IMPLEMENTASI KRIPTOGRAFI KUNCI PUBLIK DAN PRIVAT DALAM MENGAMANKAN PESAN DENGAN METODE AFFINE 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	:	MAUNO KOIVISTO PURBA
NPM	:	1414370533
PROGRAM STUDI	:	SISTEM KOMPUTER

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

ABSTRAK

MAUNO KOIVISTO PURBA
Implementasi Kriptografi Kunci Publik Dan Privat Dalam Mengamankan Pesan Dengan Metode Affine Cipher
2020

Perkembangan zaman membuat pertambahan data semaking meningkat. Dengan peningkatan data yang begitu pesat, tingkat keamanan pada data tersebut menjadi lebih rentan. Ada banyak pencurian data yang dilakukan oleh pihak yang tidak bertanggung jawab. Dibutuhkan teknik pengamanan data yang dapat menghindari penyalahgunaan pada pengiriman pesan. Kriptografi adalah salah satu teknik yang dapat digunakan dalam melakukan pengamanan data. Algoritma Affine Cipher dapat digunakan untuk melakukan proses enkripsi dan dekripsi pada pesan. Affine cipher memiliki kunci yang berbeda pada saat proses enkripsi dan dekripsi. Kunci tersebut akan dilakukan penambahan sejumah karakter. Tingkat keamanan algoritma Affine cipher adalah sangat baik. Dengan penerapan teknik ini pada pengiriman pesan, keamanan dari pesan tersebut dapat ditingkatkan.

Kata Kunci: dekripsi, enkripsi, kriptografi, Affine, Cipher

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pesan merupakan sesuatu yang disampaikan kepada orang lain dengan suatu tujuan tertentu. Dalam pengiriman pesan, ada beberapa informasi yang mungkin seharusnya tidak perlu atau tidak harus diketahui oleh orang lain. Seperti informasi data diri, akun bank, biografi atau data-data lainnya tidak perlu orang lain mengetahuinya sementara pesan tersebut harus tetap dikirimkan ke penerima. Pengiriman pesan adalah melalui media jaringan. Jika pesan tersebut tidak harus terkirim atau dengan kata lain hanya berada pada komputer yang tidak memerlukan akses internet, maka pesan atau informasi tersebut tidak harus dilengkapi dengan proteksi keamanan.

Untuk mengamankan data, diperlukan suatu cara yang dapat memberikan solusi untuk keamanan data. Proses untuk mengamankan data ini adalah kriptografi. Kriptografi berperan penting dalam mengubah data plaintext menjadi ciphertext. Ada dua proses yang terlibat dalam melakukan pengamanan data, yaitu enkripsi dan dekripsi. Kriptografi adalah seni dan ilmu membuat sistem keamanan yang mampu memberikan keamanan informasi. Kriptografi berkaitan dengan pengamanan data digital. Hal ini mengacu pada desain mekanisme berdasarkan algoritma matematika yang menyediakan layanan keamanan informasi mendasar. Suatu teknik kriptografi berfungsi sebagai toolkit besar yang berisi berbagai teknik dalam aplikasi keamanan. Tujuan utama menggunakan kriptografi adalah untuk menyediakan

layanan keamanan informasi yang dapat menjaga data dari ancaman pihak yang tidak bertanggung jawab.

Enkripsi modern adalah teknik pengelolaan kunci untuk keamanan komputer dan komunikasi yang canggih. Teknik kriptografi ini sepenuhnya didasarkan pada ide-ide perhitungan matematika seperti teori bilangan, kunci publik dan kunci privat dan teori kompleksitas komputasi serta konsep probabilitas. Dalam dunia kriptografi, ada dua jenis model yang digunakan, yaitu simetris dan asimetris. Penelitian ini akan membahas kunci asimetris dimana kunci yang digunakan pada proses enkripsi berbeda dengan kunci yang digunakan pada proses dekripsi.

Kriptografi kunci publik, atau kriptografi asimetris, adalah skema enkripsi yang menggunakan dua kunci yang saling terikat dan sudah diperhitungkan oleh perhitungan matematika. Kunci ini dikenal dengan kunci publik dan kunci privat. Tidak seperti algoritma kunci simetris yang mengandalkan satu kunci untuk mengenkripsi dan mendekripsi, setiap kunci melakukan fungsi yang berbeda. Kunci publik digunakan untuk mengenkripsi dan kunci pribadi digunakan untuk mendekripsi. Kunci publik dapat dibagikan secara bebas, memungkinkan pengguna metode yang mudah dan nyaman untuk mengenkripsi konten dan memverifikasi tanda tangan digital, dan kunci pribadi dapat dirahasiakan, memastikan hanya pemilik kunci pribadi yang dapat mendekripsi konten dan membuat tanda tangan digital.

Salah satu metode kunci publik dan kunci privat adalah Affine Cipher. Algoritma ini termasuk algoritma yang menggunakan bilangan prima dalam melakukan proses enkripsi dan dekripsi. Bilangan prima adalah bilangan yang paling sering digunakan untuk kriptografi kunci publik dan kunci privat. Algoritma ini dikenal sangat baik untuk melakukan enkripsi dan dekripsi. Proses ini dapat dibuktikan dengan menciptakan suatu program aplikasi yang akan melakukan perhitungan otomatis untuk proses enkripsi dan dekripsi. Berdasarkan latar belakang di atas maka penulis tertarik untuk memilih judul "**IMPLEMENTASI KRIPTOGRAFI KUNCI PUBLIK DAN PRIVAT DALAM MENGAMANKAN PESAN DENGAN METODE AFFINE 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 Affine Cipher?
2. Bagaimana menentukan bilangan yang digunakan pada proses enkripsi dan dekripsi?
3. Bagaimana menentukan kunci publik dan kunci privat pada algoritma Affine Cipher?
4. Bagaimana menentukan pergeseran kunci untuk menambah tingkat keamanan pada algoritma Affine Cipher?

1.3 Batasan Masalah

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

1. Algoritma Affine mengunakan karakter berdasarkan tabel ASCII.
2. Modulo yang digunakan adalah 256.
3. Pesan yang digunakan sebagai plaintext adalah pesan berbasis teks.

1.4 Tujuan Penelitian

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

1. Mengetahui proses enkripsi dan dekripsi dengan algoritma Affine Cipher.
2. Untuk menentukan bilangan yang digunakan pada proses enkripsi dan dekripsi.
3. Untuk menentukan kunci publik dan kunci privat pada algoritma Affine Cipher.
4. Untuk menentukan pergeseran kunci untuk menambah tingkat keamanan pada algoritma Affine Cipher.

1.5 Manfaat Penelitian

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

1. Dapat mengamankan pesan yang akan dikirim ke orang lain.
2. Informasi yang dikirimkan tidak mudah dapat dipecahkan sehingga terjadi penyalahgunaan informasi.

