IMPLEMENTASI PENGAMAN DATA MENGGUNAKAN ALGORITMA *VIGENERE CIPHER*

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	**:**	**SAYDIL AMRI**
NPM	**:**	**1514370066**
PROGRAM STUDI	**:**	**SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

LEMBAR PENGESAHAN

IMPLEMENTASI PENGAMANAN DATA MENGGUNAKAN ALGORITMA VIGENERE CIPHER

Disusun Oleh:

NAMA	**: SAYDIL AMRI**
NPM	**: 1514370066**
PROGRAM STUDI	**: SISTEM KOMPUTER**

Skripsi Telah Disetujui oleh Dosen Pembimbing Skripsi
Pada Tanggal :

Dosen Pembimbing I	Dosen Pembimbing II
Andysah P. U. Siahaan, S.Kom., M.Kom.,	Ika Devi Perwitasari, S.Kom., M.Kom.

Mengetahui:

Dekan Fakultas Sains dan Teknologi	Ketua Program Studi Sistem Komputer
Handam, S.T., MT	Eko Hariyanto, S.Kom., M.Kom.

PERNYATAAN ORISINALITAS

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

Medan, 29 juli 2020

SAYDIL AMRI

NPM: 1514370066

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : SAYDIL AMRI

NPM : 1514370066

Prodi : Sistem Komputer

Konsentrasi : Keamanan Jaringan Komputer

Judul Skripsi : **IMPLEMENTASI PENGAMANAN DATA MENGGUNAKAN ALGORITMA *VIGENERE CIPHER***

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, terima kasih.

Medan, 29 Juli 2020

Yang membuat pernyataan

SAYDIL AMRI

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

versitas	: Universitas Pembangunan Panca Budi
kultas	: SAINS & TEKNOLOGI
sen Pembimbing I	: Andysah Putera Utama Siahaan, S.Kom., M.Kom., Ph.D.
sen Pembimbing II	: Ika Devi Perwitasari, S.Kom., M.Kom.
ma Mahasiswa	: SAYDIL AMRI
usan/Program Studi	: Sistem Komputer
mor Pokok Mahasiswa	: 1514370066
jang Pendidikan	: S1
dul Tugas Akhir/Skripsi	: Implementasi Pengamanan Data Menggunakan Algoritma Vigenere cipher

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
/12 2018	Acc Seminar		
/3 2019	Revisi Bab I		
10/4	Revisi Bab II		
15/5	Revisi Bab III		
21/7	Revisi Bab IV		
17/10	Revisi Bab IV & V		
19/11	Acc Seminar Hasil		
9/12	Acc Sidang		
1/8	Acc [redacted] Jilid		

13/8/20
Medan, 19 November 2019
Diketahui/Disetujui oleh :
Dekan,

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

*) Coret yang tidak perlu

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

Universitas	: Universitas Pembangunan Panca Budi
Fakultas	: SAINS & TEKNOLOGI
Dosen Pembimbing I	: Andysah Putera Utama Siahaan, S.Kom, M.Kom, Ph.D.
Dosen Pembimbing II	: Ika Devi Ramayanti, S.Kom., M.Kom.
Nama Mahasiswa	: SAYDIL AMRI
Jurusan/Program Studi	: Sistem Komputer
Nomor Pokok Mahasiswa	: 1514370066
Jenjang Pendidikan	: S1
Judul Tugas Akhir/Skripsi	: Implementasi Pengamanan Data Menggunakan Algoritma Vigenere Cipher

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
24/2 2019	Revisi Judul		
10/3	Revisi Bab I		
7/4	Revisi Bab I, II		
28/4	Revisi Bab II, III		
19/5	Revisi Bab III		
23/6	Revisi Bab III, IV		
28/7	Revisi Bab IV, V		
1/9	Acc Seminar		
9/12	Acc Sidang		
8/8 2020	Acc Jilid		

Medan, 10 November 2019
Diketahui/Disetujui oleh :
Dekan

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

*) Coret yang tidak perlu

Plagiarism Detector v. 1460 - Originality Report

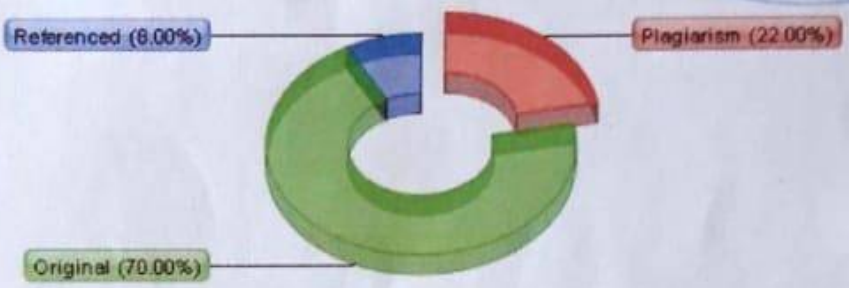
Analyzed document: 12/16/19 15:47:58

"SAYDIL AMRI_1514370066_SISTEM KOMPUTER.docx"

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:

% 6	wrds: 478	http://informatika.stei.itb.ac.id/~rinaldi.munir/Buku/Kriptografi/Bab-1_Penganta...
% 4	wrds: 348	https://1001pengertian.blogspot.com/2017/03/pengertian-uml.html
% 4	wrds: 360	http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2007-2008/Makalah1/...

[Show other Sources:]

Processed resources details:

104 - Ok / 9 - Failed

[Show other Sources:]

Important notes:

Wikipedia: Google Books: Ghostwriting services: Anti-cheating:

Telah Diperiksa oleh LPMU
dengan Plagiarisme.....22.....%
Medan,.....16 desember 2019.....
AN Ka. LPMU
HUSNI M.PILIANGA, BA., MSc.
Cahyo Pramono, SE, MM

FM-BPAA-2012-041

Hal : Permohonan Meja Hijau

Medan, 16 Desember 2019
Kepada Yth : Bapak/Ibu Dekan
Fakultas SAINS & TEKNOLOGI
UNPAB Medan
Di -
Tempat

Telah di terima berkas persyaratan dapat di proses
Medan, 16/12/2019
Ka. BPAA
an.
TEGUH WAHYONO, SE, MM.

Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama	: SAYDIL AMRI
Tempat/Tgl. Lahir	: Karang Rejo / 28 Februari 1998
Nama Orang Tua	: SUTRISNO
N. P. M	: 1514370066
Fakultas	: SAINS & TEKNOLOGI
Program Studi	: Sistem Komputer
No. HP	: 085360190089
Alamat	: Karang Rejo Dusun IX

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul **Implementasi Pengamanan Data menggunakan Algoritma Vigenere Cipher**, Selanjutnya saya menyatakan :

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

1.	[102] Ujian Meja Hijau	: Rp.	250.000
2.	[170] Administrasi Wisuda	: Rp.	1,500,000
3.	[202] Bebas Pustaka	: Rp.	100,000
4.	[221] Bebas LAB	: Rp.	5,000
	Total Biaya	: Rp.	~~1,605,000~~ 1855.000

Windy 16/12/19

UK 50% Rp 2895.000

4750.000

Periode Wisuda Ke : 64

Ukuran Toga : M

Diketahui/Disetujui oleh :

17/12

Hamdani, ST., MT
Dekan Fakultas SAINS & TEKNOLOGI

Hormat saya

SAYDIL AMRI
1514370066

Catatan :

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

TANDA BEBAS PUSTAKA
No. 1330/PERP/BP/2019
Dinyatakan tidak ada sangkut paut dengan UPT. Perpustakaan
Medan, 16 DEC 2019
Ka. UPT Perpustakaan
SALNIA S.IP

UKM CENTER PANCA BUDI
an. Ka. UKM-C
16/12/2019

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*

Saya yang bertanda tangan di bawah ini :

Nama Lengkap	: SAYDIL AMRI
Tempat/Tgl. Lahir	: Karang Rejo / 28 Februari 1998
Nomor Pokok Mahasiswa	: 1514370066
Program Studi	: Sistem Komputer
Konsentrasi	: Keamanan Jaringan Komputer
Jumlah Kredit yang telah dicapai	: 141 SKS, IPK 3.68
Nomor Hp	: 085360190089
Dengan ini mengajukan judul sesuai bidang ilmu sebagai berikut	:

No.	Judul
1.	Implementasi Pengamanan Data menggunakan Algoritma Vigenere Cipher

Catatan : Diisi Oleh Dosen Jika Ada Perubahan Judul

*Coret Yang Tidak Perlu

Rektor I,

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

Medan, ~~22 November 2019~~

Pemohon,

(Saydil Amri)

Tanggal : .....
Disahkan oleh :
Dekan

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

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

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

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

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

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

(Ika Deva Perwitasari, S.Kom., M.Kom)

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

KARTU BEBAS PRAKTIKUM

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

Nama	:	SAYDIL AMRI
N.P.M.	:	1514370066
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, 11 Desember 2019
Ka. Laboratorium

Fachrid Wadly, S. Kom

ABSTRAK

SAYDIL AMRI

Implementasi Pengamanan Data Menggunakan Algoritma *Vigenere Cipher*

2020

Keamanan data berfungsi melindungi data dari akses yang tidak sah dan pencurian data sepanjang data tersebut masih ada. Keamanan data termasuk enkripsi data, tokenization, dan praktik manajemen kunci yang melindungi data di semua aplikasi dan platform. Pencurian data sering terjadi karena terjadi kelalaian yang dilakukan oleh pemilik data tersebut. Jaringan internet yang luas menjadikan suatu informasi tersebut tidak aman dan dapat dicuri secara bebas. Teknik kriptografi sangat dibutuhkan untuk meningkatkan keamanan informasi tersebut. Teknik ini untuk menjaga agar data tersebut teracak dan tidak dapat difahami oleh orang yang mencurinya sehingga walaupun data tersebut dapat dicuri, isi dan makna data tersebut tidak dapat difahami oleh orang tersebut. Algoritma Vigenere cipher adalah algoritma yang baik dan cepat dalam melakukan proses enkripsi. Algoritma ini termasuk jenis kriptografi substitusi yang dalam proses enkripsinya, algoritma ini akan menggeser karakter plaintext sebesar kunci yang diberikan. Dengan menerapkan algoritma Vigenere Cipher, keamanan data akan lebih baik dan terjamin.

Kata Kunci: algoritma,dekripsi, enkripsi, kriptografi, Vigenere

DAFTAR GAMBAR

Gambar 2.1 Flowchart algoritma bidang matematika 14

Gambar 2.2 Flowchart algoritma bidang komputer..... 15

Gambar 2.3 Flowchart algoritma bidang pendidikan 16

Gambar 2.4 Skema enkripsi dan dekripsi dengan menggunakan kunci 19

Gambar 2.5 Tulisan yang menunjukan Heiroglyph..... 19

Gambar 2.6 Roda Kaisar 21

Gambar 2.7 M-94 22

Gambar 2.8 Antarmuka Visual Basic.NET 2010 36

Gambar 3.1 Use Case Diagram 40

Gambar 3.2 Activity Diagram 41

Gambar 3.3 Flowchart enkripsi algoritma Vigenere 43

Gambar 3.4 Flowchart dekripsi algoritma Vigenere 44

Gambar 3.5 Tampilan Menu Utama..... 44

Gambar 3.6 Tampilan Menu Vigenere Cipher 46

Gambar 3.7 Tampilan Menu Info..... 47

Gambar 3.8 Tampilan Menu About..... 47

Gambar 4.1 Halaman Menu Utama..... 51

Gambar 4.2 Halaman Info 51

Gambar 4.3 Halaman About..... 52

Gambar 4.4 Halaman kriptografi stream cipher algoritma Vigenere 53

Gambar 4.5 Halaman enkripsi algoritma Vigenere Cipher 54

Gambar 4.6 Halaman dekripsi algoritma Vigenere Cipher 55

DAFTAR ISI

KATA PENGANTAR ..... **i**

DAFTAR ISI ..... **iii**

DAFTAR GAMBAR ..... **vi**

DAFTAR TABEL ..... **vii**

BAB I PENDAHULUAN **1**

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah 2

1.4 Tujuan Penelitian 3

1.5 Manfaat Penelitian 3

BAB II LANDASAN TEORI **4**

2.1 Pencurian Informasi 4

2.1.1 Pencurian Nomor Kartu Kredit 6

2.1.2 Spoofing ATM 6

2.1.3 Pengambilan PIN 7

2.1.4 Pencurian Database 7

2.1.5 Uang Elektronik 8

2.2 Algoritma 8

2.2.1 Definisi Algoritma 9

2.2.2 Jenis-Jenis Algoritma 10

2.2.3 Analisis Kompleksitas Algoritma 13

2.3 Kriptografi 17

2.3.1 Sejarah Kriptografi 19
2.3.2 Kriptografi Simetris 22
2.4 Cipher Substitusi 23
2.5 Vigenere Cipher 24
2.6 Unified Modelling Language 25
2.6.1 Use Case Diagram 25
2.6.2 Activity Diagram 27
2.6.3 Class Diagram 29
2.6.4 Sequence Diagram 30
2.6.5 Flowchart 31
2.7 Visual Basic 34
2.7.1 Visual Basic.NET 35
2.7.2 Antarmuka Visual Basic.NET 35
BAB III METODE PENELITIAN 37
3.1 Tahapan Penelitian 37
3.2 Metode Pengumpulan Data 38
3.3 Perancangan Penelitian 39
3.3.1 Use Case Diagram 40
3.3.2 Activity Diagram 40
3.3.3 Flowchart Enkripsi 42
3.3.4 Flowchart Dekripsi 43
3.4 Desain Interface 44
3.4.1 Menu Utama 44
3.4.2 Menu Vigenere Cipher 45

3.4.3 Menu Info 46
3.4.4 Menu About 47
BAB IV HASIL DAN PEMBAHASAN 48
4.1 Spesifikasi Sistem 48
4.1.1 Spesifikasi Perangkat Keras 49
4.1.2 Spesifikasi Perangkat Lunak 49
4.2 Implementasi Antarmuka 50
4.2.1 Halaman Menu Utama 50
4.2.2 Halaman Info 51
4.2.3 Halaman About 52
4.2.4 Halaman Vigenere Cipher 52
4.2.5 Hasil Perhitungan Algoritma Vigenere 53
4.3 PengujianPerhitungan 55
BAB V PENUTUP 61
5.1 Kesimpulan 61
5.2 Saran 61
DAFTAR PUSTAKA

DAFTAR TABEL

Tabel 2.1 Elemen-Elemen Use Case 26

Tabel 2.2 Elemen-Elemen Activity Diagram 27

Tabel 2.3 Elemen-Elemen Class Diagram..... 29

Tabel 2.4 Elemen-Elemen Sequence Diagram..... 30

Tabel 2.5 Simbol-simbol Flowchart..... 33

Tabel 4.1 Spesifikasi perangkat keras 49

Tabel 4.2 Spesifikasi perangkat lunak..... 49

KATA PENGANTAR

Puji syukur kehadirat Allah SWT karena dengan anugrah dan hidyahNya penulis masih diberikan kesempatan untuk menyelesaikan skripsi ini dapat diselesaikan dengan baik dan sebagaimana mestinya. Skripsi ini berjudul ”IMPLEMENTASI PENGAMANAN DATA MENGGUNAKAN ALGORITMA *VIGENERE CIPHER*”.Penulis mengucapkan banyak terima kasih kepada banyak pihak yang telah membantu dalam penyelesaian penyusunan skripsi ini. Penulis ingin mengucapkan terima kasih kepada:

1. Orang tua saya yang telah mendukung saya untuk menyelesaikan skripsi ini.
2. Bapak Dr. H. Muhammad Isa Indrawan, S.E, M.M selaku Rektor Universitas Pembangunanan Panca Budi, Medan.
3. Bapak Ir. Bhakti Alamsyah, M.T, Ph.D., selaku Rektor I, Universitas Pembangunan Panca Budi, Medan
4. Ibu Sri Shindi Indira, ST., M.Sc., selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi, Medan.
5. Bapak Eko Hariyanto, S.Kom., M.Kom, selaku Ketua Program Studi Sistem Komputer Universitas Pembangunan Panca Budi,Medan.
6. Bapak Andysah Putera Utama Siahaan,S.Kom., M.Kom., selaku Dosen Pembimbing I yang telah memberikan arahan dan membimbing dalam peyelesaian skripsi ini.
7. Ibu Ika Devi Perwitasari, S.Kom., M.Kom, selaku Dosen Pembimbing II yang telah memberikan koreksi terhadap tata tulisuntuk penyelesaian skripsi ini.

