Label1 = New System.Windows.Forms.Label()
    Me.SuspendLayout()
    '
    'Label1
    '
    Me.Label1.BackColor = System.Drawing.Color.Cyan
    Me.Label1.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.Label1.Location = New System.Drawing.Point(28, 18)
    Me.Label1.Name = "Label1"
    Me.Label1.Size = New System.Drawing.Size(705, 392)
    Me.Label1.TabIndex = 5
    Me.Label1.Text = resources.GetString("Label1.Text")
    Me.Label1.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
    '
    'frmInfo
    '
    Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
    Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
    Me.BackColor = System.Drawing.Color.FromArgb(CType(CType(192, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
    Me.BackgroundImage = Global.HBR.My.Resources.Resources.Wall
    Me.ClientSize = New System.Drawing.Size(761, 433)
    Me.Controls.Add(Me.Label1)
    Me.Name = "frmInfo"
    Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
    Me.Text = "Info"
    Me.ResumeLayout(False)

End Sub
Friend WithEvents Label1 As System.Windows.Forms.Label
End Class

```

```

Public Class frmInfo
End Class

```

```

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class frmMenu
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        End Try
    End Sub

```



```

    Finally
        MyBase.Dispose(disposing)
    End Try
End Sub

'Required by the Windows Form Designer
Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough()> _
Private Sub InitializeComponent()
    Me.btnHBR = New System.Windows.Forms.Button()
    Me.btnInfo = New System.Windows.Forms.Button()
    Me.btnAbout = New System.Windows.Forms.Button()
    Me.btnKeluar = New System.Windows.Forms.Button()
    Me.Label1 = New System.Windows.Forms.Label()
    Me.PictureBox1 = New System.Windows.Forms.PictureBox()
    CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).BeginInit()
    Me.SuspendLayout()
    '
    'btnHBR
    '
    Me.btnHBR.BackColor = System.Drawing.Color.Cyan
    Me.btnHBR.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.btnHBR.Location = New System.Drawing.Point(34, 27)
    Me.btnHBR.Name = "btnHBR"
    Me.btnHBR.Size = New System.Drawing.Size(173, 60)
    Me.btnHBR.TabIndex = 0
    Me.btnHBR.Text = "Kriptografi HBR"
    Me.btnHBR.UseVisualStyleBackColor = False
    '
    'btnInfo
    '
    Me.btnInfo.BackColor = System.Drawing.Color.Cyan
    Me.btnInfo.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.btnInfo.Location = New System.Drawing.Point(34, 118)
    Me.btnInfo.Name = "btnInfo"
    Me.btnInfo.Size = New System.Drawing.Size(173, 60)
    Me.btnInfo.TabIndex = 1
    Me.btnInfo.Text = "Info"
    Me.btnInfo.UseVisualStyleBackColor = False
    '
    'btnAbout
    '
    Me.btnAbout.BackColor = System.Drawing.Color.Cyan
    Me.btnAbout.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.btnAbout.Location = New System.Drawing.Point(34, 210)
    Me.btnAbout.Name = "btnAbout"
    Me.btnAbout.Size = New System.Drawing.Size(173, 60)
    Me.btnAbout.TabIndex = 2
    Me.btnAbout.Text = "Tentang Saya"
    Me.btnAbout.UseVisualStyleBackColor = False
    '

```

```

        'btnKeluar
        ,
        Me.btnKeluar.BackColor = System.Drawing.Color.Cyan
        Me.btnKeluar.Font = New System.Drawing.Font("Microsoft Sans Serif", 15.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.btnKeluar.Location = New System.Drawing.Point(34, 303)
        Me.btnKeluar.Name = "btnKeluar"
        Me.btnKeluar.Size = New System.Drawing.Size(173, 60)
        Me.btnKeluar.TabIndex = 3
        Me.btnKeluar.Text = "Keluar Aplikasi"
        Me.btnKeluar.UseVisualStyleBackColor = False
        ,
        'Label1
        ,
        Me.Label1.BackColor = System.Drawing.Color.Cyan
        Me.Label1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
        Me.Label1.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label1.Location = New System.Drawing.Point(235, 285)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(433, 78)
        Me.Label1.TabIndex = 4
        Me.Label1.Text = "Rancang Bangun Keamanan Pengiriman Pesan Dengan Teknik
Pergeseran Bit Secara Hori" & _
            "zontal Pada Cipher Substitusi"
        Me.Label1.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
        ,
        'PictureBox1
        ,
        Me.PictureBox1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
        Me.PictureBox1.Image = Global.HBR.My.Resources.Resources.Eka
        Me.PictureBox1.Location = New System.Drawing.Point(339, 37)
        Me.PictureBox1.Name = "PictureBox1"
        Me.PictureBox1.Size = New System.Drawing.Size(224, 221)
        Me.PictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.PictureBox1.TabIndex = 5
        Me.PictureBox1.TabStop = False
        ,
        'frmMenu
        ,
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.BackColor = System.Drawing.Color.FromArgb(CType(CType(192, Byte), Integer),
CType(CType(255, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.BackgroundImage = Global.HBR.My.Resources.Resources.Wall
        Me.ClientSize = New System.Drawing.Size(696, 402)
        Me.Controls.Add(Me.PictureBox1)
        Me.Controls.Add(Me.Label1)
        Me.Controls.Add(Me.btnKeluar)
        Me.Controls.Add(Me.btnAbout)
        Me.Controls.Add(Me.btnInfo)
        Me.Controls.Add(Me.btnHBR)
        Me.Name = "frmMenu"
        Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "Menu Utama"
        CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
    