BAB II

LANDASAN TEORI

2.1 Keamanan Data

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

2.1.1 Elemen Keamanan Data

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

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

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

2.1.2 Pertimbangan Keamanan Data

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

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

2.1.3 Penilaian Risiko Data

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

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

2.1.4 Minimalisasi Data

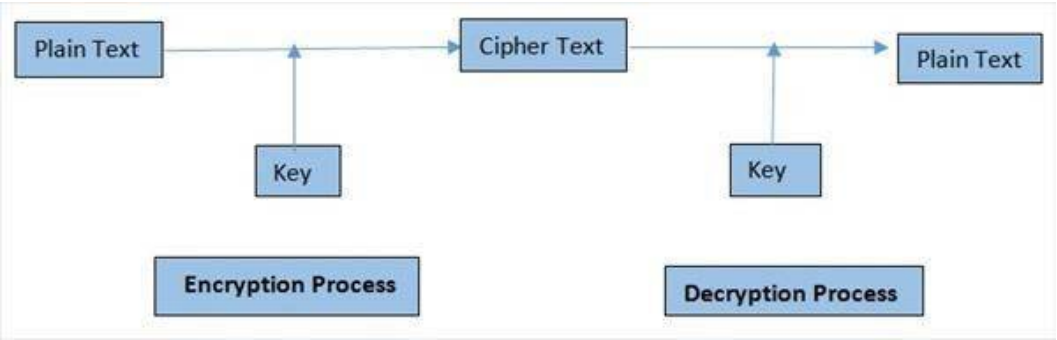
Dekade terakhir manajemen TI telah melihat pergeseran dalam persepsi data. Sebelumnya, memiliki lebih banyak data hampir selalu lebih baik daripada kurang. Sesorang tidak pernah bisa yakin sebelumnya tentang apa yang ingin dilakukan dengannya. Hari ini, data adalah kewajiban. Ancaman dari pelanggaran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

2.2 Kriptografi

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

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

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



Gambar 2.1 Skema proses enkripsi dan dekripsi
Sumber: (W. Stallings, 2013)

2.2.1 Kriptografi Simetris

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

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

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

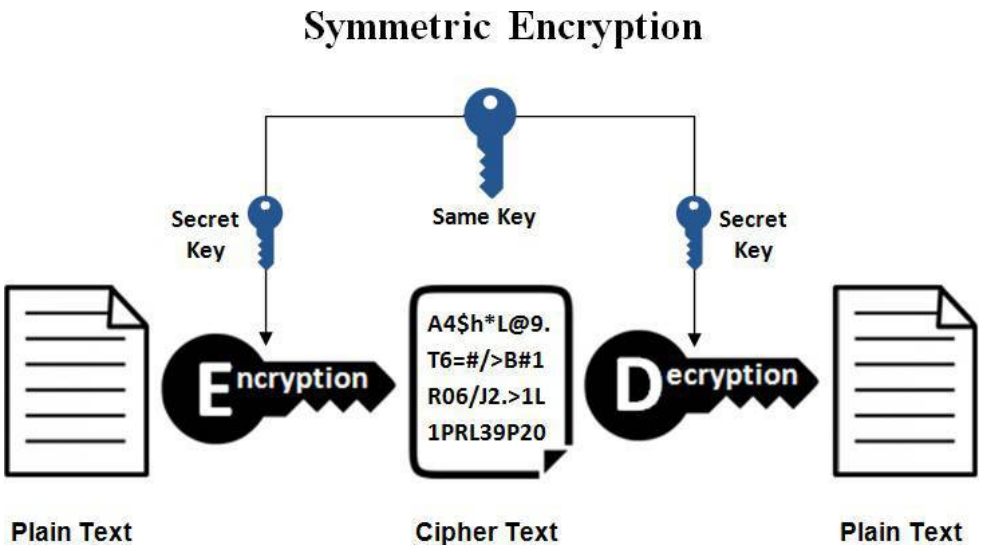
Beberapa contoh algoritma enkripsi simetris meliputi:

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

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

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

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



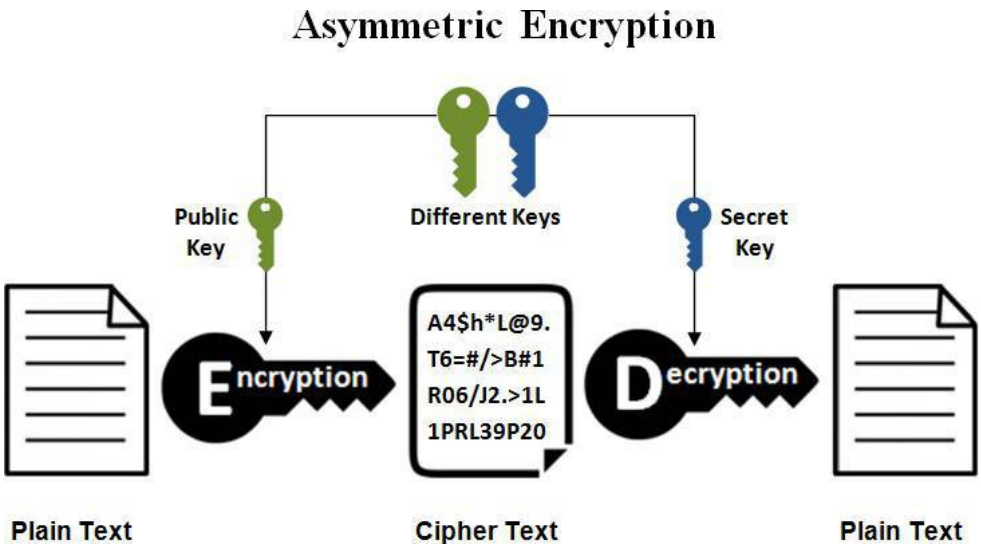
Gambar 2.2 Skema kriptografi simeris
Sumber: (Information, 2019)

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

2.2.2 Kriptografi Asimetris

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

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



Gambar 2.3 Skema kriptografi asimeris
Sumber: (Information, 2019)

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

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

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

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

2.3 Affine Cipher

Affine cipher adalah jenis cipher substitusi monoalphabetic, di mana setiap huruf dalam alfabet dipetakan dengan angka yang setara, dienkripsi menggunakan fungsi matematika sederhana, dan dikonversi kembali menjadi huruf. Rumus yang digunakan berarti bahwa setiap huruf mengenkripsi ke satu huruf lain, dan kembali lagi, yang berarti cipher pada dasarnya adalah cipher substitusi standar dengan aturan yang mengatur surat yang pergi ke mana.

Seluruh proses bergantung pada modulo yang berfungsi (panjang alfabet yang digunakan). Dalam affine cipher, huruf-huruf alfabet ukuran m pertama-tama dipetakan ke bilangan bulat dalam kisaran 0 ... m-1.

Kunci' untuk sandi Affine terdiri dari 2 angka, kami akan menyebutnya a dan b. Diskusi berikut mengasumsikan penggunaan alfabet 26 karakter (m = 26). a harus dipilih untuk menjadi relatif prima dari m (mis. a seharusnya tidak memiliki faktor yang sama dengan m). Perhitungan Affine cipher dapat diliihat pada gambar berikut ini.

Encryption: Key Values a=17, b=20

Original Text	T	W	E	N	T	Y		F	I	F	T	E	E	N
x	19	22	4	13	19	24		5	8	5	19	4	4	13
ax+b % 26*	5	4	10	7	5	12		1	0	1	5	10	10	7
Encrypted Text	F	E	K	H	F	M		B	A	B	F	K	K	H

Decryption: a^-1 = 23

Encrypted Text	F	E	K	H	F	M		B	A	B	F	K	K	H
Encrypted Value	5	4	10	7	5	12		1	0	1	5	10	10	7
23 *(x-b) mod 26	19	22	4	13	19	24		5	8	5	19	4	4	13
Decrypted Text	T	W	E	N	T	Y		F	I	F	T	E	E	N

Gambar 2.4 Perhitungan Affine Cipher

2.4 Sistem Informasi

Sistem informasi, seperangkat komponen terintegrasi untuk mengumpulkan, menyimpan, dan memproses data dan untuk menyediakan informasi, pengetahuan, dan produk digital. Perusahaan bisnis dan organisasi lain bergantung pada sistem informasi untuk melaksanakan dan mengelola operasi mereka, berinteraksi dengan pelanggan dan pemasok mereka, dan bersaing di pasar. Sistem informasi digunakan untuk menjalankan rantai pasokan antar organisasi dan pasar elektronik. Misalnya, perusahaan menggunakan sistem informasi untuk memproses akun keuangan, untuk mengelola sumber daya manusia mereka, dan untuk menjangkau pelanggan potensial mereka dengan promosi online. Banyak perusahaan besar dibangun sepenuhnya di sekitar sistem informasi. Ini termasuk eBay, pasar lelang besar; Amazon, mal elektronik yang berkembang dan penyedia layanan cloud computing; Alibaba, e-marketplace bisnis-ke-bisnis; dan Google, perusahaan mesin pencari yang memperoleh sebagian besar pendapatannya dari iklan kata kunci di pencarian Internet. Pemerintah menggunakan sistem informasi

untuk menyediakan layanan yang hemat biaya bagi warga negara. Barang digital — seperti buku elektronik, produk video, dan perangkat lunak — dan layanan online, seperti game dan jejaring sosial, dikirimkan dengan sistem informasi. Individu mengandalkan sistem informasi, umumnya berbasis Internet, untuk melakukan banyak kehidupan pribadi mereka: untuk bersosialisasi, belajar, berbelanja, perbankan, dan hiburan (Zwass, 2019).