8. Dosen-dosen pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi, Medan.
9. Seluruh staff dan karyawan pada Universitas Pembangunan Panca Budi, Medan.
10. Teman-teman penulis dari program studi Sistem Komputer Fakultas Ilmu Komputer Universitas Pembangunan Panca Budi, Medan

Penulis juga menyadari bahwa penyusunan skripsi ini belum sempurna baik dalam penulisan maupun isi disebabkan keterbatasan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca untuk kesempurnaan isi skripsi ini.

Medan, 8 Agustus 2020
Penulis

Saydil Amri
1514370066

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pencurian data adalah kejahatan untuk memperoleh informasi pribadi atau keuangan orang lain dengan tujuan untuk mengasumsikan nama atau identitas orang tersebut untuk melakukan transaksi atau pembelian. Pencurian data dilakukan dengan berbagai cara. Beberapa pencuri identitas menyaring sampah untuk mencari rekening bank dan laporan kartu kredit; metode lain yang lebih canggih melibatkan pengaksesan database perusahaan untuk mencuri daftar informasi pelanggan. Begitu mereka memiliki informasi yang mereka cari, pencuri identitas dapat merusak peringkat kredit seseorang dan kedudukan informasi pribadi lainnya.

Dalam melakukan pencurian, penjahat menggambarkan dirinya sebagai orang lain untuk mendapatkan informasi dari komputer sesorang. Pencurian dapat dilakukan di jaringan global. Banyak cara yang dapat dilakukan untuk menghindari pencurian informasi. Salah satunya dengan cara mengamankan informasi yang akan dipertukarkan pada jaringan internet. Teknik yang dapat digunakan untuk mencegah kejahatan adalah kriptografi. Kriptografi akan mengubah pesan atau informasi yang memiliki arti menjadi pesan yang tidak terbaca atau terenkripsi. Ada banyak metode yang dapat digunakan dalam menjalankan kriptografi. Salah satunya adalah dengan menggunakan algoritma

Vigenere Cipher. Algoritma ini bekerja dengan cara mensubstitusikan karakter dengan karakter lainnya sesuai dengan karakter yang masih dalam tabel ASCII.

Proses kriptografi tidak akan berjalan tanpa adanya suatu aplikasi pendukung. Aplikasi akan dibuat dan diprogram untuk menjalankan serangkaian perintah untuk proses enkripsi dan dekripsi tersebut. Hal ini untuk membuktikan kebenaran proses algoritma *Vigenere cipher* apa sudah sesuai dengan seharusnya. Program aplikasi akan dibuat menggunakan *Visual Studio* 2010. Berdasarkan latar belakang yang sudah dipaparkan, maka penulis tertarik untuk memilih judul "**Implementasi Pengamanan Data Menggunakan Algoritma *Vigenere Cipher***".

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 algoritma *Vigenere Cipher*?
2. Bagaimana menentukan blok kunci pada algoritma *Vigenere Cipher*?
3. Bagaimana menggunakan modulo 256 pada algoritma *Vigenere Cipher*?

1.3 Batasan Masalah

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

1. Kriptografi substitusi yang digunakan adalah algoritma *Vigenere Cipher*.
2. Kunci yang digunakan berupa karakter huruf dan angka.

3. Pesan yang digunakan pada proses enkripsi adalah pesan berbasis teks.

1.4 Tujuan Penelitian

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

1. Untukmelakukan proses enkripsi dan dekripsi dengan algoritma *Vigenere Cipher*.
2. Untuk menentukan blok kunci pada algoritma *Vigenere Cipher*.
3. Untuk menggunakan modulo 256 pada algoritma *Vigenere Cipher*.

1.5 Manfaat Penelitian

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

1. Menghindari pencurian data oleh pihak yang tidak bertanggung jawab.
2. Informasi yang dikirimkan tidak dapat dipahami oleh pihak yang tidak bertanggung jawab.
3. Memberikan pemahaman algoritma *Vigenere Cipher*

BAB II

LANDASAN TEORI

2.1 Pencurian Informasi

Ada semakin banyak kasus pencurian informasi selama beberapa tahun terakhir. Sementara semakin banyak langkah-langkah keamanan elektronik telah dilakukan untuk melindungi harta benda dan informasi orang-orang, teknologi baru ini memiliki bug dan cacat desain yang membuka seluruh dunia baru bagi penjahat yang maju secara teknologi(Yakub, 2012).

Kejahatan komputer adalah masalah yang semakin besar ukurannya karena komputer menjadi lebih luas dan saling berhubungan. Salah satu alasan kejahatan komputer nampak begitu penting adalah bahwa hal itu dapat melibatkan pemindahan sejumlah besar uang secara *virtual* secara instan, hanya karena begitu banyak catatan yang ada hanya atau terutama pada komputer. Namun, ini sekali lagi pandangan sempit tentang dampak kejahatan komputer. Ketika seseorang mendefinisikan kejahatan komputer sebagai kejahatan dimana komputer merupakan faktor utama, terlihat bahwa tidak hanya uang, tetapi juga anak-anak dan *privasi* berada dalam risiko. Baru-baru ini, sebuah situs web anti-aborsi radikal terbukti bersalah mengancam nyawa dokter yang melakukan. Ini merupakan cara baru menggunakan komputer untuk melakukan kejahatan yang berada di luar banyak definisi lama, tetapi jelas merupakan kegiatan yang sangat penting untuk dicegah di masa depan(Stallings, 2013).

Persentase populasi Amerika Serikat dan negara maju lainnya yang besar - dan terus meningkat - kini online. Banyak bisnis menggunakan komputer dalam rutinitas sehari-hari. Pemulihan baru-baru ini dari fiksi ilmiah di media mencerminkan bagaimana perubahan ini berdampak pada budaya kita. Kejahatan komputer bahkan terkait dengan sebagian besar masalah utama lainnya dalam berita komputer saat ini: hak kekayaan intelektual, pornografi anak, dan masalah privasi.

Bukti *anekdotal* menunjukkan bahwa penipuan, akses tidak sah, pornografi anak, dan kejahatan terkait merupakan mayoritas kejahatan komputer, dan ini semua termasuk dalam definisi kami. Namun, dalam situs web ini, kami terutama berfokus pada dua komponen pertama kejahatan komputer. Ini karena pornografi anak agak berbeda dari dua komponen lainnya dalam hal efeknya. Efek dari dua komponen pertama sebagian besar materi di alam, sedangkan pornografi anak adalah perhatian yang lebih humanistik. Kami tidak menyiratkan bahwa pornografi anak kurang penting atau bukan kejahatan komputer. Sebaliknya, kami hanya mempersempit fokus kami untuk menghasilkan gambaran yang lebih rinci tentang penipuan dan pembobolan komputer. Namun, kami masih menawarkan beberapa informasi tentang pornografi anak, dan kami mendorong mereka yang tertarik pada subjek untuk memeriksa halaman web berikut, sumber informasi yang sangat baik tentang pornografi anak dan langkah-langkah pencegahan(Nasution, Rossanty, Siahaan, & Aryza, 2018). Berikut ini adalah tindak kejahatan yang sering dilakukan oleh orang yang tidak bertanggung jawab.

2.1.1 Pencurian Nomor Kartu Kredit

Orang-orang menggunakan kartu kredit untuk semakin banyak pembelian mereka seiring berjalannya waktu. Ini membuka arena yang lebih besar dan lebih besar untuk penipuan kartu kredit. Kartu kredit sangat mudah digunakan secara curang karena tidak memerlukan nomor identifikasi tambahan untuk digunakan. Yang dibutuhkan oleh pencuri hanyalah informasi murni - mereka tidak memerlukan kartu itu, tetapi hanya nomor yang ada di kartu itu. Baru-baru ini, dengan orang-orang membelanjakan lebih banyak untuk pembelian yang ditransaksikan melalui internet, penipuan kartu kredit menjadi lebih mudah. Sekarang pencuri tidak perlu masuk dalam jarak 5.000 mil dari orang yang mereka curi. Yang mereka perlukan hanyalah situs web yang cepat dan kotor (yang dapat dihosting secara gratis, dan secara anonim) mengiklankan beberapa produk fiksi, dan termasuk formulir untuk membeli secara online. Seketika pelaku akan memiliki daftar nomor kartu kredit yang dikaitkan dengan nama dan alamat surat, siap digunakan untuk apa pun yang mereka inginkan.

2.1.2 Spoofing ATM

Penjahat ini telah menarik beberapa perampokan rumit yang mengesankan. Satu kelompok penjahat mendirikan mesin ATM yang benar-benar palsu di dalam sebuah mal di Connecticut. Itu terlihat dan berfungsi seperti yang asli, kecuali bahwa setelah memberikannya kartu Anda dan mengetik pin Anda, itu akan menolak layanan Anda mengatakan itu rusak. Itu kemudian memiliki catatan kartu dan nomor PIN dari semua orang yang mencoba menggunakan mesin. Pencuri

kemudian menggunakan mesin ATM yang sah di seluruh kota untuk menarik lebih dari $3.000 dari rekening ini.

2.1.3 Pengambilan PIN

Sekelompok penjahat lain menjelajahi daerah di seberang jalan dari ATM yang sibuk, mencari tempat yang sempurna untuk menyembunyikan kamera video yang diarahkan ke kunci-kunci pada mesin ATM. Mereka menemukan tempat seperti itu dan mengatur kamera mereka. Setelah setiap identifikasi nomor PIN berhasil yang mereka catat, salah satu anggota kelompok akan memeriksa tanda terima yang dibuang di ATM. Jika mereka menemukan satu, grup memiliki nomor kartu dan nomor PIN.

2.1.4 Pencurian Database

Semua kegiatan kriminal sebelumnya ditujukan untuk mengumpulkan basis data dari informasi yang diperoleh secara curang dari orang satu per satu. Ini membutuhkan waktu, dan orang-orang ini hanya memiliki waktu terbatas sebelum operasi mereka akan dikenali dan ditutup. Ini membatasi jumlah orang yang informasinya dapat diperoleh para penjahat ini. Namun, ada basis data besar dari jenis informasi yang telah dibangun secara lambat dan legal oleh perusahaan internet yang sopan dan sopan. Misalnya, Layanan Musik BMG memungkinkan pelanggan memberikan nomor kartu kredit mereka saat mendaftar, sehingga mereka tidak perlu repot setiap kali melakukan pembelian. Ada ribuan pengguna layanan ini, banyak di antaranya kemungkinan menggunakan fitur ini.

Kombinasikan ini dengan fakta bahwa ratusan sistem komputer diretas setiap hari, dan kami memiliki situasi di mana peretas dapat mencuri basis data ukuran informasi industri seperti ini, dan menjadi liar.

2.1.5 Uang Elektronik

Kita sudah berada di jalan menuju masyarakat bebas uang tunai. Orang sekarang menggunakan kartu ATM, kartu kredit, dan kartu cek untuk sebagian besar pembelian mereka. Ketika bergerak lebih jauh dari masyarakat uang kertas, ke ekonomi elektronik murni, jenis kejahatan baru akan muncul. Jenis-jenis apa yang sebenarnya akan tergantung pada bentuk-bentuk keamanan baru apa yang perlu dipecahkan oleh penjahat besok. Apakah orang akan mensintesis otorisasi suara? Atau menjalankan serangan replay pada pemindai retina? Atau bahkan belajar meniru gaya mengetik korban. Yang bisa diyakini, adalah bahwa penjahat masa depan, seperti orang-orang dari abad terakhir dan orang-orang hari ini, akan terus berinovasi.

2.2 Algoritma

Algoritma adalah seperangkat instruksi yang dirancang untuk melakukan tugas tertentu. Ini bisa berupa proses sederhana, seperti mengalikan dua angka, atau operasi yang rumit, seperti memutar file video terkompresi. Mesin pencari menggunakan algoritma kepemilikan untuk menampilkan hasil yang paling relevan dari indeks pencarian mereka untuk permintaan tertentu(Hidayat, 2012).

Dalam pemrograman komputer, algoritma sering dibuat sebagai fungsi. Fungsi-fungsi ini berfungsi sebagai program kecil yang dapat dirujuk oleh program yang lebih besar. Misalnya, aplikasi tampilan gambar dapat menyertakan pustaka fungsi yang masing-masing menggunakan algoritme khusus untuk membuat format file gambar yang berbeda. Program pengeditan gambar dapat berisi algoritma yang dirancang untuk memproses data gambar. Contoh algoritme pemrosesan gambar termasuk pemangkasan, pengubahan ukuran, penajaman, pengaburan, reduksi mata merah, dan peningkatan warna(Firmansyah, 2012).

Dalam banyak kasus, ada beberapa cara untuk melakukan operasi tertentu dalam program perangkat lunak. Oleh karena itu, programmer biasanya berusaha membuat algoritma yang seefisien mungkin. Dengan menggunakan algoritma yang sangat efisien, pengembang dapat memastikan program mereka berjalan secepat mungkin dan menggunakan sumber daya sistem minimal. Tentu saja, tidak semua algoritma diciptakan dengan sempurna untuk pertama kalinya. Oleh karena itu, pengembang sering meningkatkan algoritme yang ada dan memasukkannya dalam pembaruan perangkat lunak di masa mendatang. Ketika Anda melihat versi baru dari program perangkat lunak yang telah "dioptimalkan" atau memiliki "kinerja lebih cepat," sebagian besar berarti versi baru mencakup algoritma yang lebih efisien(Edraw, 2019).

2.2.1 Definisi Algoritma

Sebagai metode yang efektif, suatu algoritma dapat diekspresikan dalam jumlah ruang dan waktu yang terbatas dan dalam Bahasa formal yang terdefinisi

dengan baik untuk menghitung suatu fungsi. Mulai dari keadaan awal dan input awal (mungkin kosong), instruksi menjelaskan perhitungan yang, ketika dijalankan, berlanjut melalui sejumlah terbatas dari negara berturut-turut yang terdefinisi dengan baik, akhirnya menghasilkan "output" dan berakhir pada kondisi akhir akhir(Sumandri, 2017). Transisi dari satu negara ke negara lain tidak selalu bersifat deterministik; beberapa algoritma, yang dikenal sebagai algoritma acak, memasukkan input acak.Ada empat fitur utama dari algoritma dari definisi:

1. Algoritma bekerja untuk menghasilkan output tertentu.
2. Algoritma bekerja dengan saling terhubung dan berkelanjutan.
3. Algoritma adalah kumpulan dari perintah-perintah kecil untuk mencapai tujuan tertentu.
4. Hasil algoritma akan muncul ketika serangkaian proses sudah terlaksana dengan baik.

Pada prinsipnya, algoritma akan bekerja dengan cara yang masuk akal dan mengerjakan perintah-perintah untuk menghasilkan keluaran yang benar.

2.2.2 Jenis-Jenis Algoritma

Algoritme adalah serangkaian urutan instruksi atau tindakan yang berisi ruang terbatas atau urutan dan yang akan memberikan hasil untuk masalah tertentu dalam jumlah waktu terbatas. Ini adalah pendekatan logis dan matematis untuk memecahkan atau memecahkan masalah menggunakan metode apa pun

yang mungkin. Ada banyak jenis algoritme tetapi jenis algoritme yang paling mendasar adalah:

1. Algoritma *rekursif*

   Ini memecahkan kasus dasar secara langsung dan kemudian berulang dengan input yang lebih sederhana atau lebih mudah setiap kali (Nilai dasar ditetapkan di awal yang diakhiri algoritma). Ini digunakan untuk memecahkan masalah yang dapat dipecah menjadi masalah yang lebih sederhana atau lebih kecil dari jenis yang sama.

2. Algoritma pemrograman dinamis

   Algoritma pemrograman dinamis (juga dikenal sebagai algoritma optimasi dinamis) mengingat hasil masa lalu dan menggunakannya untuk menemukan hasil baru berarti memecahkan masalah kompleks dengan memecahnya menjadi kumpulan subproblem yang lebih sederhana, kemudian menyelesaikan masing-masing subproblem tersebut hanya sekali, dan menyimpannya solusi mereka untuk digunakan di masa depan alih-alih menghitung ulang solusi mereka lagi.