```

```

End Sub
Friend WithEvents btnHBR As System.Windows.Forms.Button
Friend WithEvents btnInfo As System.Windows.Forms.Button
Friend WithEvents btnAbout As System.Windows.Forms.Button
Friend WithEvents btnKeluar As System.Windows.Forms.Button
Friend WithEvents Label1 As System.Windows.Forms.Label
Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
End Class

```

```
Public Class frmMenu
```

```

    Private Sub btnHBR_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnHBR.Click
        frmHBR.ShowDialog()
    End Sub

```

```

    Private Sub btnInfo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnInfo.Click
        frmInfo.ShowDialog()
    End Sub

```

```

    Private Sub btnAbout_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAbout.Click
        frmTentang.ShowDialog()
    End Sub

```

```

    Private Sub btnKeluar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnKeluar.Click
        Application.Exit()
    End Sub
End Class

```

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
```

```
Partial Class frmTentang
```

```
    Inherits System.Windows.Forms.Form
```

```
    'Form overrides dispose to clean up the component list.
```

```
    <System.Diagnostics.DebuggerNonUserCode()> _
```

```
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
```

```
        Try
```

```
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
```

```
        Finally
```

```
            MyBase.Dispose(disposing)
```

```
        End Try
```

```
    End Sub
```

```
    'Required by the Windows Form Designer
```

```
    Private components As System.ComponentModel.IContainer
```

```
    'NOTE: The following procedure is required by the Windows Form Designer
```

```

'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough(> _
Private Sub InitializeComponent()
    Me.Label1 = New System.Windows.Forms.Label()
    Me.PictureBox1 = New System.Windows.Forms.PictureBox()
    CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).BeginInit()
    Me.SuspendLayout()
    '
    'Label1
    '
    Me.Label1.BackColor = System.Drawing.Color.Cyan
    Me.Label1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D
    Me.Label1.Font = New System.Drawing.Font("Microsoft Sans Serif", 14.25!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
    Me.Label1.Location = New System.Drawing.Point(54, 315)
    Me.Label1.Name = "Label1"
    Me.Label1.Size = New System.Drawing.Size(360, 191)
    Me.Label1.TabIndex = 5
    Me.Label1.Text = "Nama: Eka Andy Nova" & Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & "NPM: 1414370458" &
Global.Microsoft.VisualBasic.ChrW(13) & Global.Microsoft.VisualBasic.ChrW(10) & "Program
Studi: Sistem Komputer" & Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & "Fakultas: " & _
    "Sains dan Teknologi" & Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & "Universitas Pembangunan Panca Budi" &
Global.Microsoft.VisualBasic.ChrW(13) & Global.Microsoft.VisualBasic.ChrW(10) & "2019"
    Me.Label1.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
    '
    'PictureBox1
    '
    Me.PictureBox1.BackColor = System.Drawing.Color.Transparent
    Me.PictureBox1.Image = Global.HBR.My.Resources.Resources.Logo_Unpab
    Me.PictureBox1.Location = New System.Drawing.Point(107, 36)
    Me.PictureBox1.Name = "PictureBox1"
    Me.PictureBox1.Size = New System.Drawing.Size(250, 250)
    Me.PictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage
    Me.PictureBox1.TabIndex = 0
    Me.PictureBox1.TabStop = False
    '
    'frmTentang
    '
    Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
    Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
    Me.BackColor = System.Drawing.Color.FromArgb(CType(CType(192, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
    Me.BackgroundImage = Global.HBR.My.Resources.Resources.Wall
    Me.ClientSize = New System.Drawing.Size(461, 531)
    Me.Controls.Add(Me.Label1)
    Me.Controls.Add(Me.PictureBox1)
    Me.Name = "frmTentang"
    Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
    Me.Text = "Tentang Saya"
    CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).EndInit()
    Me.ResumeLayout(False)
End Sub