2.5 Algoritma

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

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

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

2.6 Unified Modelling Language (UML)

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

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

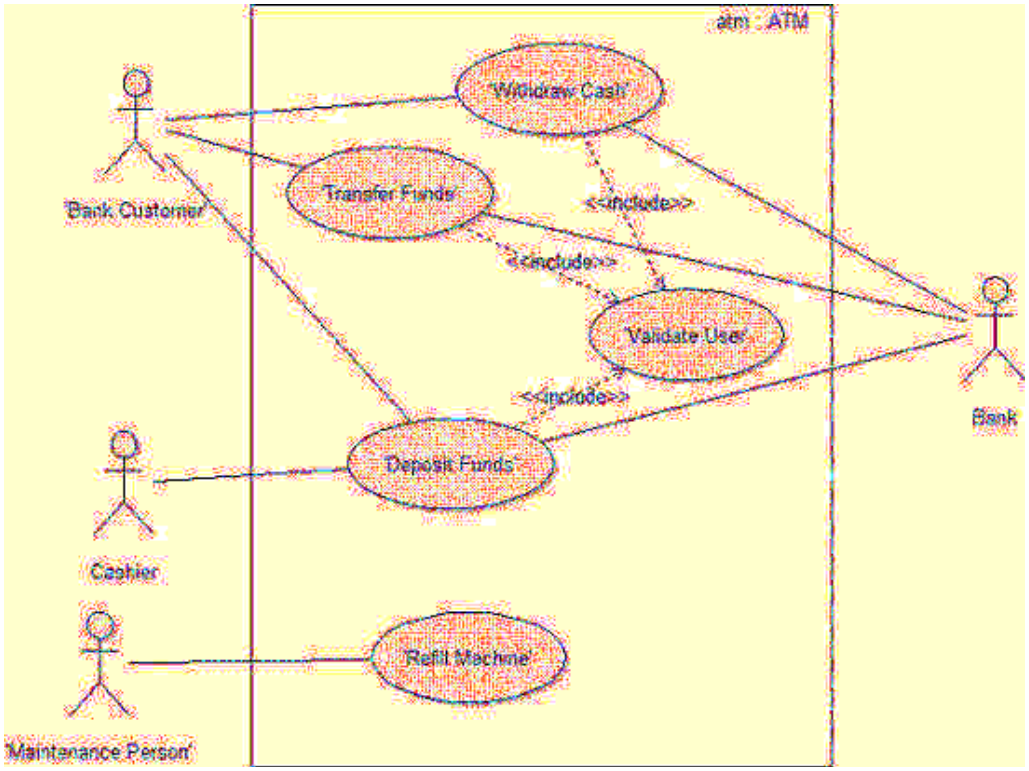
2.6.1 Use Case Diagram

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

Model use-case dapat berisi paket yang digunakan untuk menyusun model untuk menyederhanakan analisis, komunikasi, navigasi, pengembangan,

pemeliharaan, dan perencanaan. Faktanya, sebagian besar model use-case adalah tekstual, dengan teks yang ditangkap dalam Spesifikasi *Use-Case* yang terkait dengan setiap elemen model use-case. Spesifikasi ini menjelaskan alur peristiwa *use case*. Model *use-case* berfungsi sebagai utas pemersatu sepanjang pengembangan sistem. Ini digunakan sebagai spesifikasi utama dari persyaratan fungsional untuk sistem, sebagai dasar untuk analisis dan desain, sebagai input untuk perencanaan iterasi, sebagai dasar mendefinisikan kasus uji dan sebagai dasar untuk dokumentasi pengguna. (Kurniawan, 2018).

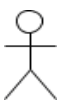
Use case diagram merupakan suatu diagram yang berisi *use case*, *actor*, serta *relationship* diantaranya. *Use Case Diagram* dapat digunakan untuk kebutuhan apa saja yang diperlukan dalam suatu sistem, sehingga sistem dapat digambarkan dengan jelas bagaimana proses dari sistem tersebut, bagaimana cara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

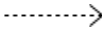
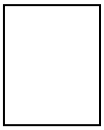


Gambar 2.5. Use-case Diagram ATM
Sumber: (Uml-diagrams.org, 2019)

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

Tabel 2.1 Simbol Use Case Diagram

No	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.

2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		*Generalization*	Hubungan dimana objek anak berbagi perilaku dan struktur data dari objek yang ada diatasnya .
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang

			menghasilkan suatu hasil yang terukur bagi suatu actor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.6.2 Activity Diagram

Activity Diagram (Diagram Aktifitas) menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir (Ladjamudin, 2017).

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

Tabel 2.2 Simbol Activity Diagram

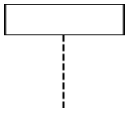
No	Gambar	Nama	Keterangan
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk /diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk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

Sumber: (Kurniawan, 2018)

2.6.3 Sequence Diagram

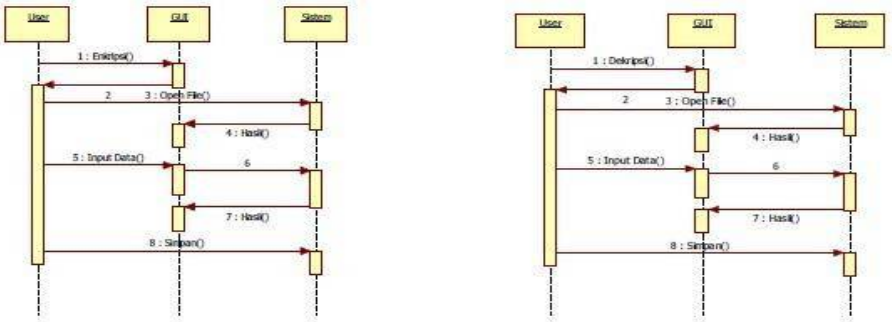
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode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

Tabel 2.3 Simbol Sequence Diagram

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

Sumber: (Kurniawan, 2018)

Contoh *Sequence Diagram*:



Gambar 2.6 Contoh Sequence Diagram
Sumber: (Kurniawan, 2018)

2.7 Flowchart

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

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

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

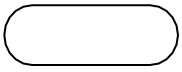
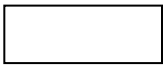
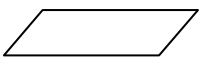
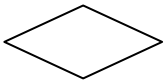
Diagram alir menggambarkan aspek-aspek tertentu dari proses dan biasanya dilengkapi dengan jenis diagram lainnya. Misalnya, Kaoru Ishikawa, mendefinisikan diagram alir sebagai salah satu dari tujuh alat dasar kendali mutu, di sebelah histogram, diagram Pareto, lembar periksa, diagram kontrol, diagram

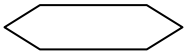
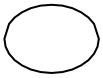
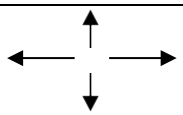
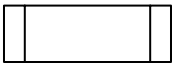
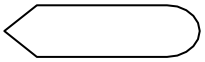
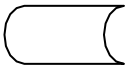
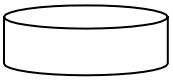
sebab-akibat, dan diagram sebaran. Demikian pula, di UML, notasi pemodelan konsep standar yang digunakan dalam pengembangan perangkat lunak, diagram aktivitas, yang merupakan jenis diagram alur, hanyalah salah satu dari banyak jenis diagram yang berbeda.