3. Algoritma *Backtracking*

   Bagaimana kalau belajar mengulang menggunakan contoh katakanlah ada suatu masalah "MONK" dan akan dibagi menjadi empat masalah yang lebih kecil "M, R , A, A". Mungkin masalahnya solusi dari masalah ini tidak diterima sebagai solusi dari "MONK ". Faktanya, tidak tahu yang mana tergantung. Jadi algoritma akan memeriksa masing-masing dari untaian tersebut satu per satu sampai ditemukan solusi untuk "MONK".

Jadi pada dasarnya algoritma berusaha memecahkan submasalah tetapi jika tidak mencapai solusi yang diinginkan akan membatalkan apa pun yang telah dilakukan dan mulai dari awal lagi sampai menemukan solusinya.

4. Algoritma *Divide and Conquer*

   Membagi dan menaklukkan terdiri dari dua bagian, pertama-tama, membagi masalah menjadi sub-masalah yang lebih kecil dari jenis yang sama dan menyelesaikannya secara rekursif dan kemudian menggabungkannya untuk membentuk solusi dari masalah asli.

5. Algoritma *Greedy*

   Algoritma *Greedy* adalah algoritma yang memecahkan masalah dengan mengambil solusi optimal di tingkat lokal (tanpa memperhatikan konsekuensi apa pun) dengan harapan menemukan solusi optimal di tingkat global. Algoritma Greedy digunakan untuk menemukan solusi optimal tetapi tidak perlu bahwa akan menemukan solusi optimal dengan mengikuti algoritma ini. Seperti ada beberapa masalah di mana solusi optimal tidak ada (saat ini) ini disebut masalah *NP-complete*.

6. Algoritma *Brute Force*

   Algoritma Brute Force hanya mencoba semua kemungkinan sampai solusi yang memuaskan ditemukan. Jenis algoritma seperti itu juga digunakan untuk menemukan solusi optimal (terbaik) karena memeriksa semua solusi yang mungkin. Dan juga digunakan untuk menemukan solusi yang

memuaskan (bukan yang terbaik), cukup berhenti segera setelah solusi untuk masalah ditemukan.

7. Algoritma acak

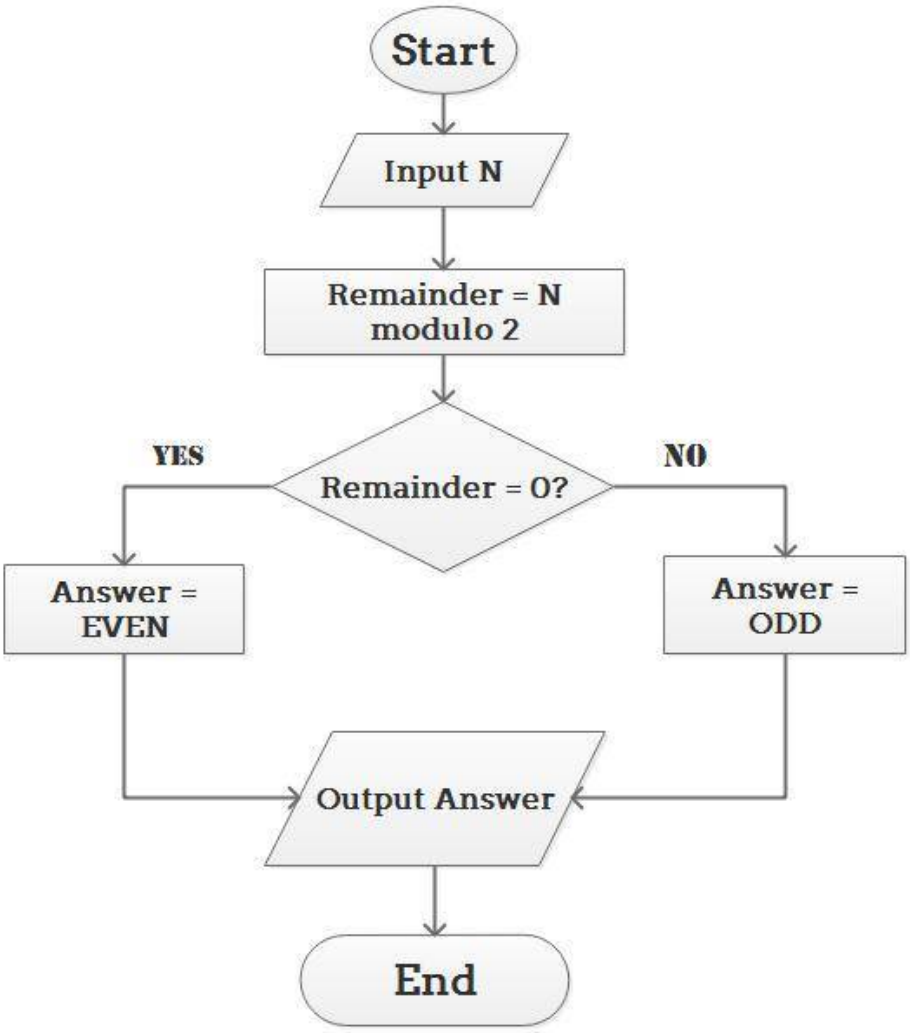
   Algoritma acak menggunakan nomor acak setidaknya sekali selama perhitungan untuk membuat keputusan.

2.2.3 Analisis Kompleksitas Algoritma

Analisis suatu algoritma mengacu pada proses memperoleh estimasi untuk waktu dan ruang yang dibutuhkan untuk menjalankan algoritma. Penting untuk memperkirakan waktu (mis., Jumlah langkah) dan ruang (mis., Jumlah variabel) yang dibutuhkan oleh algoritma. Mengetahui waktu dan ruang yang dibutuhkan oleh algoritma memungkinkan kita untuk membandingkan algoritma yang memecahkan masalah yang sama. Sebagai contoh, jika satu algoritma mengambil n langkah untuk menyelesaikan masalah dan algoritma lainnya mengambil n ^ 2 langkah untuk memecahkan masalah yang sama, kami lebih suka algoritma pertama. Estimasi waktu dan ruang yang diperlukan untuk menjalankan algoritma ini disebut kompleksitas waktu dan ruang dari algoritma.

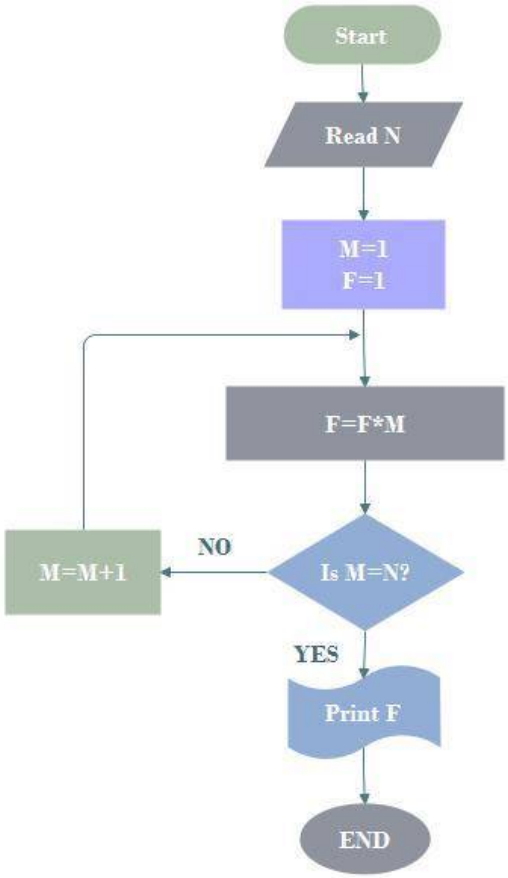
Waktu yang diperlukan untuk menjalankan suatu algoritma adalah fungsi dari input. Alih-alih berurusan langsung dengan input, parameter digunakan untuk mengkarakterisasi ukuran input. misalnya jika input adalah himpunan yang berisi n elemen, ukuran input n. Ada tiga kasus yang perlu dicatat tentang kompleksitas waktu suatu algoritma karena menentukan kompleksitas waktu yang tepat dari suatu algoritma dalam tugas yang sulit.

1. Kasus terburuk: f (n) diwakili oleh jumlah maksimum langkah yang diambil pada setiap instance ukuran n.
2. Kasing terbaik: f (n) diwakili oleh jumlah minimum langkah yang diambil pada setiap instance ukuran n.
3. Kasus rata-rata: f (n) diwakili oleh jumlah rata-rata langkah yang diambil pada setiap contoh ukuran n.



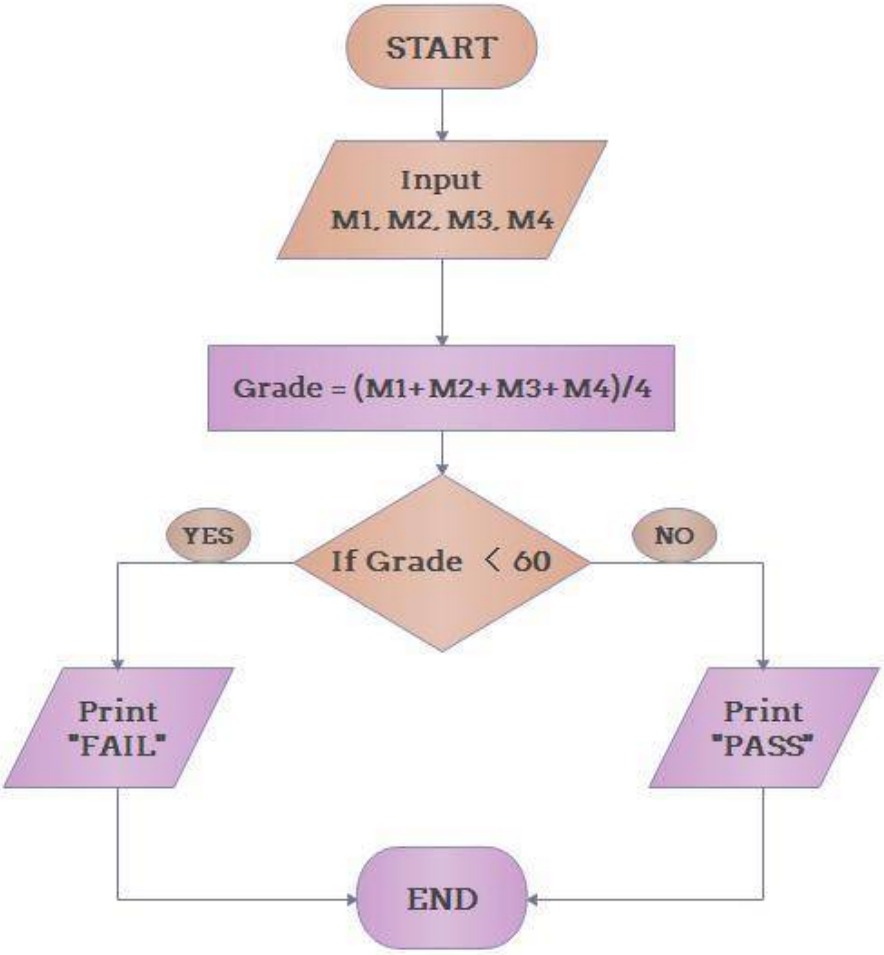
Gambar 2.1 Flowchart algoritma bidang matematika
Sumber:(Edraw, 2019)

Berikut ini adalah contoh flowchart aplikasi algoritma untuk bidang komputer yaitu menentukan hasil faktorial dari bilangan N.



Gambar 2.2 Flowchart algoritma bidang komputer
Sumber: (Edraw, 2019)

Berikut ini adalah contoh flowchart aplikasi algoritma untuk bidang pendidikan formal atau sekolah yaitu menentukan kelulusan mahasiswa.



Gambar 2.3 Flowchart algoritma bidang pendidikan
Sumber: (Edraw, 2019)

Gambar-gambar sebelumnya adalah diagram penggunaan algoritma pada beberapa bidan. Contoh-contoh tersebut memberikan demonstrasi yang jelas dari aplikasi algoritma dalam matematika, pemrograman komputer dan pendidikan.

2.3 Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan enkripsi dan dekripsi(Munir, 2006). Pesan yang akan dienkripsi disebut sebagai *plaintext* (teks biasa). Disebut demikian karena informasi ini dengan mudah dapat dibaca dan dipahami oleh siapa saja. Algoritma yang dipakai untuk mengenkripsi dan mendekripsi sebuah *plaintext* melibatkan penggunaan suatu bentuk kunci. Pesan *plaintext* yang telah dienkripsi (atau dikodekan) dikenal sebagai *ciphertext* (teks sandi). Di dalam kriptografi kita akan sering menemukan berbagai istilah atau terminologi.Beberapa istilah yang harus diketahui yaitu :

1. Pesan, *Plainteks*, dan *Cipherteks*

   Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah (*plaintext*) atau teks jelas (*cleartext*).

2. Pengirim dan Penerima

   Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan.

   Enkripsi dan dekripsi

3. Proses menyandikan plainteks menjadi *cipherteks* disebut enkripsi (*encryption*) atau *enciphering* (standard nama menurut ISO 7498-2). Sedangkan proses mengembalikan cipherteks menjadi plainteks semula

disebut dekripsi (*decryption*) atau deciphering (standard nama menurut ISO 7498-2).

4. *Cipher* dan kunci

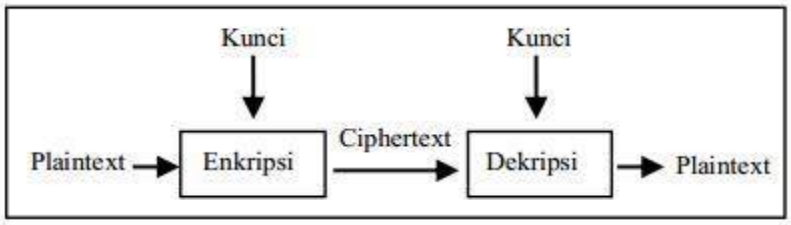
   Algoritma kriptografi disebut juga *cipher*, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen *plainteks* dan himpunan yang berisi *cipherteks*. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut. Misalkan P menyatakan plainteks dan C menyatakan cipherteks, maka :

   E(P) = C → fungsi enkripsi E memetakan P ke C

   D(C) = P → fungsi dekripsi D memetakan C ke P

   Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka persamaan D(E(P)) = P harus benar. Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini Algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya. Kunci (*key*) adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi. Kunci biasanya berupa string atau deretan bilangan. Dengan menggunakan kunci K, maka fungsi enkripsi

dan dekripsi dapat ditulis sebagai skema yang dijelaskan pada Gambar 2.4.



Gambar 2.4 Skema enkripsi dan dekripsi dengan menggunakan kunci
Sumber: (Edraw, 2019)

2.3.1 Sejarah Kriptografi

Kriptografi mempunyai sejarah yang sangat panjang. Kriptografi sudah digunakan 4000 tahun yang lalu, diperkenalkan oleh orang-orang Mesir lewat *hieroglyph.* Jenis tulisan ini bukanlah bentuk standar untuk menulis pesan(Ariyus, 2008).