```

```

    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
    Friend WithEvents Label1 As System.Windows.Forms.Label
End Class

Public Class frmTentang

End Class

Public Class frmHBR
    Dim Log As String
    Dim PT, CT As String
    Dim PT_Bit(), CT_Bit() As String
    Dim CT_ASCII() As Byte
    Dim Kunci, Arah As Byte

    Private Function Rotate(ByVal Str As String, ByVal n As Byte, ByVal Direction As
Byte) As String
        Dim Result As String = ""

        n = n Mod 8

        If Direction = 0 Then
            Result = Str.Remove(0, n) & Str.Substring(0, n)
        ElseIf Direction = 1 Then
            Result = Str.Substring(Str.Length - n, n) & Str.Remove(Str.Length - n, n)
        End If
        Return Result
    End Function

    Private Sub btnEnkripsi_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEnkripsi.Click
        Log = ""
        txtLog.Text = ""
        PT = txtPT.Text
        CT = ""
        Kunci = txtKunci.Text

        If rdbKiri.Checked Then
            Arah = 0
        ElseIf rdbKanan.Checked Then
            Arah = 1
        End If

        ReDim PT_Bit(PT.Length - 1)
        ReDim CT_Bit(PT.Length - 1)
        ReDim CT_ASCII(PT.Length - 1)

        For i = 0 To PT.Length - 1
            PT_Bit(i) = Convert.ToString(Convert.ToByte(PT(i)), 2)
            For h = 0 To 8 - PT_Bit(i).Length - 1
                PT_Bit(i) = "0" & PT_Bit(i)
            Next

            CT_Bit(i) = Rotate(PT_Bit(i), Kunci, Arah)
            CT_ASCII(i) = Convert.ToByte(CT_Bit(i), 2)
            CT &= Convert.ToChar(CT_ASCII(i))
        Next
    End Sub
End Class

```

```

        Log &= "PT[" & i & "]" & vbTab & "= " & PT(i) & " = " &
Convert.ToByte(PT(i)) & " = " & PT_Bit(i) & vbTab & "--ROT " & Kunci & "-->" & vbTab &
"CT[" & i & "]" & vbTab & "= " & CT_Bit(i) & " = " & Convert.ToByte(CT_Bit(i), 2) &
vbTab & " = " & CT(i) & vbCrLf

    Next

    txtCT.Text = CT
    txtLog.Text = Log
End Sub

Private Sub btnDekripsi_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDekripsi.Click
    Log = ""
    txtLog.Text = ""
    CT = txtCT.Text

    PT = ""
    Kunci = txtKunci.Text

    If rdbKiri.Checked Then
        Arah = 0
    ElseIf rdbKanan.Checked Then
        Arah = 1
    End If

    ReDim PT_Bit(CT_ASCII.Length - 1)
    ReDim CT_Bit(CT_ASCII.Length - 1)

    For i = 0 To CT_ASCII.Length - 1
        CT_Bit(i) = Convert.ToString(CT_ASCII(i), 2)
        For h = 0 To 8 - CT_Bit(i).Length - 1
            CT_Bit(i) = "0" & CT_Bit(i)
        Next

        PT_Bit(i) = Rotate(CT_Bit(i), Kunci, Arah)
        PT &= Convert.ToChar(Convert.ToByte(PT_Bit(i), 2))
        Log &= "CT[" & i & "]" & vbTab & "= " & CT(i) & " = " & CT_ASCII(i) & " = "
& CT_Bit(i) & vbTab & "--ROT " & Kunci & "-->" & vbTab & "PT[" & i & "]" & vbTab & "= "
& PT_Bit(i) & " = " & Convert.ToByte(PT_Bit(i), 2) & vbTab & " = " & PT(i) & vbCrLf

    Next

    txtPT2.Text = PT
    txtLog.Text = Log
End Sub
End Class

```

```

'-----
' <auto-generated>
' This code was generated by a tool.
' Runtime Version:4.0.30319.42000
'
' Changes to this file may cause incorrect behavior and will be lost if
' the code is regenerated.