Diagram Nassi-Shneiderman dan Drakon-chart adalah notasi alternatif untuk aliran proses. Nama alternatif umum termasuk diagram alir, diagram alur proses, diagram alur fungsional, peta proses, diagram proses, diagram proses fungsional, model proses bisnis, model proses, diagram alir proses, diagram alur kerja, diagram alir bisnis. Istilah "diagram alur" dan "diagram alir" digunakan secara bergantian (Nakatsu, 2009).

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 2.4 Simbol Flowchart

NO	SIMBOL	FUNGSI
1.		**Terminal**, untuk memulai atau mengakhiri suatu program
2.		**Proses**, suatu simbol yang menunjukkan setiap pengolahan yang dilakukan.
3.		**Input-Output**, untuk memasukkan menunjukkan hasil dari suatu proses
4.		**Decision**, suatu kondisi yang akan menghasilkan 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 keluarannya suatu prosedur pada lembaran kertas lainnya
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 ditunjukkan ke suatu device, seperti printer, dan sebagainya
11		Penyimpanan file secara sementara
12		Menunjukkan input / Output Hardisk (media penyimpanan)

Sumber: (Kurniawan, 2018)

2.8 Bahasa Pemrograman

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

Bahasa tingkat tinggi dirancang agar mudah dibaca dan dipahami. Ini memungkinkan programmer untuk menulis kode sumber dengan cara alami, menggunakan kata-kata dan simbol yang logis. Misalnya, kata-kata yang dicadangkan seperti fungsi, sementara, jika, dan lainnya digunakan dalam sebagian besar bahasa pemrograman utama. Simbol seperti <,>, ==, dan! = Adalah operator umum. Banyak bahasa tingkat tinggi cukup mirip sehingga pemrogram dapat dengan mudah memahami kode sumber yang ditulis dalam berbagai bahasa.

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

Bahasa tingkat rendah meliputi bahasa assembly dan bahasa mesin. Bahasa assembly berisi daftar instruksi dasar dan jauh lebih sulit dibaca daripada bahasa

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

2.8.1 Kode Program

Kode ini hampir seperti menulis paragraf instruksi atau membuat daftar tugas untuk komputer. Tidak seperti kita manusia, daftar tugas dan instruksi yang Anda tulis untuk komputer harus sangat rinci dan ditulis dalam beberapa logika. Dengan kode dan pemrograman, Anda dapat membuat komputer menggambar bentuk yang rumit dan membuat grafik komputer yang kaya, dan kemudian membuat program yang memahami mekanika game dan membantu Anda membuat game yang terasa nyata dengan gravitasi dan tabrakan partikel, dengan program ini yang paling bisa Anda buat semua permainan intens dan imersif. Dengan kode dan pemrograman, Anda dapat membuat dan mengirim konten di seluruh dunia dengan blog Anda dan situs web pribadi dan gaya blog Anda untuk memenuhi gaya Anda. Anda dapat membangun solusi bisnis yang digerakkan oleh teknologi dan menjangkau pelanggan yang lebih luas serta memenuhi kebutuhan yang lebih luas. Selanjutnya, dengan kode dan pemrograman, Anda dapat membuat aplikasi rumah pintar, seperti pengumpan hewan peliharaan otomatis, cermin pintar atau bahkan

membuat robot yang dapat membantu menyelesaikan tugas-tugas rumah tangga dan menjadi asisten virtual Anda untuk berbicara dan memahami Anda. Berbeda dengan apa yang dipikirkan banyak orang, ada banyak seni yang terlibat dalam teknik komputer dan ilmu komputer. Anda mungkin tertarik dengan Apa itu pemrograman? posting blog untuk tahu lebih banyak (Gabbrielli & Martini, 2010).

2.8.2 Intepreter

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

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

Dalam kebanyakan kasus, kompiler lebih disukai karena outputnya berjalan jauh lebih cepat dibandingkan dengan interpretasi baris demi baris. Namun, karena

interpretasi terjadi per baris atau pernyataan, itu dapat dihentikan di tengah eksekusi untuk memungkinkan modifikasi kode atau debugging. Keduanya memiliki kelebihan dan kekurangan masing-masing dan tidak saling eksklusif; ini berarti bahwa mereka dapat digunakan bersamaan karena sebagian besar lingkungan pengembangan terintegrasi menggunakan kompilasi dan terjemahan untuk beberapa bahasa tingkat tinggi.

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

2.8.3 Compiler

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

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

2.9 Visual Basic.Net

Visual Basic.Net merupakan salah satu *tool development Microsoft* yang dapat digunakan untuk membuat aplikasi di lingkungan kerja berbasis sistem operasi *Windows*. Visual Basic.Net menyediakan *tools* bagi para *developer* untuk membangun aplikasi yang berjalan di *.Net Framework* (Safik, 2016 : 2)

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

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

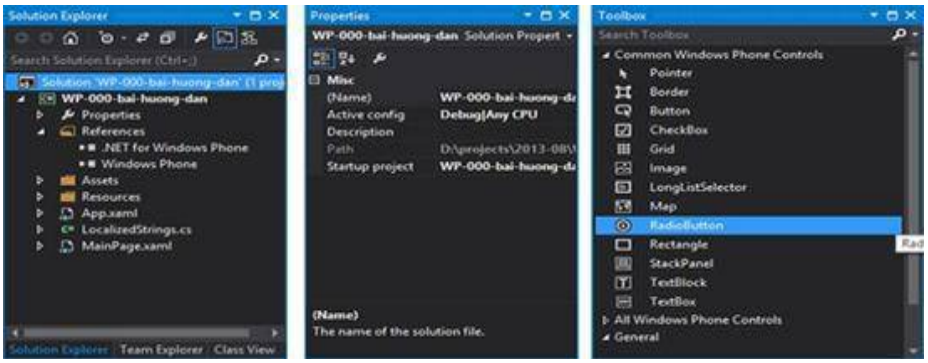
Edition (*VBScript*), mirip seperti halnya *Visual Basic*, tetapi cara kerjanya yang berbeda.

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

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

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

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



Gambar 2.7 Tampilan Toolbox Visual Basic
Sumber: (Lee, 2014)

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

Tabel 2.5 Toolbox Visual Basic

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

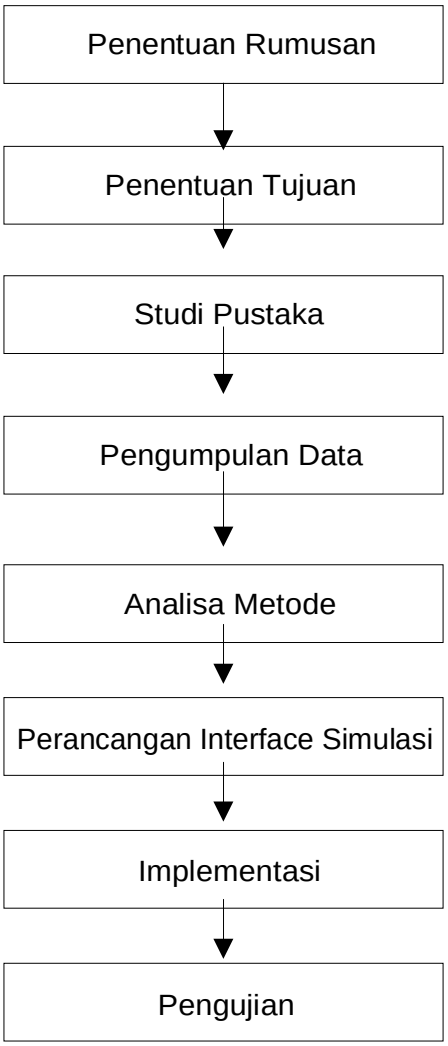
Sumber: (Lee, 2014)

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian merupakan langkah-langkah yang akan diambil untuk menyelesaikan penelitian. Gambar 3.1 adalah tahapan penelitian yang dilakukan.