Gambar 2.5 Tulisan yang menunjukan Heiroglyph
Sumber: (Ariyus, 2008)

Dikisahkan, pada Zaman Romawi Kuno, Pada Suatu saat Jelius Caesar ingin mengirimkan pesan rahasia kepada seseorang jendral di medan perang. Pesan tersebut dikirimkan melalui kurir. Karena pesan tersebut mengandung rahasia, jelius Cesar kemudian memikirkan bagaimana mengatasinya. Ia kemudian mengacak pesan tersbut hingga menjadi suatu pesan yang tidak dapat dipahami oleh siapapun terkecuali oleh jendralnya saja. Tentu pengirim ingin mengenkripsi suatu pesan menggunakan kode *Vigenere* dengan kunci *WARTHOG.*

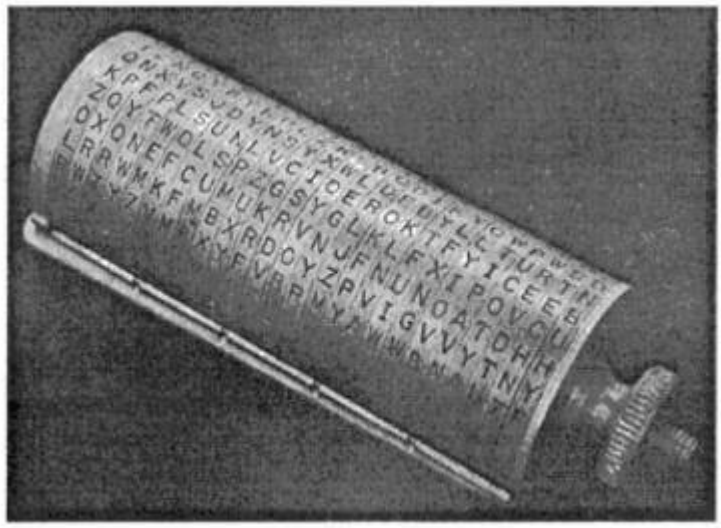
Untuk mengenkripsi huruf pertama, pengirim memutar *W* di potongan silindris dalam hingga berdampingan dengan *a* di silindris luar. Kemudian cari huruf teks-kode di potongan silindris dalam yang cocok dengan huruf teks-asli yang diinginkan di potongan silindris luar.

Selanjutnya mengirim mengenkripsi huruf kedua dengan memutar *A* di potongan silindris dalam hingga berdampingan dengan *a* di potongan silindris luar. Setelah itu cari huruf teks-kode di potongan silindris dalam yang cocok dengan huruf teks-asli yang dinginkan di potongan silindris luar.

Untuk seterusnya, ulang proses untuk huruf R,T,H,O,G dan kemudian ulang lagi dari W hingga seluruh pesan telah terenkripsi(EDUCBA, 2017).

Gambar 2.6 Roda Kaisar
Sumber: (Ariyus,2008)

Bentuk roda kode sejak versi Jefferson hingga M94 terdiri dari sejumlah potongan silindris yang tersusun di suatu sumbu besi. Setiap potongan silindris memiliki susunan alphabet secara acak di bagaian luar. Potongan-potongan silindris tersebut menjadi dalam mengenkripsi dan mendeskripsi pesan dari pihak penerima dan pengirim. Setiap potongan silindris dapat diputar untuk menyusun alphabet menjadi teks kode ataupun menjadi teks asli. Untuk mengenkripsi suatu pesan, pengirim M94 memiliki bentuk yang hampir sama dengan roda kode jaferson. Bedanya, M94 terbuat dari alumunium. Untuk potongan ke 17, susunan alfabetnya berupa ARMYOFTHEU. Susunan ARMY OFTHEUS menunjukan "Army Orgin of thw M94". M94 memiliki 100 pilihan potongan silindris, walau mungkin yang dipilih untuk digunakan untuk suatu kunci hanya 25 buah. Hal tersebut dapat memperbanyak kemungkinan solusi dari roda kunci.



Gambar 2.7 M-94
Sumber: (Ariyus,2008)

2.3.2 Kriptografi Simetris

Kriptografi Simetri (Kriptografi Kunci-*Privat*) Pada sistem kriptografi kunci-simetri, kunci untuk enkripsi sama dengan kunci untuk dekripsi, oleh karena itulah dinamakan kriptografi simetri. Keamanan sistem kriptografi simetri terletak pada kerahasiaan kuncinya. Ada banyak algoritma kriptografi modern yang termasuk ke dalam sistem kriptografi simetri, diantaranya adalah :

1. DES (Data Encryption Standard),
2. Blowfish,
3. Twofish,
4. Triple-DES,
5. IDEA,
6. Serpent,
7. AES (Advanced Encryption Standard).

Algoritma kriptografi (*cipher*) simetri dapat dikelompokkan menjadi dua kategori, yaitu:

1. *Cipher* aliran (*stream cipher*)

   Algoritma kriptografi beroperasi pada plainteks/cipherteks dalam bentuk bit tunggal, yang dalam hal ini rangkaian bit dienkripsikan/didekripsikan bit per bit.

2. *Cipher* blok (*block cipher*)

   Algoritma kriptografiberoperasi pada plainteks/cipherteks dalam bentuk blok bit, yang dalam hal ini rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya.

2.4 Cipher Substitusi

Dalam kriptografi, *cipher* substitusi adalah metode enkripsi dengan mana 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. Penerima menguraikan teks dengan melakukan substitusi terbalik. *Cipher* substitusi dapat dibandingkan dengan cipher transposisi. Dalam sandi transposisi, satuan *plaintext* disusun ulang dalam urutan yang berbeda dan biasanya cukup kompleks, tetapi satuan itu sendiri tidak berubah. Sebaliknya, dalam *cipher subtitusi*, unit *plaintext* dipertahankan dalam urutan yang sama dalam *ciphertext*, tetapi unit itu sendiri diubah(Wagner, 2003).

Ada sejumlah jenis *cipher* substitusi yang berbeda. Jika sandi beroperasi pada huruf tunggal, itu disebut sandi substitusi sederhana; sandi yang beroperasi pada kelompok huruf yang lebih besar disebut poligrafi. *Cipher monoalphabetic* menggunakan substitusi tetap atas seluruh pesan, sedangkan *cipher polyalphabetic* menggunakan sejumlah substitusi pada posisi yang berbeda dalam pesan, di mana unit dari *plaintext* dipetakan ke salah satu dari beberapa kemungkinan dalam *ciphertext* dan sebaliknya(Weerasinghe, 2013).

2.5 Vigenere Cipher

Cipher Vigenère adalah metode mengenkripsi teks alfabet dengan menggunakan serangkaian sandi Caesar yang terjalin berdasarkan huruf kata kunci. Meskipun 'Chiffre indéchiffrable' mudah dipahami dan diterapkan, selama tiga abad ia menolak semua upaya untuk mematahkannya. Metode ini adalah mengenkripsi teks alfabet dengan menggunakan serangkaian sandi Caesar yang terjalin, berdasarkan huruf kata kunci. Ini menggunakan bentuk substitusi *polyalphabetic*. Pertama kali dijelaskan oleh Giovan Battista Bellaso pada tahun 1553, sandi tersebut mudah dipahami dan di implementasikan, tetapi ia menolak semua upaya untuk memecahkannya hingga tahun 1863, tiga abad kemudian. Ini menghasilkan deskripsi *le chiffre indéchiffrable* (bahasa Perancis untuk 'sandi tidak dapat diuraikan'). Banyak orang telah mencoba menerapkan skema enkripsi yang pada dasarnya adalah *cipher Vigenère*. Pada tahun 1863, Friedrich Kasiski adalah yang pertama menerbitkan metode umum mengartikan sandi *Vigenère*.

Ciphertext dihasilkan dengan menerapkan operasi pergeseran karakter sebesar kunci yang telah ditentukan. Keuntungan menggunakan operasi substitusi adalah mudah dilakukan dan tidak menggunakan kinerja komputer yang berat, cukup dengan melakukan operasi yang sama pada saat dekripsi. Dengan kata lain:

$$plaintext + key = ciphertext$$

$$ciphertext + key = plaintext$$

2.6 Unified Modelling Language

Unified Modeling Language adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Enggineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa "pemrograman berorientasi objek" (OOP)(Wasserkrug et al., 2009).

Beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi(Sukmawati & Priyadi, 2019).

2.6.1 *Use Case* Diagram

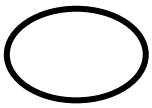
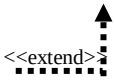
Use case diagram adalah abstraksi dari interaksi antara sistem dan aktor. *Use case diagram* bekerja dengan cara mendeskripsikan tipe interaksi antara user

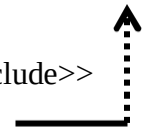
sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case diagram* berguna dalam tiga hal:

1. Menjelaskan fasilitas yang ada (*requirement*).
2. Komunikasi dengan klien.
3. Membuat *test* dari kasus-kasus secara umum.

Adapun simbol-simbol dalam *Use Case Diagram* yaitu:

Tabel 2.1 Elemen-Elemen *Use Case*

SIMBOL	NAMA	KETERANGAN
	*Actor*	Menspesifikan himpunan peran yang pengguna mainkan ketika berintraksi dengan *use case*
	*UseCase*	Deskripsikan urutan aksi-sistem yang menghasilkan suatu hasil terukur
	*System*	Menspesifikan paket yang menampilkan system secara terbatas
	*Association*	Simbol yang menghubungkan antara satu dengan objek lainya
	*Dependency*	Hubungan dimna perubahan yang terjadi pada suatu elemen mandiri(independent) akan mempengaruhi elemn yang tidak mandiri
	*Generalization*	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada tidak mandiri
	*Extend*	Menspesifkan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan

<<include>> 	*Include*	Menspesifikasikan bahwa *usecase* sumber Secara eksplisit

Sumber: (Kurniawan, 2018)

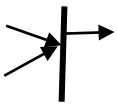
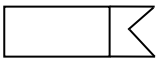
2.6.2 *Activity* Diagram

Activity diagram menyediakan analis dengan kemampuan untuk memodelkan proses dalam suatu sistem informasi. *Activity* diagram dapat digunakan untuk alur kerja model, *usecaseindividual*, atau logika keputusan yang terkandung dalam metode individual. *Activity* diagram juga menyediakan pendekatan untuk proses pemodelan paralel(Ladjamudin, 2005).

Pada dasarnya, diagram aktivitas canggih dan merupakan diagram aliran data yang terbaru. Secara teknis, diagram aktivitas menggabungkan ide-ide proses pemodelan dengan teknik yang berbeda termasuk model cara, statecharts. *Activity* diagram mempunyai beberapa elemen dalam memodelkan sebuah sistem, yaitu:

Tabel 2.2 Elemen-Elemen *Activity* Diagram

SIMBOL	NAMA	KETERANGAN
	*Action State*	Menandakan sebuah aktivitas
	*Initial State*	Titik awal untuk memulai suatu aktivitas
	*Final State*	Titik akhir untuk mengakhiri aktivitas

	*Decision*	Pilihan untuk mengambil keputusan
	*Flow Final*	Untuk mengakhiri suatu aliran
	*Transition*	Menunjukkan aktifitas selanjutnya setelah aktifitas sebelumnya
	*Synchronization*	Dibagi menjadi 2 yaitu fork dan join: fork digunakan untuk memcah behavior menjadi actifity atau action yang pararel, sedangkan join untuk menggabungkan kembali actifity atau action yang pararel
	Swimlane	Untuk melakukan partisi atau pembagian
	*Signal Accept State*	Tanda penerimaan
	*Signal Send State*	Tanda penerimaan

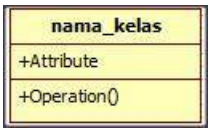
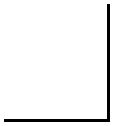
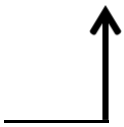
Sumber : (Kurniawan,2018)

2.6.3 *Class* Diagram

Tujuan utama dari *class* diagram adalah untuk menciptakan sebuah kosa kata yang digunakan oleh analis dan pengguna. *Class* diagram biasanya merupakan hal-hal, ide-ide atau konsep yang terkandung dalam aplikasi. Misalnya, jika sedang membangun sebuah aplikasi penggajian, diagram kelas mungkin akan berisi kelas yang mewakili hal-hal seperti karyawan, cek, dan pendaftaran gaji.

Class diagram juga akan menggambarkan hubungan antara kelas. Berikut komponen-komponen yang ada pada *class* diagram.

Tabel 2.3 Elemen-Elemen *Class* Diagram

SIMBOL	NAMA	KETERANGAN
	*Class*	Kelas pada struktur sistem
	*Association*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai dengan multiplicity
	*Directed Association*	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi berarah biasanya juga disertai dengan multiplicity
	*Dependency*	Relasi antar kelas dengan makna kebergantungan antar kelas.

Sumber: : (Kurniawan, 2018)

2.6.4 Sequence Diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*. Berikut komponen-komponen yang ada pada *sequence* diagram.

Tabel 2.4 Elemen-Elemen Sequence Diagram

SIMBOL	NAMA	KETERANGAN
	*Objek*	Menggambarkan objek/orang yang berintraksi di dalam sistem
	*Initial State*	Menggambarkan pengiriman pesan
	*Self Stimulus*	Menyatakan suatu objek mengirimkan pesan untuk menjalankan oprasi yang ada pada objek lain.

Sumber: (Kurniawan, 2018)

2.6.5 Flowchart

Flowchart adalah jenis diagram yang mewakili alur kerja atau proses. Diagram alir juga dapat didefinisikan sebagai representasi diagram dari suatu algoritma, pendekatan langkah demi langkah untuk menyelesaikan suatu tugas. Diagram alur menunjukkan langkah-langkah sebagai kotak dari berbagai jenis, dan urutannya dengan menghubungkan kotak-kotak dengan panah. Representasi diagram ini menggambarkan model solusi untuk masalah yang diberikan. *Flowchart* digunakan dalam menganalisis, merancang, mendokumentasikan, atau mengelola suatu proses a tau program di berbagai bidang. *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 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. Sebuah keputusan, biasanya dilambangkan sebagai berlian.

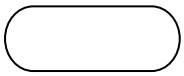
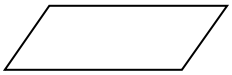
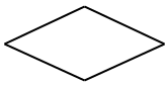
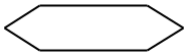
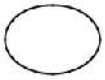
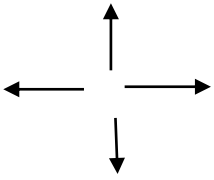
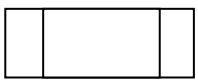
Diagram alir digambarkan sebagai "lintas fungsional" ketika bagan dibagi menjadi bagian *vertikal* atau *horizontal* yang berbeda, untuk menggambarkan control unit organisasi yang berbeda. Simbol yang muncul di bagian tertentu

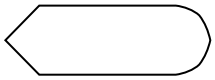
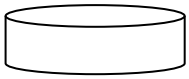
berada dalam kendali unit organisasi itu. *Flowchart* lintas fungsional memungkinkan penulis untuk menemukan tanggung jawab untuk melakukan suatu tindakan atau membuat keputusan dengan benar, dan untuk menunjukkan tanggung jawab masing-masing unit organisasi untuk bagian-bagian berbeda dari satu proses tunggal(Nakatsu, 2009).

Struktur grafik yang mendasari diagram alur adalah grafik aliran, yang mengabstraksi jenis simpul, isinya, dan informasi tambahan lainnya.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 control , 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. 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-simbol Flowchart

NO	SIMBOL	FUNGSI
1.		**Terminal**, untuk memulai atau mengakhiri suatu program
2.		**Proses**, suatu symbol yang menunjukkan setiap pengolahan yang dilakukan
3.		***Input-output***, untuk memasukkan menunjukkan hasil dari suatu proses
4.		***Decision***, suatu kondisi yang akan me4ngahsilkan beberapa kemungkinan jawaban atau pilihan
5.		***Preparation***, suatu symbol yang menyediakan tempat pengolahan
6.		***Connector***, suatu prosedur penghubung yang akan masuk atau keluar melalui symbol ini dalam lembar yang sama
7.		***Off-page Connector***, merupakan symbol masuk atau keluarnyasuatu prosedur pada lembaran kertas lainya
8.		***Arus/Flow***, 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.		***Predefined Process***, Untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur

10.		Simbol untuk output, yang ditunjukan kesuatu device, seperti printer, dan sebagainya
11.		Penyimpanan file secara permanen
12.		Menunjukkan input / Output Hardisk (media penyimpanan)

Sumber: (Kurniawan,2018)

2.7 Visual Basic

Visual Basic (VB) adalah bahasa pemrograman yang digerakkan oleh peristiwa dan lingkungan dari *Microsoft* yang menyediakan antarmuka pengguna grafis (GUI) yang memungkinkan programmer untuk memodifikasi kode hanya dengan menyeret dan menjatuhkan objek dan menentukan perilaku dan penampilan mereka. VB berasal dari bahasa pemrograman BASIC dan dianggap event-driven dan berorientasi objek.VB dimaksudkan agar mudah dipelajari dan cepat untuk menulis kode Akibatnya, kadang-kadang disebut sistem pengembangan aplikasi cepat (RAD) dan digunakan untuk prototipe aplikasi yang nantinya akan ditulis dalam bahasa yang lebih sulit tetapi efisien(Lee, 2014).