```

```
' </auto-generated>
```

```
Option Strict On  
Option Explicit On
```

```
Imports System
```

```
Namespace My.Resources
```

```
'This class was auto-generated by the StronglyTypedResourceBuilder  
'class via a tool like ResGen or Visual Studio.  
'To add or remove a member, edit your .ResX file then rerun ResGen  
'with the /str option, or rebuild your VS project.  
'''<summary>  
''' A strongly-typed resource class, for looking up localized strings, etc.  
'''</summary>
```

```
<Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTy  
pedResourceBuilder", "4.0.0.0"), _  
Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _  
Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _  
Global.Microsoft.VisualBasic.HideModuleNameAttribute()> _  
Friend Module Resources
```

```
Private resourceMan As Global.System.Resources.ResourceManager
```

```
Private resourceCulture As Global.System.Globalization.CultureInfo
```

```
'''<summary>  
''' Returns the cached ResourceManager instance used by this class.  
'''</summary>
```

```
<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.Edito  
rBrowsableState.Advanced)> _
```

```
Friend ReadOnly Property ResourceManager() As  
Global.System.Resources.ResourceManager
```

```
Get  
If Object.ReferenceEquals(resourceMan, Nothing) Then  
Dim temp As Global.System.Resources.ResourceManager = New  
Global.System.Resources.ResourceManager("HBR.Resources", GetType(Resources).Assembly)  
resourceMan = temp  
End If  
Return resourceMan  
End Get  
End Property
```

```
'''<summary>  
''' Overrides the current thread's CurrentUICulture property for all  
''' resource lookups using this strongly typed resource class.  
'''</summary>
```

```
<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.Edito  
rBrowsableState.Advanced)> _
```

```
Friend Property Culture() As Global.System.Globalization.CultureInfo  
Get  
Return resourceCulture  
End Get
```

```

        Set
            resourceCulture = value
        End Set
    End Property

    '''<summary>
    ''' Looks up a localized resource of type System.Drawing.Bitmap.
    '''</summary>
    Friend ReadOnly Property Eka() As System.Drawing.Bitmap
        Get
            Dim obj As Object = ResourceManager.GetObject("Eka", resourceCulture)
            Return CType(obj,System.Drawing.Bitmap)
        End Get
    End Property

    '''<summary>
    ''' Looks up a localized resource of type System.Drawing.Bitmap.
    '''</summary>
    Friend ReadOnly Property Logo_Unpab() As System.Drawing.Bitmap
        Get
            Dim obj As Object = ResourceManager.GetObject("Logo Unpab",
resourceCulture)
            Return CType(obj,System.Drawing.Bitmap)
        End Get
    End Property

    '''<summary>
    ''' Looks up a localized resource of type System.Drawing.Bitmap.
    '''</summary>
    Friend ReadOnly Property Wall() As System.Drawing.Bitmap
        Get
            Dim obj As Object = ResourceManager.GetObject("Wall", resourceCulture)
            Return CType(obj,System.Drawing.Bitmap)
        End Get
    End Property
End Module
End Namespace

```

```

'-----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'-----

```

```

Option Strict On
Option Explicit On

```

```

Namespace My

```

```

    <Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _

```



```

Global.System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.Set
tingsDesigner.SettingsSingleFileGenerator", "10.0.0.0"), _

Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.Editor
BrowsableState.Advanced)> _
    Partial Friend NotInheritable Class MySettings
        Inherits Global.System.Configuration.ApplicationSettingsBase

        Private Shared defaultInstance As MySettings =
CType(Global.System.Configuration.ApplicationSettingsBase.Synchronized(New MySettings),
MySettings)

#Region "My.Settings Auto-Save Functionality"
#If _MyType = "WindowsForms" Then
    Private Shared addedHandler As Boolean

    Private Shared addedHandlerLockObject As New Object

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.Editor
BrowsableState.Advanced)> _
        Private Shared Sub AutoSaveSettings(ByVal sender As Global.System.Object, ByVal e
As Global.System.EventArgs)
            If My.Application.SaveMySettingsOnExit Then
                My.Settings.Save()
            End If
        End Sub
#End If
#End Region

        Public Shared ReadOnly Property [Default]() As MySettings
            Get

#If _MyType = "WindowsForms" Then
                If Not addedHandler Then
                    SyncLock addedHandlerLockObject
                        If Not addedHandler Then
                            AddHandler My.Application.Shutdown, AddressOf
AutoSaveSettings
                                addedHandler = True
                            End If
                        End SyncLock
                    End If
                End If
            End Get
        End Property
    End Class
End Namespace

Namespace My

    <Global.Microsoft.VisualBasic.HideModuleNameAttribute(), _
    Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
    Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(> _
    Friend Module MySettingsProperty

```

```
<Global.System.ComponentModel.Design.HelpKeywordAttribute("My.Settings")> _  
Friend ReadOnly Property Settings() As Global.HBR.My.MySettings  
    Get  
        Return Global.HBR.My.MySettings.Default  
    End Get  
End Property  
End Module  
End Namespace
```