Gambar 3.1 Tahapan Penelitian

Berikut adalah tahapan penelitian yang dilakukan:

1. Penentuan Rumusan Masalah

   Bagian ini akan menentukan apa topik permasalahan yang sedang terjadi. Perumusan berhubungan apa yang akan diteliti pada penelitian ini.

2. Penentuan Tujuan dan Manfaat

   Tujuannya dan manfaat agar meningkatkan keamanan pengiriman data dan dapat menghindari pencurian data.

3. Studi Pustaka

   Studi literatur dilakukan untuk mendapatkan kajian-kaijan yang berhubungan dengan ilmu kriptografi khususnya algoritma Affine Cipher.

4. Pengumpulan Data

   Data yang digunakan berupa plaintext yang dapat diambil dari berbagai data yang memiliki kode ASCII. Kode ASCII berperan untuk melakukan perhitungan matematika pada proses enkripsi dan dekripsi.

5. Analisa

   Bagian ini menjelaskan proses analisa permasalahan dan bagaimana permasalahan dapat diselesaikan dengan terstruktur. Analisa terbagi dua yaitu analisa sistem yang sedang berjalan dengan analisa yang akan dilakukan dengan sistem yang baru.

6. Perancangan Algoritma

   Bagian ini membahas bagaimana cara kerja dan tahapan yang dilakukan oleh algoritma Affine Cipher. Perancangan pembuatan kunci juga dilakukan pada perancangan algoritma.

7. Perancangan Interface Simulasi

   Bagian ini akan menentukan bagaimana cara membuat antarmuka yang akan berhubungan dengan pengguna program aplikasi.

8. Implementasi

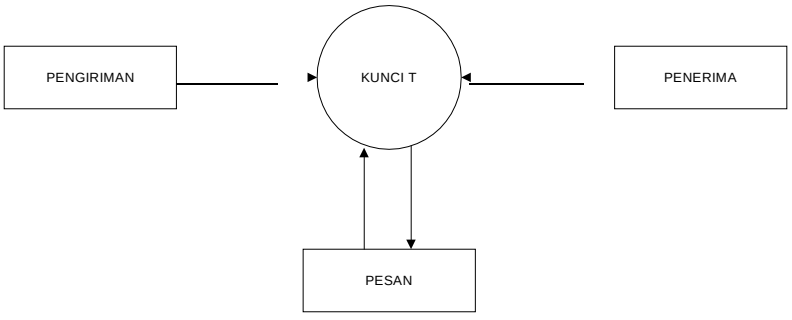
   Bagian ini membahas bagaimana cara menerapkan perancangan yang sudah dilakukan menjadi program aplikasi yang utuh. Program akan dibuat menggunakan bahasa pemrograman Microsoft Visual Basic.Net 2010..

9. Pengujian Sistem

   Bagian ini melakukan perhitungan terhadap algoritma Affine Cipher untuk melihat kebenaran program aplikasi yang telah dibuat menggunakan bahasa pemrograman.

3.2 Skema Pengiriman Pesan

Plaintext merupakan pesan yang akan dikirim kepada penerima dalam bentuk tersandi atau rahasia. Pesan tersebut seharusnya dilakukan proses enkripsi terlebih dahulu sebelum dikirimkan agar terhindar dari serangan. Gambar 3.2 adalah skema pengiriman pesan.



Gambar 3.2 Skema Pengiriman Pesan

3.3 Target Pencapaian Hasil Penelitian

Penelitian ini memiliki pencapaian agar memiliki hasil yang baik. Pengharapan yang penulis lakukan, program aplikasi dapat berjalan dengan baik dan dapat menjadi suatu solusi dalam pengiriman pesan rahasia. Target dilakukan berdasarkan kekurangan yang teradi sebelum penelitian yang berhubungan dengan pengiriman data dilakukan. Tabel 3.1 adalah target hasil yang akan diperoleh.

Tabel 3.1 Target yang akan dicapai

No.	Sistem Berjalan	Target	Hasil
1	Pengiriman pesan menggunakan kunci yang sama pada proses enkripsi dan dekripsi.	Pengirim dan penerima memiliki kunci masing-masing untuk membuka pesan	Kunci yang digunakan lebih efektif dan tidak bocor.
2	Kunci tidak memiliki pergeseran/shift.	Penambahan pergeseran/shift pada Affine Cipher	Hasil enkripsi lebih baik dengan tambahan shift.
3	Pengirim tidak menggunakan proses enkripsi dan dekripsi pada saat melakukan pengiriman data	Plaintext akan dienkripsi sehingga menghasilkan ciphertext.	Ciphertext diharapkan dapat menghindari pencurian dan penyalahgunaan informasi.

3.4 Perancangan Penelitian

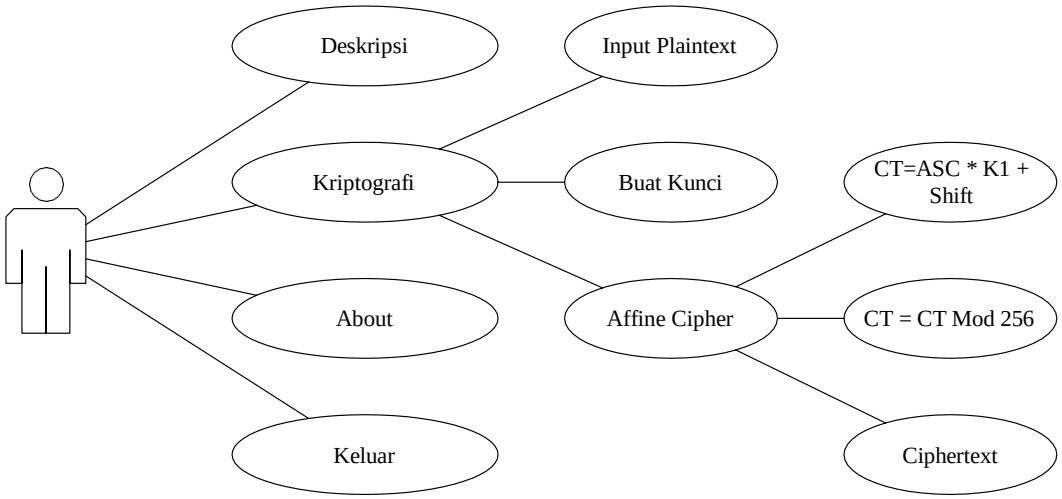
Sebelum program aplikasi dibuat, penulis melakukan suatu perancangan agar program yang dihasilkan memiliki kualitas yang baik. Perancangan dibutuhkan untuk memberikan susunan dan tata letak yang sempurna pada saat program aplikasi diciptakan. Tujuannya agar program aplikasi tersebut layak untuk

berjalan dan memproses algoritma Affine Cipher. Perancangan ini dimodelkan dengan cara menampilkan dalam bentuk *Unified Modelling Language (UML)*.