Versi terakhir VB, Visual Basic 6, dirilis pada tahun 1998, tetapi sejak itu telah digantikan oleh VB. NET, Visual Basic for Applications (VBA) dan Visual Studio .NET. VBA dan Visual Studio adalah dua kerangka kerja yang paling umum digunakan saat ini. VB adalah alat pengembangan berbasis GUI yang menawarkan RAD lebih cepat daripada kebanyakan bahasa pemrograman lainnya.

VB juga memiliki fitur sintaksis yang lebih mudah daripada bahasa lain, lingkungan visual yang mudah dipahami dan konektivitas basis data yang tinggi.

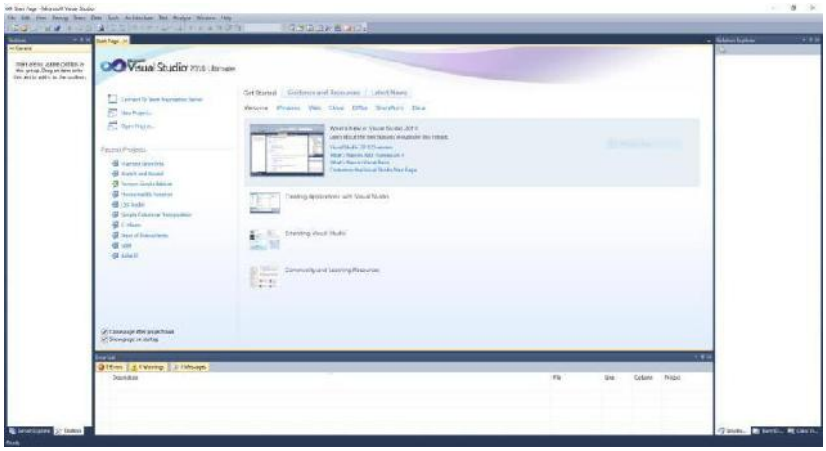
2.7.1 Visual Basic.NET

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi Web. *Visual Studio* mencakup kompiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++, Visual C#, Visual Basic, Visual Basic.NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.*

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows)* ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

2.7.2 Antar muka *Visual Basic.NET*

Visual Basic.Net memiliki beberapa versi. Berikut ini adalah tampilan dari *Visual Basic.Net versi* 2010.



Gambar 2.8Antarmuka Visual Basic.NET 2010
Sumber: (Rahmel, 2008)

BAB III

METODE PENELITIAN

1.1 Tahapan Penelitian

Penelitan ini akan dikategorikan kepada beberapa bagian. Studi ini dilakukan berdasarkan pesan teks yang berupa masukan untuk proses enkripsi dan dekripsi. Algoritma *Vigenere cipher* akan diciptakan dan dialkukan pengujian berdasarkan beberapa bagian agar output yang dihasilkan sesuai dengan yang sebelumnya direncanakan dan tidak berbeda dengan perhitungan manual. Tahapan berikut ini adalah proses yang dilakukan dalam penelitian ini:

1. Studi Literatur

   Bagianakan dilakukan pencarian literatur dan buku yang berhubungan dengan metode *Vigenere cipher*. Sumber-sumber bahan bacaan dapat diperoleh dari web dan buku teks.

2. Analisa

   Bagian ini menjelaskan proses analisa permasalahan dan penentuan cara penyelesaian terhadap masalah yang dialami. Proses ini melakukan analisis terhadap permasalahan yang terjadi dan bagaimana permasalahan tersebut akan diselesaikan.

3. Pembahasan

   Bagian ini membahas tentang perhitungan algoritma *Vigenere cipher* pada proses enkripsi dan dekripsi. Hasil perhitungan akan dissuaikan dengan uji manual.

4. Implementasi dan pengujian

   Bagian ini dilakukan pengujian hasil yang dikeluarkan oleh program aplikasi yang sudah dibuat menggunakan *Microsoft Visual Basic.Net* 2010. Hasil akan dibandingkan dengan perhitungan yang dilakukan secara manual.

3.2 Metode Pengumpulan Data

Metode pengumpulan data bertujuan untuk memperoleh hasil berdasarkan perancangan yang sudah dilakukan. Tahapan yang dilakukan dalam melakukan pengumpulan data adalah sebagai berikut:

1. Studi Kepustakaan

   Studi kepustakaan dilakukan dengan cara mengumpulkan sumber, mempelajari, dan membaca berbagai materi seperti buku, jurnal, majalah, dan internet.

2. Wawancara

   Wawancara dilakukan untuk mendapatkan jawaban yang lebih baik dari orang yang memiliki pengetahuan lebih tentang kriptografi dan algoritma Vigenere Cipher. Hasil wawancara berguna untuk menjadi referensi dalam pembuatan program aplikasi.

3. Pengamatan

   Pengamatan dilakukan dengan cara megnamati hasil program aplikasi. Uji coba dilakukan beberapa kalai untuk melihat konsistensi hasil.

3.3 Perancangan Penelitian

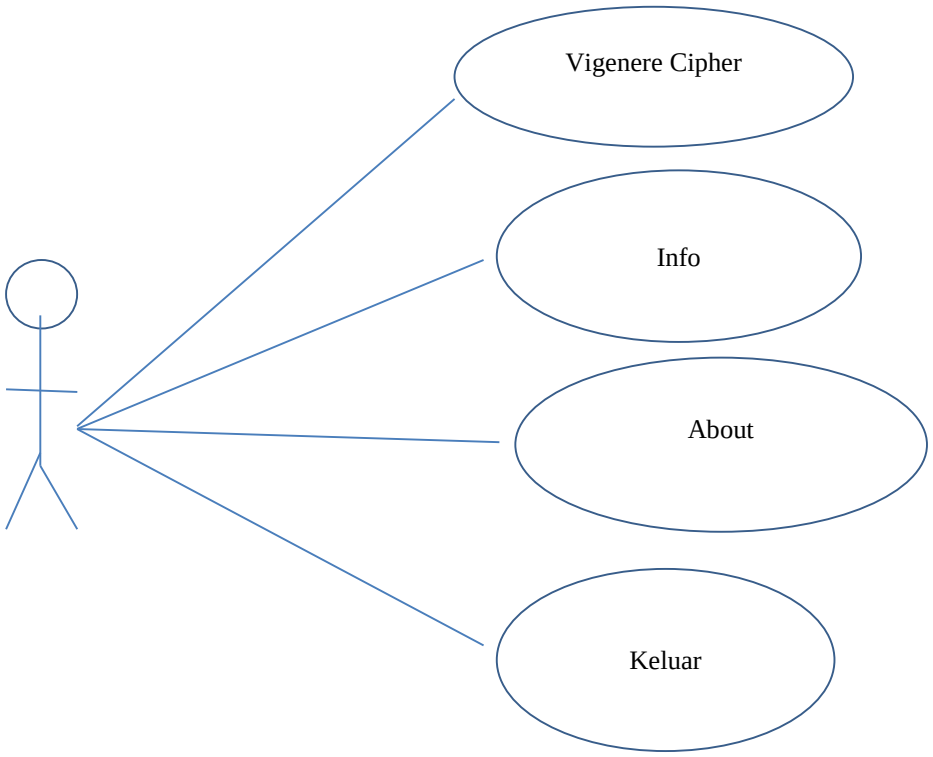
Desain penelitian didefinisikan sebagai kerangka kerja metode dan teknik yang dipilih oleh seorang peneliti untuk menggabungkan berbagai komponen penelitian dengan cara yang cukup logis sehingga masalah penelitian ditangani secara efisien. Ini memberikan wawasan tentang "bagaimana" melakukan penelitian menggunakan metodologi tertentu. Setiap peneliti memiliki daftar pertanyaan penelitian yang perlu dinilai. Itu bisa dilakukan dengan desain penelitian. Sketsa tentang bagaimana penelitian harus dilakukan dapat disusun menggunakan desain penelitian. Oleh karena itu, studi penelitian pasar akan dilakukan berdasarkan desain penelitian.

Desain topik penelitian digunakan untuk menjelaskan jenis penelitian dan juga sub-jenisnya.Jenis masalah penelitian yang dihadapi organisasi akan menentukan desain penelitian dan bukan sebaliknya. Variabel, alat yang ditunjuk untuk mengumpulkan informasi, bagaimana alat akan digunakan untuk mengumpulkan dan menganalisis data dan faktor-faktor lain diputuskan dalam desain penelitian berdasarkan teknik penelitian yang diputuskan. Desain penelitian yang berdampak biasanya bisa menciptakan minimum dalam data dan meningkatkan kepercayaan pada informasi penelitian yang dikumpulkan dan dianalisis. Desain penelitian yang menghasilkan margin kesalahan terkecil dalam penelitian eksperimental dapat disebut-sebut sebagai yang terbaik.

Pada bagian ini akan dijelaskan perancangan penelitian untuk mendefinisikan setiap bagian yang berfungsi untuk melengkapi alur pengguna terhadap gambaran penelitian yang akan dibuat.

3.3.1 *Use Case* Diagram

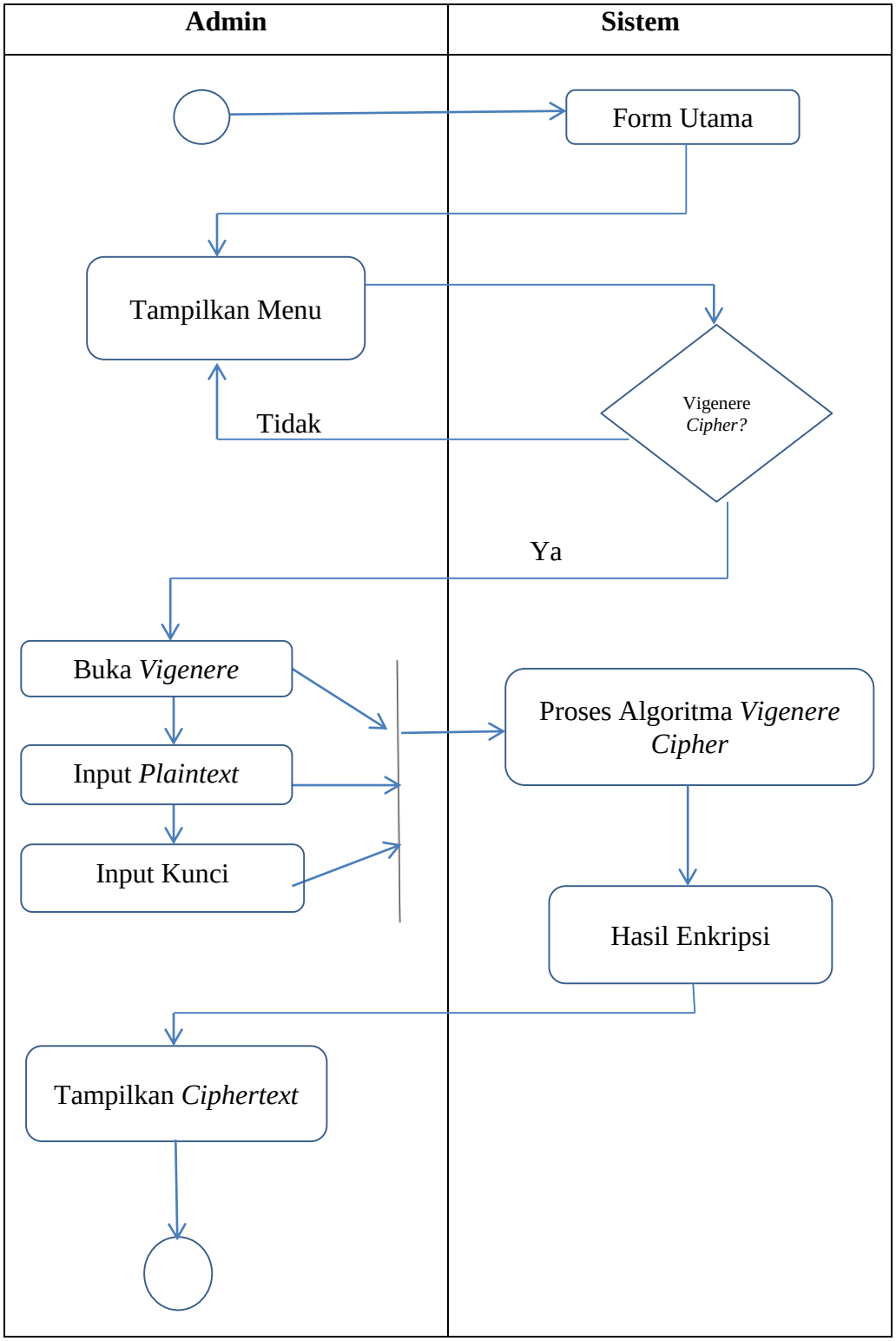
Use Case adalah penjelasan fungsi dari sebuah sistem dari segi pengguna. *Use Case* bekerja dengan cara menjelaskan interaksi antar *User* (pengguna) dengan sistemnya sendiri melalui sebuah bagan bagaimanasuatu sistem dipakai. Gamabr 3.1 adalah perancangan *Use Case* untuk admin dari algoritma *Vigenere Cipher*.



Gambar 3.1 *Use Case* Diagram

3.3.2 *Activity* Diagram

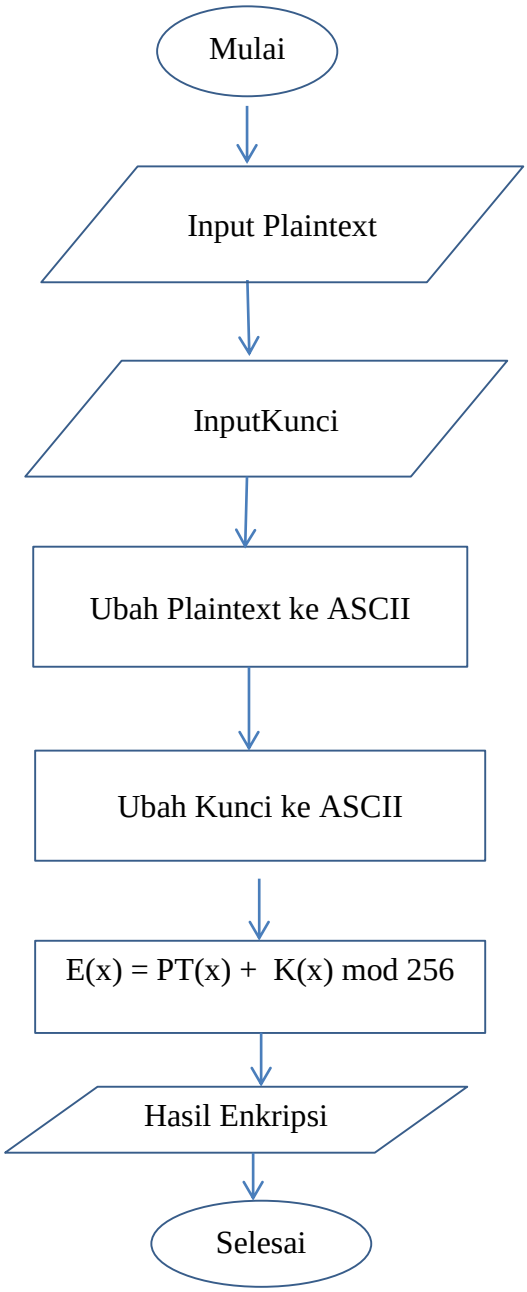
Activity diagramakan menggambarkan alur aktifitas dari sistem, untuk *Activity diagram* dari kriptrografi simetris dengan menggunakan algoritma *Vigenere cipher*. Gambar berikut ini akan menjelaskan *Activity diagram* tersebut.



Gambar 3.2 *Activity* Diagram

3.3.3 *Flowchart* Enkripsi

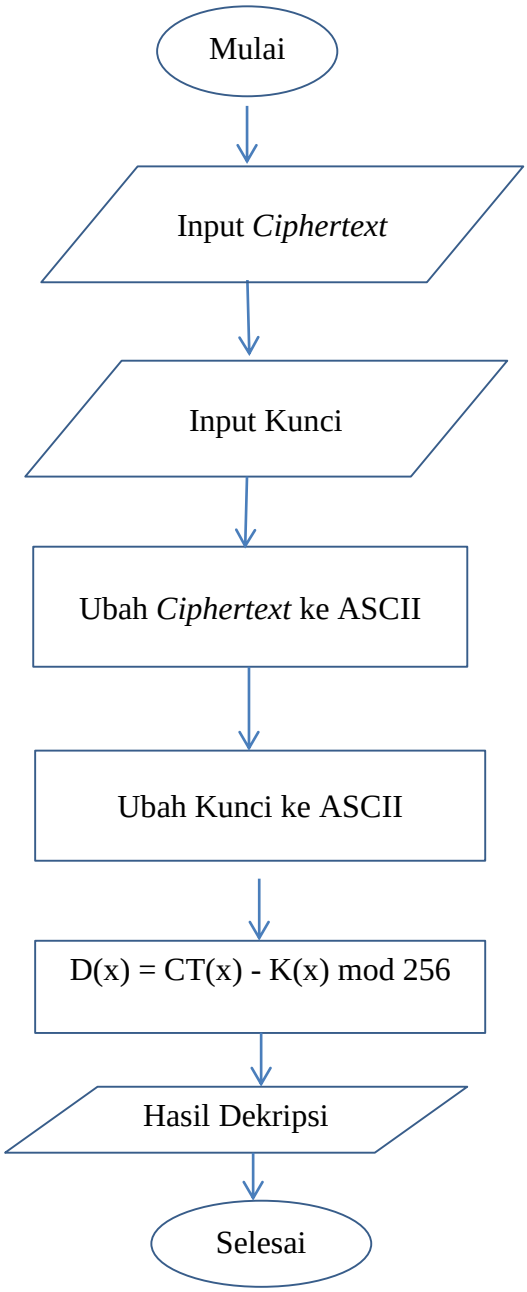
Flowchart enkripsi akan menerangkan proses enkripsi dengan metode substitusi dengan algoritma *Vigenere Cipher*. *Flowchart* enkripsi dapat dilihat pada gambar berikut ini.



Gambar 3.3 *Flowchart* enkripsi algoritma *Vigenere*

3.3.4 *Flowchart* Dekripsi

Flowchart dekripsi akan menjelaskan alur dari proses dekripsi dengan metode substitusi dengan algoritma *Vigenere Cipher*. *Flowchart* dekripsi dapat dilihat pada gambar berikut ini.



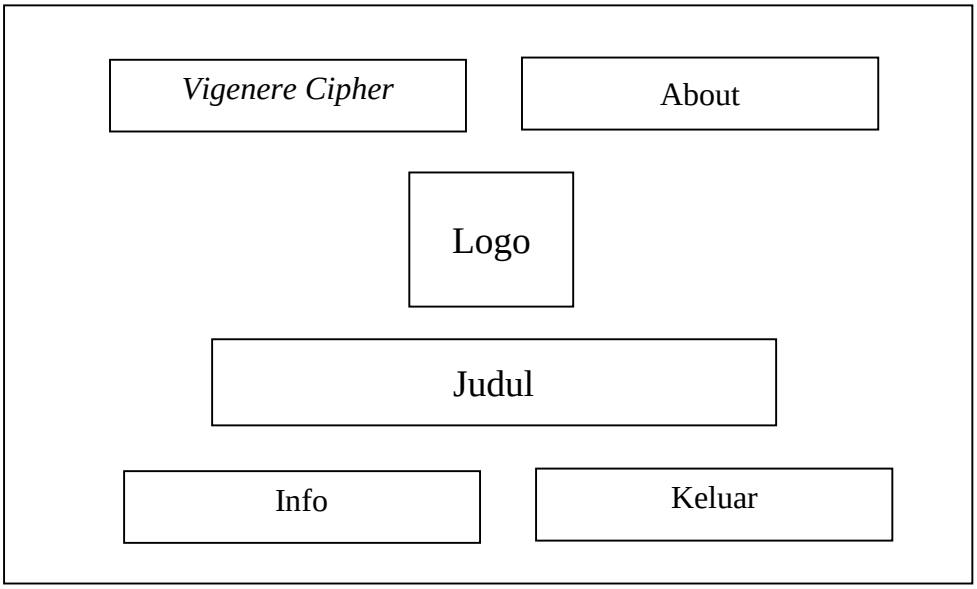
Gambar 3.4 *Flowchart* dekripsi algoritma *Vigenere*

3.4 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. Tahapan berikut ini merupakan desain *interface* dari menu-menu yang ada pembuatan program aplikasi algoritma *Vigenere Cipher*.

3.4.1 Menu Utama

Menu utama adalah tampilan yang pertama sekali muncul pada saat progam aplikasi dijalankan. Tampilan pada gambar berikut ini adalah hasil perancangan menu utama yang memiliki beberapa komponen lainnya.



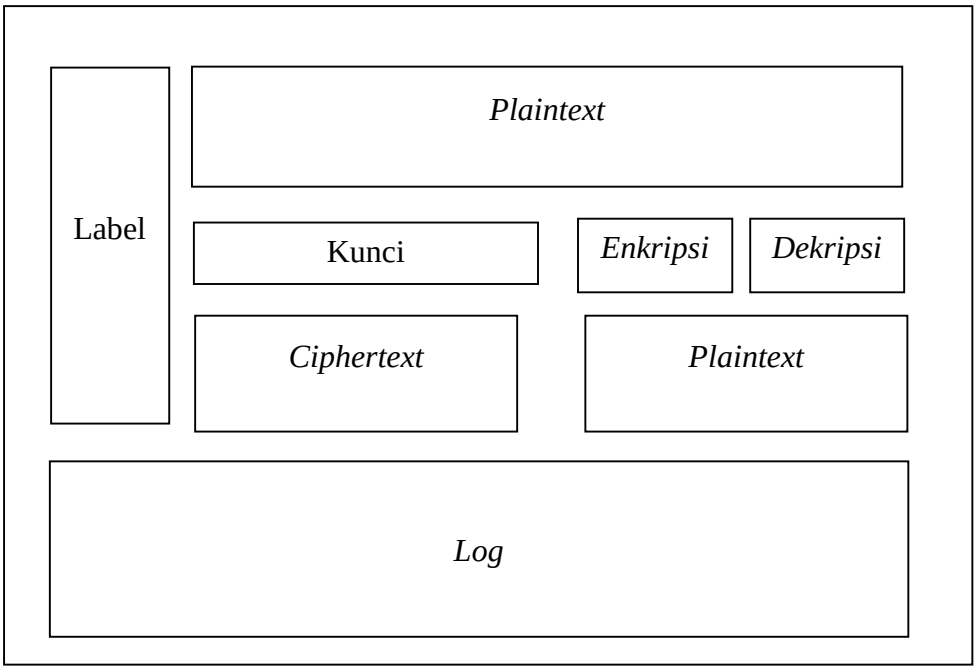
Gambar 3.5 Tampilan Menu Utama

Menu ini memiliki berapa sub-menu antara lain:

1. *Vigenere Cipher*
2. *About*
3. Logo
4. *Info*
5. Keluar

3.4.2 Menu *Vigenere Cipher*

Menu ini adalah yang terpenting dalam program aplikasi karena menjalankan tugas utama yaitu *Vigenere Cipher*. Menu ini berfungsi untuk melakukan proses enkripsi dan dekripsi. Menu ini terdiri dari input, proses, *output* dan riwayat perhitungan. Gambar 3.6 adalah tampilan menu ini.



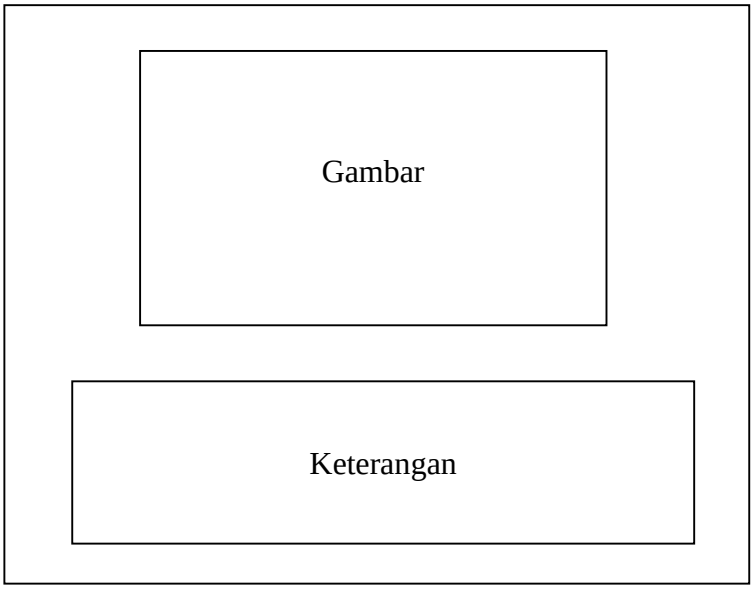
Gambar 3.6 Tampilan Menu *Vigenere Cipher*

Menu algoritma *Vigenere Cipher* memiliki beberapa bagian antara lain:

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

3.4.3 Menu Info

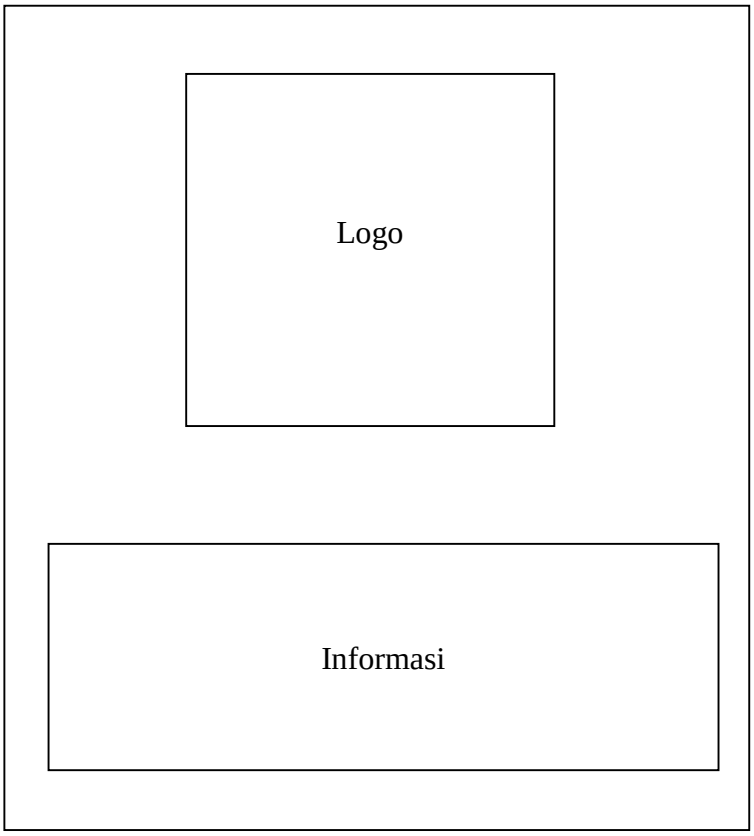
Menu ini menampilkan tentang keterangan singkat algoritma Vigenere *Cipher*. Menu ini memiliki dua buah objek, yaitu objek gambar dan keterangan. Gambar berikut ini adalah hasil perancangan menu Info.



Gambar 3.7 Tampilan Menu Info

3.4.4 Menu *About*

Menu ini menampilkan keterangan tentang penulis. Menu ini teridiri dari logo Universitas Pembangunan Panca Budi dan beberapa keterangan singkat. Menu ini terdiri dari dua objek, yaitu logo dan informasi. Gambar berikut ini adalah hasil tampilan dari menu About.



Gambar 3.8 Tampilan Menu About

BAB IV

HASIL DAN PEMBAHASAN

Implementasi adalah pelaksanaan, pelaksanaan, atau praktik rencana, metode, atau desain, ide, model, spesifikasi, standar, atau kebijakan apa pun untuk melakukan sesuatu. Dengan demikian, implementasi adalah tindakan yang harus mengikuti pemikiran awal apa pun agar sesuatu benar-benar terjadi. Dalam konteks teknologi informasi, implementasi perangkat lunak atau perangkat keras mencakup semua proses purnajual yang terlibat dalam sesuatu yang beroperasi dengan baik di lingkungannya, termasuk menganalisis persyaratan, instalasi, konfigurasi, penyesuaian, berjalan, pengujian, integrasi sistem, pelatihan pengguna, pengiriman dan pembuatan yang diperlukan. perubahan.

1.1 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 *Vigenere Cipher* pada metode kriptografi substitusi membutuhkan perangkat keras untuk menjalankan sistem. Hal ini sebagai sarana pendukung utama. Tabel 4.1 adalah spesifikasi perangkat keras yang digunakan pada 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	14inch

4.1.2 Spesifikasi Perangkat Lunak

Tahap spesifikasi perangkat lunak memiliki tujuan, deskripsi kebutuhan dan persiapan validasi aplikasi perangkat lunak. Deskripsi kebutuhan memunculkan file spesifikasi aplikasi perangkat lunak.Kebutuhan akan perangkat lunak sebagaisarana non-fisik sangat mendukuing hasil keluaran.Tabel 4.2 adalah spsesifikasi perangkatlunak yang digunakan pada penelitian ini.

Tabel 4.2 Spesifikasi perangkat lunak

No.	Nama Komponen	Spesifikasi
1	Sistem Operasi	Windows 1064 Bit
2	IDE Pemrograman	Microsoft Visual Basic.NET 2010
3	Tangkap Gambar	Snipping Tool
4	Data Editor	Microsoft Excel

4.2 Implementasi Antarmuka

Menerapkan desain berarti benar-benar melakukan pekerjaan untuk mengubah ide (desain) menjadi sesuatu yang nyata. Proses mendesain apa pun akan memiliki langkah-langkah umum termasuk mengumpulkan persyaratan, mengidentifikasi solusi yang mungkin, menganalisis solusi tersebut, dll. Agar proses implementasi berhasil, banyak tugas antara berbagai departemen perlu diselesaikan secara berurutan. Perusahaan berusaha untuk menggunakan metodologi yang telah terbukti dan meminta bantuan profesional untuk membimbing mereka melalui penerapan suatu sistem tetapi kegagalan dari banyak proses implementasi sering berasal dari kurangnya perencanaan yang akurat pada tahap awal proyek karena sumber daya yang tidak memadai atau masalah tak terduga yang muncul .

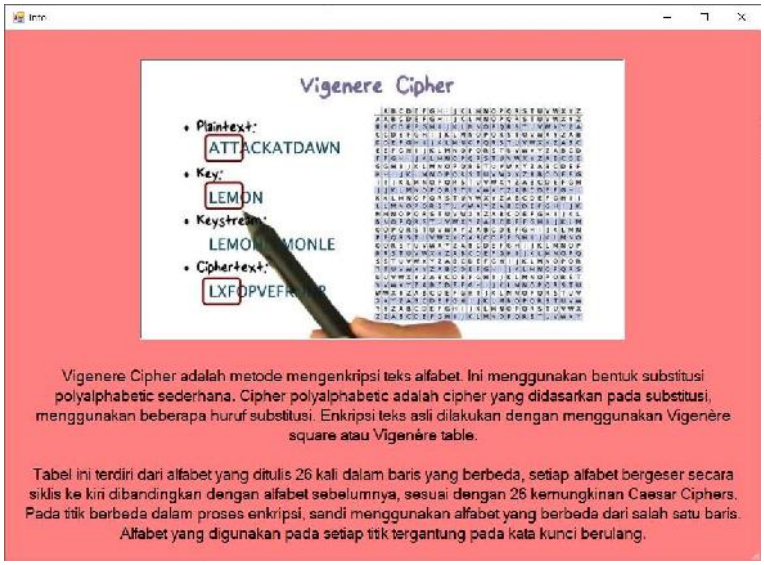
4.2.1 Halaman Menu Utama

Halaman menu utama adalah tampilan yang pertama sekali muncul pada saat program aplikasi dibuka. Ada beberapa menu yang akan dimunculkan untuk menentukan pilihan menu berikutya. Pengguna akan memilih ke bagian mana pengguna tersebut ingin jalankan. Menu utama bertujuan untuk memberikan pilihan fungi dari fasilitas-fasilitas yang ditawarkan pada suatu program aplikasi. Menu utama pada penelitian inii terdiri dari tiga buah sub-menu dan satu buah tombol untuk keluar dari aplikasi tersebut. Gambar 4.1 adalah hasil tampilan menu utama.