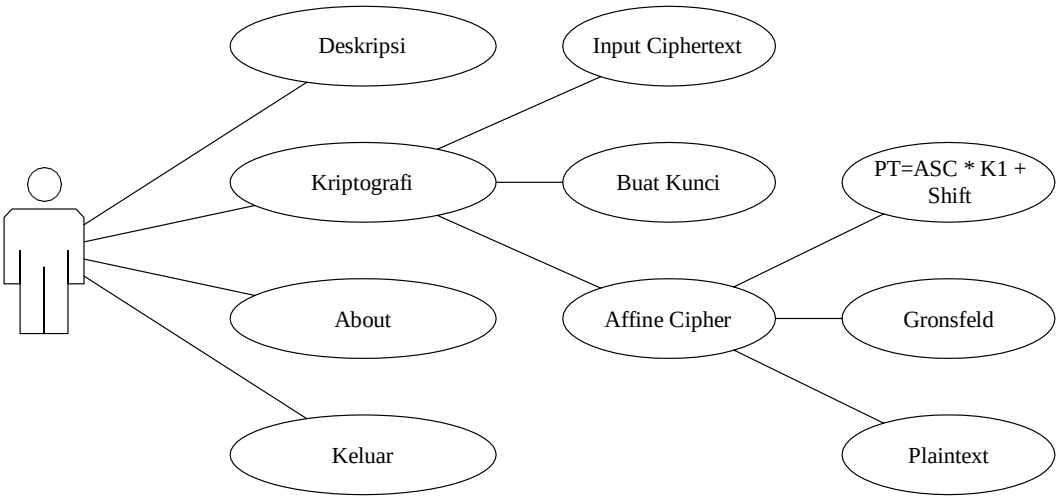
Digaram ini memberikan petunjuk dan alur penelitian sehingga setiap arah penelitian dijelaskan secara detail. Diagram ini juga dapat memberikan kemudahan bagi penulis dan pembaca untuk memberikan hasil yang sesuai dengan yang sudah direncanakan sebelumnya.

3.4.1 Use Case Diagram

Use case diagram adalah diagram perilaku atau dinamis di UML. *Use case diagram* memodelkan fungsionalitas suatu sistem menggunakan aktor dan use case. Use case diagram dapat dilakukan untuk memberi gambaran program aplikasi yang dibuat. Hal ini bertujuan agar setiap menu pada program aplikasi dapat tergambarkan dengan baik dan terarah. Tujuannya agar program aplikasi tidak menghasilkan kesalahan pada saat uji coba program. Gambar 3.3 dan 3.4 adalah perancangan u*se case diagram* untuk algoritma Affine Cipher pada proses enkripsi dan dekripsi. Kedua proses memiliki cara yang sama tetapi menggunakan nilai parameter yang berbeda.



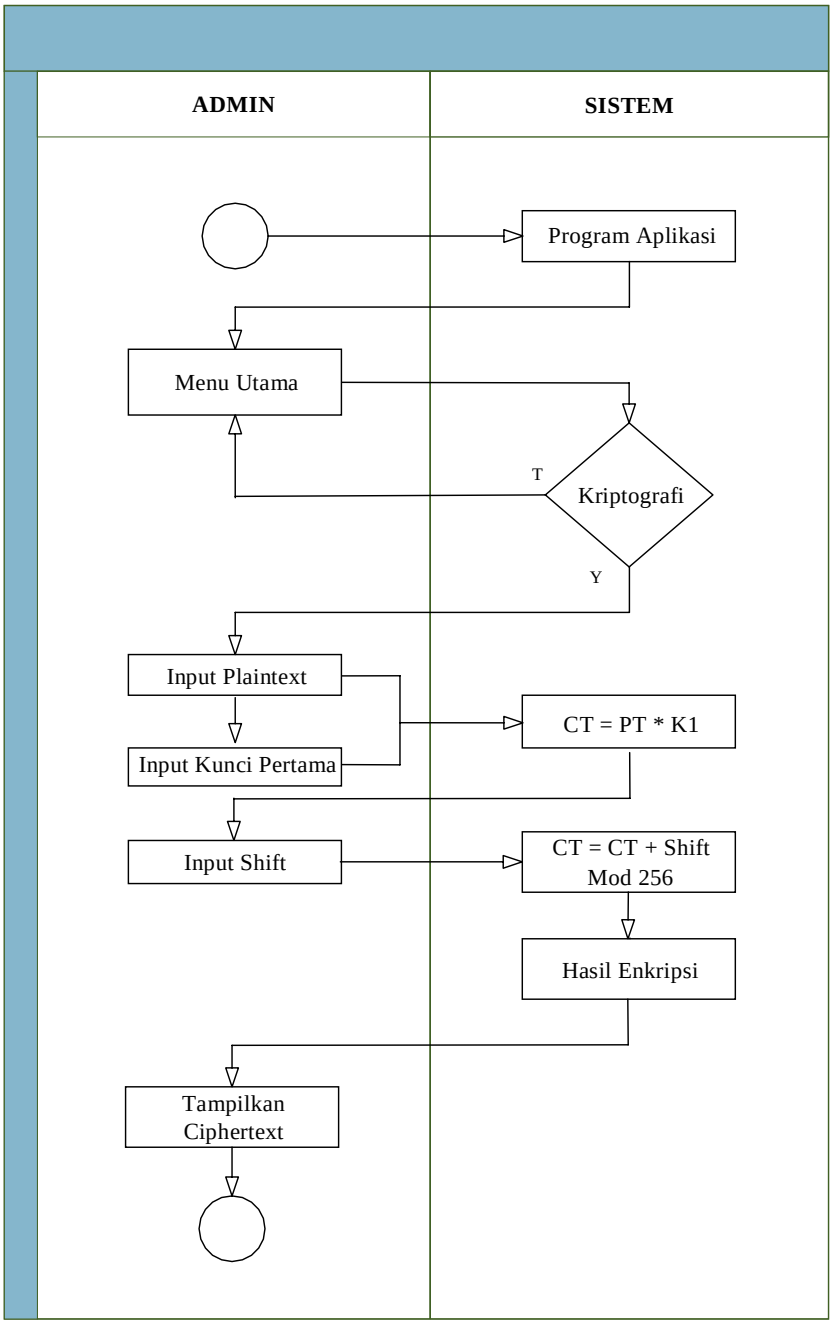
Gambar 3.3 Use Case Diagram proses enkripsi