Gambar 4.1 Halaman Menu Utama

4.2.2 Halaman Info

Halaman info adalah menu yang menampilkan penjelasan singkat tentang algoritma Vigenere Cipher. Halaman ini akan menampilkan sebuah gambar dan sebuah keterangan. Gambar 4. 2 adalah hasil tampilan dari halaman info.



Gambar 4.2 Halaman Info

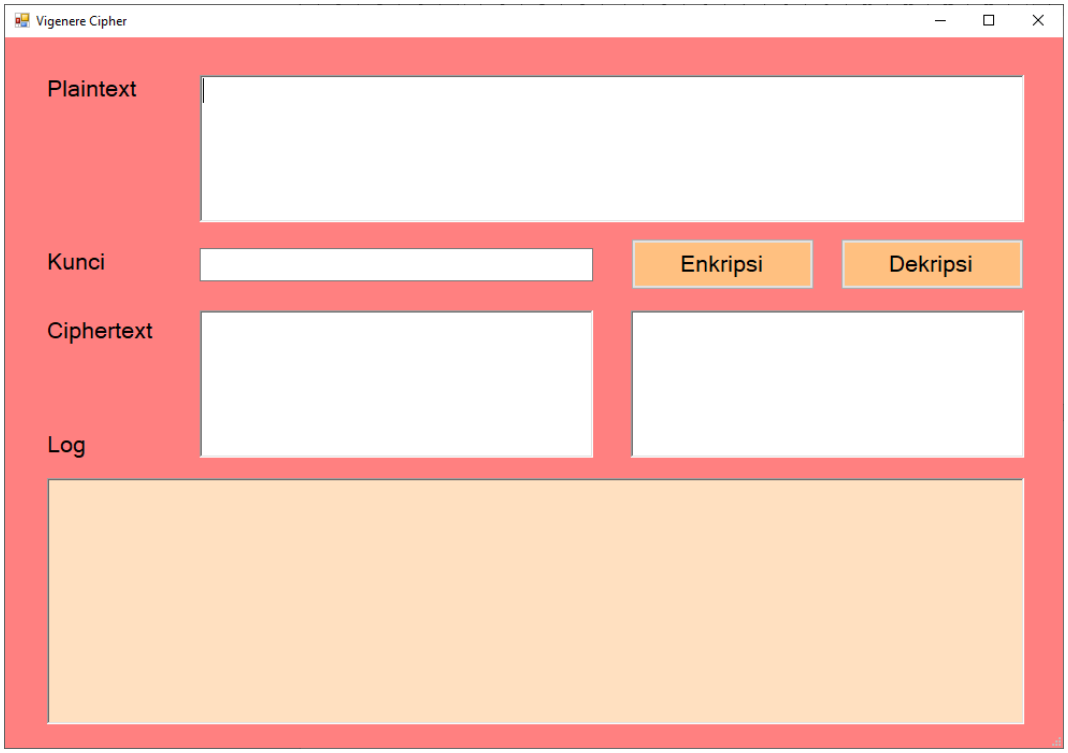
4.2.3 Halaman *About*

Halaman about adalah tampilan seputar keterangan mengenai penulis. Halaman ini menampilkan nama, NPM, fakutas, program studi dan universitas. Gamabr 4.3 adalah tampilan dari halaman About.

Gambar 4.3 Halaman *About*

4.2.4 Halaman *Vigenere Cipher*

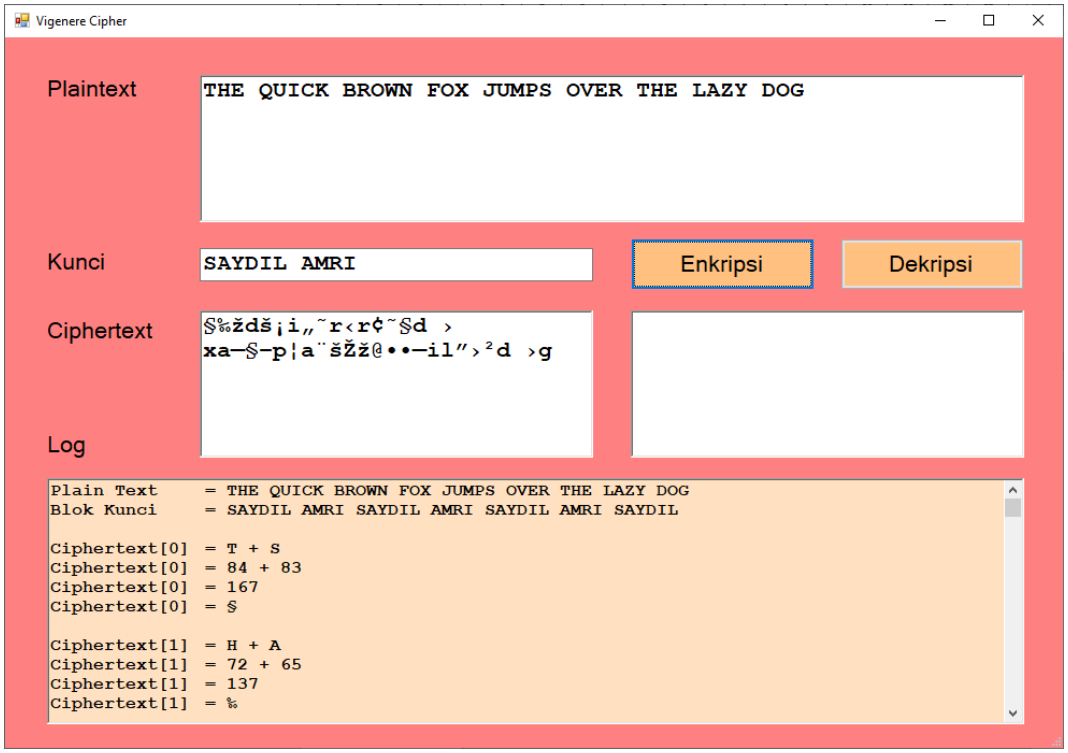
Halaman ini merupakan proses untuk melakukan enkripsi dan dekripsi. Pada halaman ini juga ditampilkan perhitungan lengkap algoritma *Vigenere Cipher* tersebut. Halaman terdiri dari dua buah *plaintext*, sebuah kunci dan sebuah ciphertext yang dibentuk dari objek textbox. Sementara untuk proses enkripsi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



Gambar 4.4 Halaman *kriptografi stream cipher* algoritma *Vigenere*

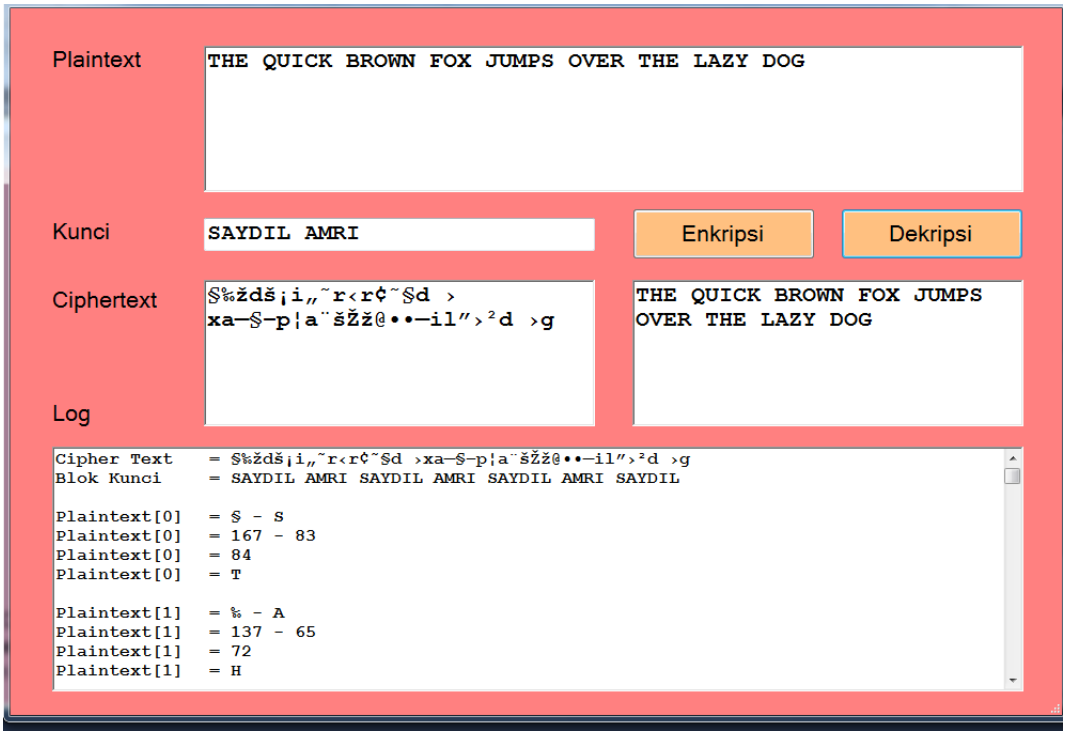
4.2.5 Hasil Perhitungan Algoritma *Vigenere*

Halaman ini berisi tentang hasil tangkap gambar dari proses yang dkerjakan 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 membentuk blok kunci. Blok kunci akan memiliki panjang yang sama dengan *plaintext*. Setiap karakter pada *plaintext* akan dilakukan pergeseran terhadap kunci untuk mendapatkan *ciphertext*.Gambar 4.5 adalah tampilan dari hasil perhitungan proses enkripsi algoritma *Vigenere Cipher*.



Gambar 4.5 Halaman enkripsi algoritma Vigenere Cipher

Proses dekripsi akan melakukan hal yang sama yaitu melakukan proses pergeseran ciphertext terhadap blok kunci sebelumnya. Hasil yang benar akan menampilkan bahwa plaintext sebelum proses enkripsi harus sama dengan plaintext yang dihasilkan setelah proses dekirpsi. Pada textbox plaintext harus menampilkan nilai yang sama seperti pada plaintext sebelumnya.Perhitungasn dinyatakan salah apabila ada satu karakter yang memiliki perbedaan nilai ASCII dengan plaintext sebelumnya. Gambar 4. 6 adalah tampilan dari hasil perhitungan proses dekripsi algoritma *Vigenere Cipher*.



Gambar 4.6 Halaman dekripsi algoritma *Vigenere Cipher*

4.3 Pengujian Perhitungan

Pengujian adalah melakukan uji coba hasil perhitungan proses enkripsi dan dekripsi algoritma *Vigenere Cipher*. Pengujian dilakukan dengan melakukan perhitungan matematika terhadap nilai ASCII pada plaintext dan kunci dan sebaliknya pada proses dekripsi akan melakukan perhitungan nilai ASCII ciphertext dan kunci. Hasil program aplikasi harus sesuai dengan hasil perhitungan yang dilakukan secara manual. Sebelum melakukan perhitungan, ada beberapa tahap yang perlu dilakukan yaitu memberikan nilai pada *plaintext* dan kunci. Berikut ini adalah penjelasan dan perhitungan lengkap proses dekripsi dan enkripsi pada algoritma *Vigenere Cipher.*

```
Plain Text  = THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Blok Kunci  = SAYDIL AMRI SAYDIL AMRI SAYDIL AMRI SAYDIL

Ciphertext[0]     = T + S
Ciphertext[0]     = 84 + 83
Ciphertext[0]     = 167
Ciphertext[0]     = §

Ciphertext[1]     = H + A
Ciphertext[1]     = 72 + 65
Ciphertext[1]     = 137
Ciphertext[1]     = ‰

Ciphertext[2]     = E + Y
Ciphertext[2]     = 69 + 89
Ciphertext[2]     = 158
Ciphertext[2]     = ž

Ciphertext[3]     =   + D
Ciphertext[3]     = 32 + 68
Ciphertext[3]     = 100
Ciphertext[3]     = d

Ciphertext[4]     = Q + I
Ciphertext[4]     = 81 + 73
Ciphertext[4]     = 154
Ciphertext[4]     = š

Ciphertext[5]     = U + L
Ciphertext[5]     = 85 + 76
Ciphertext[5]     = 161
Ciphertext[5]     = ¡

Ciphertext[6]     = I +
Ciphertext[6]     = 73 + 32
Ciphertext[6]     = 105
Ciphertext[6]     = i

Ciphertext[7]     = C + A
Ciphertext[7]     = 67 + 65
Ciphertext[7]     = 132
Ciphertext[7]     = „

Ciphertext[8]     = K + M
Ciphertext[8]     = 75 + 77
Ciphertext[8]     = 152
Ciphertext[8]     = ˜

Ciphertext[9]     =   + R
Ciphertext[9]     = 32 + 82
Ciphertext[9]     = 114
Ciphertext[9]     = r

Ciphertext[10]    = B + I
Ciphertext[10]    = 66 + 73
Ciphertext[10]    = 139
Ciphertext[10]    = ‹

Ciphertext[11]    = R +
Ciphertext[11]    = 82 + 32
Ciphertext[11]    = 114
Ciphertext[11]    = r

Ciphertext[12]    = O + S
Ciphertext[12]    = 79 + 83
Ciphertext[12]    = 162
Ciphertext[12]    = ¢

Ciphertext[13]    = W + A
Ciphertext[13]    = 87 + 65
Ciphertext[13]    = 152
Ciphertext[13]    = ˜

Ciphertext[14]    = N + Y
Ciphertext[14]    = 78 + 89
Ciphertext[14]    = 167
Ciphertext[14]    = §

Ciphertext[15]    =   + D
Ciphertext[15]    = 32 + 68
Ciphertext[15]    = 100
Ciphertext[15]    = d

Ciphertext[16]    = F + I
Ciphertext[16]    = 70 + 73
Ciphertext[16]    = 143
Ciphertext[16]    = •

Ciphertext[17]    = O + L
Ciphertext[17]    = 79 + 76
Ciphertext[17]    = 155
Ciphertext[17]    = ›

Ciphertext[18]    = X +
Ciphertext[18]    = 88 + 32
Ciphertext[18]    = 120
Ciphertext[18]    = x

Ciphertext[19]    =   + A
Ciphertext[19]    = 32 + 65
Ciphertext[19]    = 97
Ciphertext[19]    = a

Ciphertext[20]    = J + M
Ciphertext[20]    = 74 + 77
Ciphertext[20]    = 151
Ciphertext[20]    = —

Ciphertext[21]    = U + R
```

```
Ciphertext[21]    = 85 + 82
Ciphertext[21]    = 167
Ciphertext[21]    = §

Ciphertext[22]    = M + I
Ciphertext[22]    = 77 + 73
Ciphertext[22]    = 150
Ciphertext[22]    = –

Ciphertext[23]    = P +
Ciphertext[23]    = 80 + 32
Ciphertext[23]    = 112
Ciphertext[23]    = p

Ciphertext[24]    = S + S
Ciphertext[24]    = 83 + 83
Ciphertext[24]    = 166
Ciphertext[24]    = ¦

Ciphertext[25]    =   + A
Ciphertext[25]    = 32 + 65
Ciphertext[25]    = 97
Ciphertext[25]    = a

Ciphertext[26]    = O + Y
Ciphertext[26]    = 79 + 89
Ciphertext[26]    = 168
Ciphertext[26]    = ¨

Ciphertext[27]    = V + D
Ciphertext[27]    = 86 + 68
Ciphertext[27]    = 154
Ciphertext[27]    = š

Ciphertext[28]    = E + I
Ciphertext[28]    = 69 + 73
Ciphertext[28]    = 142
Ciphertext[28]    = Ž

Ciphertext[29]    = R + L
Ciphertext[29]    = 82 + 76
Ciphertext[29]    = 158
Ciphertext[29]    = ž

Ciphertext[30]    =   +
Ciphertext[30]    = 32 + 32
Ciphertext[30]    = 64
Ciphertext[30]    = @

Ciphertext[31]    = T + A
Ciphertext[31]    = 84 + 65
Ciphertext[31]    = 149
Ciphertext[31]    = •

Ciphertext[32]    = H + M
Ciphertext[32]    = 72 + 77
Ciphertext[32]    = 149
Ciphertext[32]    = •

Ciphertext[33]    = E + R
Ciphertext[33]    = 69 + 82
Ciphertext[33]    = 151
Ciphertext[33]    = —

Ciphertext[34]    =   + I
Ciphertext[34]    = 32 + 73
Ciphertext[34]    = 105
Ciphertext[34]    = i

Ciphertext[35]    = L +
Ciphertext[35]    = 76 + 32
Ciphertext[35]    = 108
Ciphertext[35]    = l

Ciphertext[36]    = A + S
Ciphertext[36]    = 65 + 83
Ciphertext[36]    = 148
Ciphertext[36]    = ”

Ciphertext[37]    = Z + A
Ciphertext[37]    = 90 + 65
Ciphertext[37]    = 155
Ciphertext[37]    = ›

Ciphertext[38]    = Y + Y
Ciphertext[38]    = 89 + 89
Ciphertext[38]    = 178
Ciphertext[38]    = ²

Ciphertext[39]    =   + D
Ciphertext[39]    = 32 + 68
Ciphertext[39]    = 100
Ciphertext[39]    = d

Ciphertext[40]    = D + I
Ciphertext[40]    = 68 + 73
Ciphertext[40]    = 141
Ciphertext[40]    = •

Ciphertext[41]    = O + L
Ciphertext[41]    = 79 + 76
Ciphertext[41]    = 155
Ciphertext[41]    = ›

Ciphertext[42]    = G +
Ciphertext[42]    = 71 + 32
Ciphertext[42]    = 103
Ciphertext[42]    = g
```