Gambar 3.4 Use Case Diagram proses deskripsi

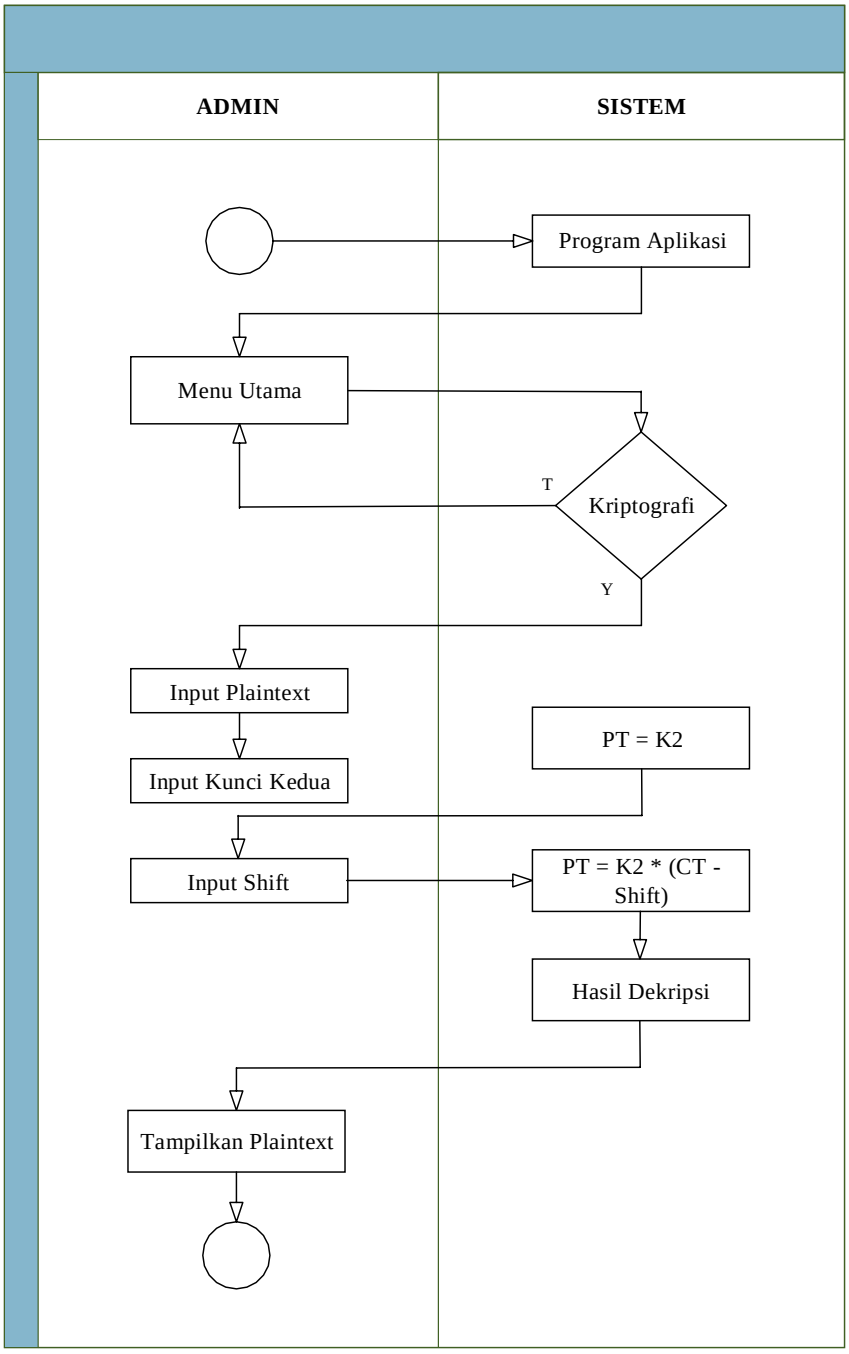
3.4.2 Activity Diagram

Gambar 3.5 adalah activity diagram algoritma Affine Cipher pada proses enkripsi.



Gambar 3.5 Activity Diagram proses enkripsi

Gambar 3.6 adalah activity diagram algoritma Affine Cipher pada proses dekripsi.

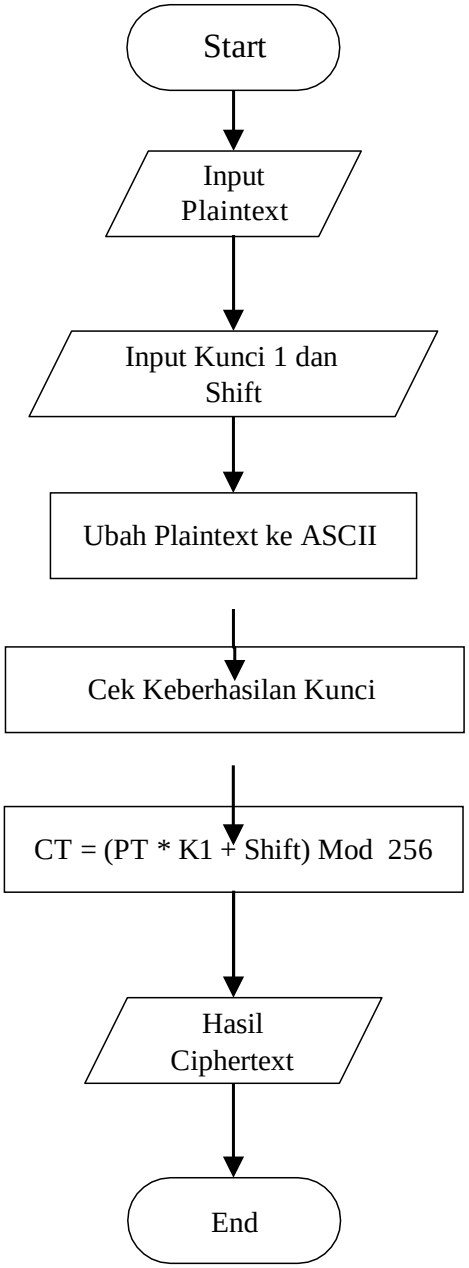


Gambar 3.6 Activity Diagram proses dekripsi

3.5 Flowchart Sistem

3.5.1 Flowchart Enkripsi

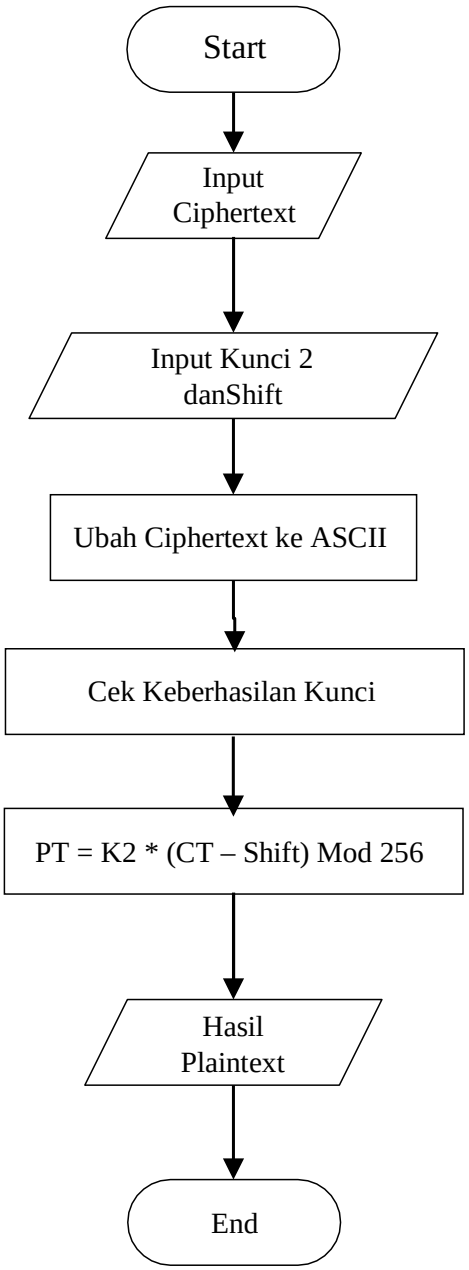
Flowchart enkripsi akan menerangkan cara kerja algoritma Affine Cipher. Gambar 3.7 adalah flowchart enkripsi.



Gambar 3.7 Flowchart enkripsi

3.5.2 Flowchart Dekripsi

Flowchart dekripsi akan menerangkan cara kerja algoritma Affine Cipher. Gambar 3.8 adalah flowchart dekripsi.



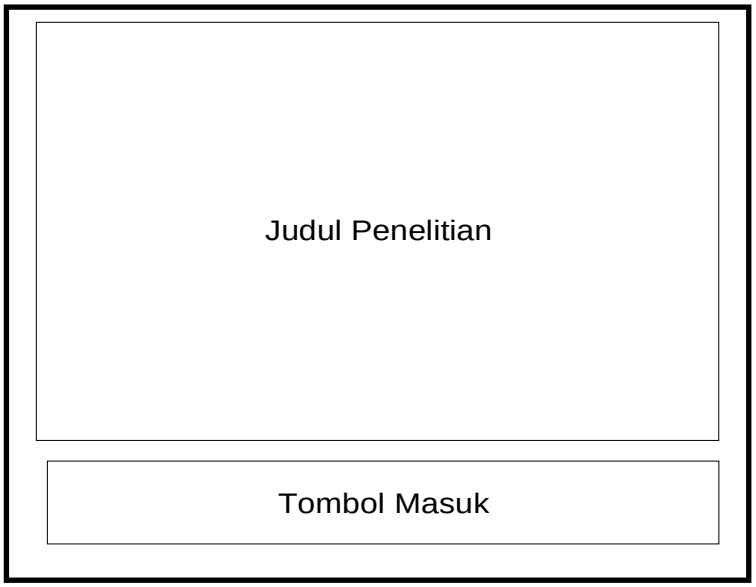
Gambar 3.8 Flowchart dekripsi

3.6 Perancangan Antarmuka

Perancangna antarmuka dilakukan untuk menentukan bentuk tampilan dari program aplikasi yang akan dibuat. Bentuk ini menggambarkan ilustrasi bagaimana tata letak komponen yang ada pada program aplikasi tersebut. Perancangan berfungsi untuk memberi efeisiensi dari penggunaan sumber daya.