Cipher Text = §‰ždš¡i„˜r‹ r¢˜§d□›xa—§–p¦a¨šŽž@••—il”›²d□›g

```
Cipher Text = §‰ždš¡i„˜r‹r¢˜§d□›xa—§-p¦a¨šŽž@••—il”›²d□›g
Blok Kunci  = SAYDIL AMRI SAYDIL AMRI SAYDIL AMRI SAYDIL

Plaintext[0]      = § - S
Plaintext[0]      = 167 - 83
Plaintext[0]      = 84
Plaintext[0]      = T

Plaintext[1]      = ‰ - A
Plaintext[1]      = 137 - 65
Plaintext[1]      = 72
Plaintext[1]      = H

Plaintext[2]      = ž - Y
Plaintext[2]      = 158 - 89
Plaintext[2]      = 69
Plaintext[2]      = E

Plaintext[3]      = d - D
Plaintext[3]      = 100 - 68
Plaintext[3]      = 32
Plaintext[3]      =

Plaintext[4]      = š - I
Plaintext[4]      = 154 - 73
Plaintext[4]      = 81
Plaintext[4]      = Q

Plaintext[5]      = ¡ - L
Plaintext[5]      = 161 - 76
Plaintext[5]      = 85
Plaintext[5]      = U

Plaintext[6]      = i -
Plaintext[6]      = 105 - 32
Plaintext[6]      = 73
Plaintext[6]      = I

Plaintext[7]      = „ - A
Plaintext[7]      = 132 - 65
Plaintext[7]      = 67
Plaintext[7]      = C

Plaintext[8]      = ˜ - M
Plaintext[8]      = 152 - 77
Plaintext[8]      = 75
Plaintext[8]      = K

Plaintext[9]      = r - R
Plaintext[9]      = 114 - 82
Plaintext[9]      = 32
Plaintext[9]      =

Plaintext[10]     = ‹ - I
Plaintext[10]     = 139 - 73
Plaintext[10]     = 66

Plaintext[10]     = B

Plaintext[11]     = r -
Plaintext[11]     = 114 - 32
Plaintext[11]     = 82
Plaintext[11]     = R

Plaintext[12]     = ¢ - S
Plaintext[12]     = 162 - 83
Plaintext[12]     = 79
Plaintext[12]     = O

Plaintext[13]     = ˜ - A
Plaintext[13]     = 152 - 65
Plaintext[13]     = 87
Plaintext[13]     = W

Plaintext[14]     = § - Y
Plaintext[14]     = 167 - 89
Plaintext[14]     = 78
Plaintext[14]     = N

Plaintext[15]     = d - D
Plaintext[15]     = 100 - 68
Plaintext[15]     = 32
Plaintext[15]     =

Plaintext[16]     = • - I
Plaintext[16]     = 143 - 73
Plaintext[16]     = 70
Plaintext[16]     = F

Plaintext[17]     = › - L
Plaintext[17]     = 155 - 76
Plaintext[17]     = 79
Plaintext[17]     = O

Plaintext[18]     = x -
Plaintext[18]     = 120 - 32
Plaintext[18]     = 88
Plaintext[18]     = X

Plaintext[19]     = a - A
Plaintext[19]     = 97 - 65
Plaintext[19]     = 32
Plaintext[19]     =

Plaintext[20]     = — - M
Plaintext[20]     = 151 - 77
Plaintext[20]     = 74
Plaintext[20]     = J

Plaintext[21]     = § - R
```

```
Plaintext[21]     = 167 - 82
Plaintext[21]     = 85
Plaintext[21]     = U

Plaintext[22]     = - - I
Plaintext[22]     = 150 - 73
Plaintext[22]     = 77
Plaintext[22]     = M

Plaintext[23]     = p -
Plaintext[23]     = 112 - 32
Plaintext[23]     = 80
Plaintext[23]     = P

Plaintext[24]     = ¦ - S
Plaintext[24]     = 166 - 83
Plaintext[24]     = 83
Plaintext[24]     = S

Plaintext[25]     = a - A
Plaintext[25]     = 97 - 65
Plaintext[25]     = 32
Plaintext[25]     =

Plaintext[26]     = ¨ - Y
Plaintext[26]     = 168 - 89
Plaintext[26]     = 79
Plaintext[26]     = O

Plaintext[27]     = š - D
Plaintext[27]     = 154 - 68
Plaintext[27]     = 86
Plaintext[27]     = V

Plaintext[28]     = Ž - I
Plaintext[28]     = 142 - 73
Plaintext[28]     = 69
Plaintext[28]     = E

Plaintext[29]     = ž - L
Plaintext[29]     = 158 - 76
Plaintext[29]     = 82
Plaintext[29]     = R

Plaintext[30]     = @ -
Plaintext[30]     = 64 - 32
Plaintext[30]     = 32
Plaintext[30]     =

Plaintext[31]     = • - A
Plaintext[31]     = 149 - 65
Plaintext[31]     = 84
Plaintext[31]     = T

Plaintext[32]     = • - M
Plaintext[32]     = 149 - 77

Plaintext[32]     = 72
Plaintext[32]     = H

Plaintext[33]     = — - R
Plaintext[33]     = 151 - 82
Plaintext[33]     = 69
Plaintext[33]     = E

Plaintext[34]     = i - I
Plaintext[34]     = 105 - 73
Plaintext[34]     = 32
Plaintext[34]     =

Plaintext[35]     = l -
Plaintext[35]     = 108 - 32
Plaintext[35]     = 76
Plaintext[35]     = L

Plaintext[36]     = ” - S
Plaintext[36]     = 148 - 83
Plaintext[36]     = 65
Plaintext[36]     = A

Plaintext[37]     = › - A
Plaintext[37]     = 155 - 65
Plaintext[37]     = 90
Plaintext[37]     = Z

Plaintext[38]     = ² - Y
Plaintext[38]     = 178 - 89
Plaintext[38]     = 89
Plaintext[38]     = Y

Plaintext[39]     = d - D
Plaintext[39]     = 100 - 68
Plaintext[39]     = 32
Plaintext[39]     =

Plaintext[40]     = • - I
Plaintext[40]     = 141 - 73
Plaintext[40]     = 68
Plaintext[40]     = D

Plaintext[41]     = › - L
Plaintext[41]     = 155 - 76
Plaintext[41]     = 79
Plaintext[41]     = O

Plaintext[42]     = g -
Plaintext[42]     = 103 - 32
Plaintext[42]     = 71
Plaintext[42]     = G
```

Pengujian perangkat lunak didefinisikan sebagai kegiatan untuk memeriksa apakah hasil aktual sesuai dengan hasil yang diharapkan dan untuk memastikan bahwa sistem perangkat lunak bebas Cacat. Ini melibatkan pelaksanaan komponen perangkat lunak atau komponen sistem untuk mengevaluasi satu atau beberapa sifat yang menarik. Pengujian perangkat lunak juga membantu mengidentifikasi kesalahan, kesenjangan atau persyaratan yang hilang bertentangan dengan persyaratan aktual. Ini dapat dilakukan secara manual atau menggunakan alat otomatis. Pengujian perangkat lunak seharusnya berbanding lurus dengan pengujian sistem secara manual.

Pengujian sebelumnya menunjukkan hasil perhitungan enkripsi dapat dikembalikan dengan baik sehingga menjadi *plaintext* yang sempurna pada proses dekripsi. Hal ini menandakan bahwa tidak ada kesalahan yang terjadi pada perhitungan manual dan perhitungan program aplikasi. Ini berarti program aplikasi sudah berjalan dengan baik dah tidak ada penyimpangan perhitungan.

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 *Vigenere* bekerja dengan cepat dalam melakukan proses kriptografi substitusi.
2. Panjang kunci dapat ditentukan sebesar panjanga *plaintext* yang tersedia.
3. Proses dekripsi algoritma *Vigenere Cipher* bekerja dengan baik sehingga dapat mengembalikan *ciphertext* ke *plaintext*.

5.2 Saran

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

1. Hendaknya algoritma *Vigenere* dapat dikombinasikan dengan algoritma lain agar menambah tingkat keamanan pada informasi yang dikirim.
2. *Vigenere Cipher* akan lebih baik apabila menerapkan skema *Three-pass Protocol* untuk menghindari pertukaran kunci.
3. Pergeseran hendaknya dilakukan pada operasi bilangan biner agar meningkatkan kemanan pada algoritma *Vigenere Cipher*.

DAFTAR PUSTAKA

Ariyus, D. (2008). *Pengantar Ilmu Kriptografi: Teori Analisis & Implementasi*. Yogyakarta: Andi Offset.

Aryza, S., Irwanto, M., Lubis, Z., Siahaan, A. P. U., Rahim, R., & Furqan, M. (2018). A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive. In IOP Conference Series: Materials Science and Engineering (Vol. 300, No. 1, p. 012067). IOP Publishing.

Bahri, S. (2018). Metodologi Penelitian Bisnis Lengkap Dengan Teknik Pengolahan Data SPSS. Penerbit Andi (Anggota Ikapi). Percetakan Andi Ofsset. Yogyakarta.

Bahri, S. (2018). Metodologi Penelitian Bisnis Lengkap Dengan Teknik Pengolahan Data SPSS. Penerbit Andi (Anggota Ikapi). Percetakan Andi Ofsset. Yogyakarta.

Diantoro, M., Maftuha, D., Suprayogi, T., Iqbal, M. R., Mufti, N., Taufiq, A., ... & Hidayat, R. (2019). Performance of Pterocarpus Indicus Willd Leaf Extract as Natural Dye TiO2-Dye/ITO DSSC. Materials Today: Proceedings, 17, 1268-1276.

Edraw. (2019). What is Algorithm - Definition, Types and Application. Retrieved October 27, 2019, from https://www.edrawsoft.com/algorithm-definition.php

EDUCBA. (2017). What is Cryptography? | Types and Advantages of Cryptography. Retrieved October 23, 2019, from https://www.educba.com/what-is-cryptography/

Firmansyah, E. R. (2012). Algoritma Kriptografi & Contohnya. *Universitas Islam Negeri Syarif Hidayatullah Jakarta*.

Fitriani, W., Rahim, R., Oktaviana, B., & Siahaan, A. P. U. (2017). Vernam Encypted Text in End of File Hiding Steganography Technique. Int. J. Recent Trends Eng. Res, 3(7), 214-219.

Hariyanto, E., Iqbal, M., Siahaan, A. P. U., Saragih, K. S., & Batubara, S. (2019, March). Comparative Study of Tiger Identification Using Template Matching Approach based on Edge Patterns. In Journal of Physics: Conference Series (Vol. 1196, No. 1, p. 012025). IOP Publishing.

Hidayat, A. (2012). Algoritma Kriptografi Vigenere Cipher. Retrieved November 4, 2019, from https://arfianhidayat.com/algoritma-kriptografi-vigenere-cipher

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

Ladjamudin, A.-B. bin. (2005). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.

Lee, C. (2014). *Buku Pintar Pemrograman Visual Basic 2010*. Jakarta: Elex Media Komputindo.

Lubis, A., & Batubara, S. (2019, December). Sistem Informasi Suluk Berbasis Cloud Computing Untuk Meningkatkan Efisiensi Kinerja Dewan Mursyidin Tarekat Naqsyabandiyah Al Kholidiyah Jalaliyah. In Prosiding SiManTap: Seminar Nasional Matematika dan Terapan (Vol. 1, pp. 717-723).

Munir, R. (2006). *Kriptografi*. Bandung: Informatika.

Nakatsu, R. T. (2009). *Reasoning with Diagrams : Decision-Making and Problem- Solving with Diagrams.* John Wiley & Sons.

Nasution, M. D. T. P., Rossanty, Y., Siahaan, A. P. U., & Aryza, S. (2018). The Phenomenon of Cyber-Crime and Fraud Victimization in Online Shop. *International Journal of Civil Engineering and Technology*, *9*(6), 1583–1592.

Pratama, G. M., & Tamatjita, E. N. (2015). MODIFIKASI ALGORITMA VIGENERE CIPHER MENGGUNAKAN METODE CATALAN NUMBER DAN DOUBLE COLUMNAR TRANSPOSITION. *Compiler*, *4*(1), 31–40.

Rahim, R. (2018, October). A Novelty Once Methode Power System Policies Based On SCS (Solar Cell System). In International Conference of ASEAN Prespective and Policy (ICAP) (Vol. 1, No. 1, pp. 195-198).

Rahmel, D. (2008). *Visual Basic.NET*. New York: McGraw-Hill.

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.

Stallings, W. (2013). *Cryptography and Network Security: Principles and Practice*. New Jersey: Prentice Hall Press.

Suherman, S., & Khairul, K. (2018). Seleksi Pegawai Kontrak Menjadi Pegawai Tetap Dengan Metode Profile Matching. IT Journal Research and Development, 2(2), 68-77.

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

Sulistianingsih, I., Suherman, S., & Pane, E. (2019). Aplikasi Peringatan Dini Cuaca Menggunakan Running Text Berbasis Android. IT Journal Research and Development, 3(2), 76-83.

Sumandri. (2017). Studi Model Algoritma Kriptografi Klasik dan Modern. *SEMINAR MATEMATIKA DAN PENDIDIKAN MATEMATIKA*.

Tarigan, A. D., & Pulungan, R. (2018). Pengaruh Pemakaian Beban Tidak Seimbang Terhadap Umur Peralatan Listrik. RELE (Rekayasa Elektrikal dan Energi): Jurnal Teknik Elektro, 1(1), 10-15.

Tarigan, A. D. (2018, October). A Novelty Method Subjectif of Electrical Power Cable Retirement Policy. In International Conference of ASEAN Prespective and Policy (ICAP) (Vol. 1, No. 1, pp. 183-186).

Wagner, N. R. (2003). *The Laws of Cryptography with Java Code*.

Wahyuni, S., Lubis, A., Batubara, S., & Siregar, I. K. (2018, September). IMPLEMENTASI ALGORITMA CRC 32 DALAM MENGIDENTIFIKASI KEASLIAN FILE. In Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 1-6).

Wasserkrug, S., Dalvi, N., Munson, E. V., Gogolla, M., Sirangelo, C., Fischer-Hübner, S., … Snodgrass, R. T. (2009). Unified Modeling Language. In *Encyclopedia of Database Systems* (pp. 3232–3239). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-39940-9_440

Weerasinghe, T. D. B. (2013). An effective RC4 stream cipher. In *2013 IEEE 8th International Conference on Industrial and Information Systems* (pp. 69–74). IEEE. https://doi.org/10.1109/ICIInfS.2013.6731957

Wibowo, P., Lubis, S. A., & Hamdani, Z. T. (2017). Smart Home Security System Design Sensor Based on Pir and Microcontroller. International Journal of Global Sustainability, 1(1), 67-73.

Yakub. (2012). *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.