3.6.1 Rancangan Halaman Judul

Rancangan halaman judul merupakan halaman yang akan terbuka saat program aplikasi dimulai. Gambar 3.9 adalah perancangan halaman judul.

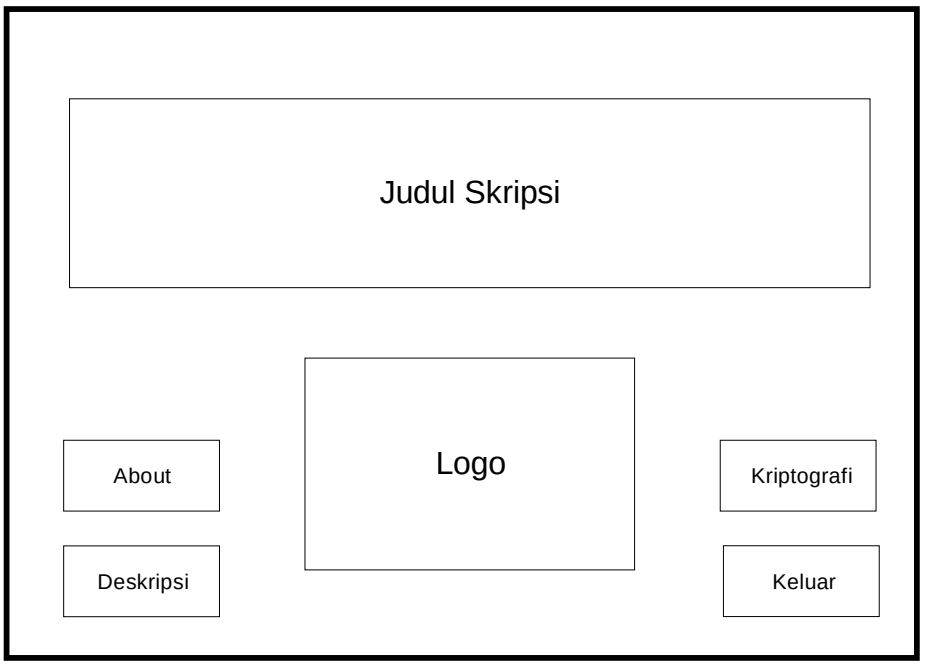


Gambar 3.9 Rancangan Halaman Judul

Rancangan halaman judul menampilkan judul skripsi penulis. Dengan menekan tombol masuk, program aplikasi akan berpindah ke menu berikutnya.

3.6.2 Rancangan Halaman Menu Utama

Rancangan menu utama merupakan menu yang dapat menampilkan beberapa pilihan yang berfungsi untuk membawa pengguna menuju menu-menu lainnya. Gambar 3.8 adalah hasil perancangan menu utama.



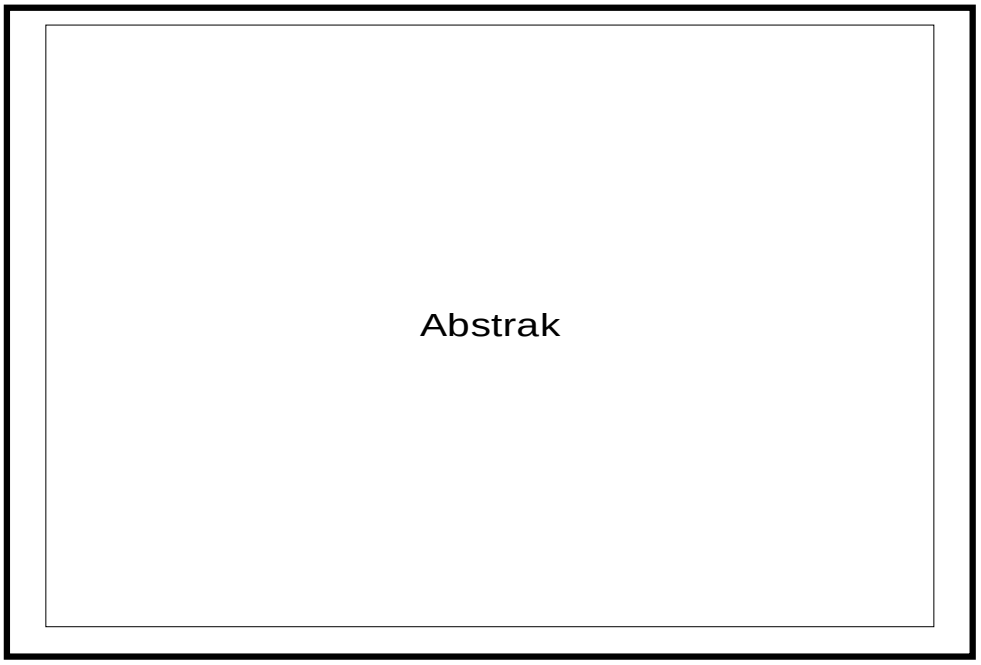
Gambar 3.10 Rancangan Halaman Menu Utama

Perancangan tersebut memiliki beberapa menu, antara lain.

1. Menu Kriptografi berfungsi untuk menghubungkan ke menu kriptografi.
2. Menu Deskripsi berfungsi untuk menampilkan menu Abstrak.
3. Menu About berfungsi untuk menghubungkan pengguna ke menu About.
4. Menu Keluar berfungsi untuk keluar dari program aplikasi.

3.6.3 Rancangan Halaman Menu Deskripsi

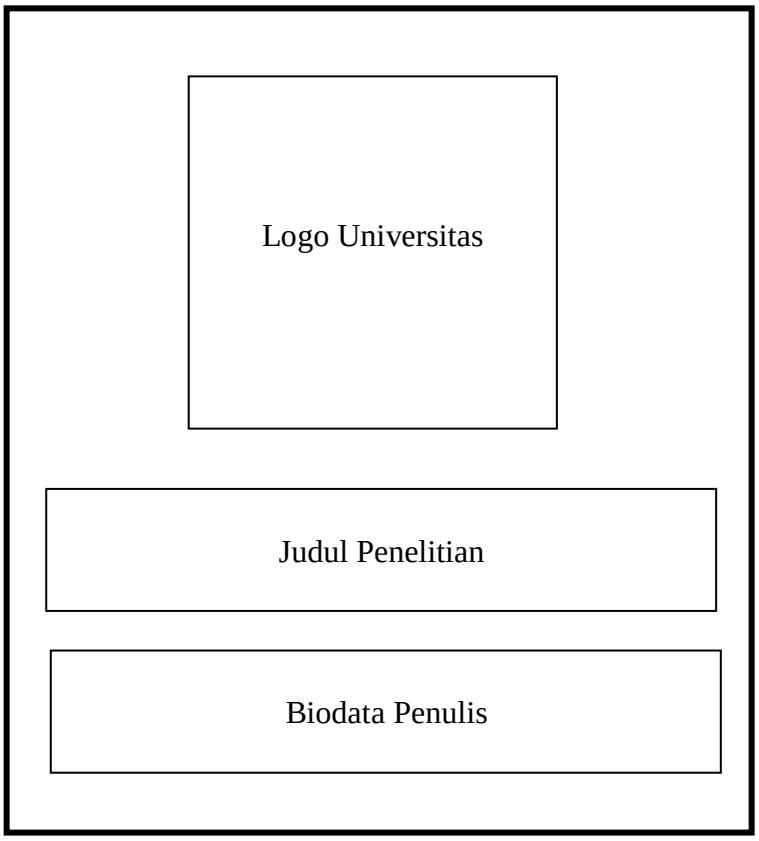
Rancangan halaman menu deskripsi berfungsi menampilkan keterangan singkat tentang pencapaian yang dilakukan oleh penulis dalam meneliti topik algoritma Affine Cipher. Gambar 3.11 adalah hasil perancangan menu Abstrak.



Gambar 3.11 Rancangan Halaman Menu Abtrak

3.6.4 Rancangan Halaman Menu About

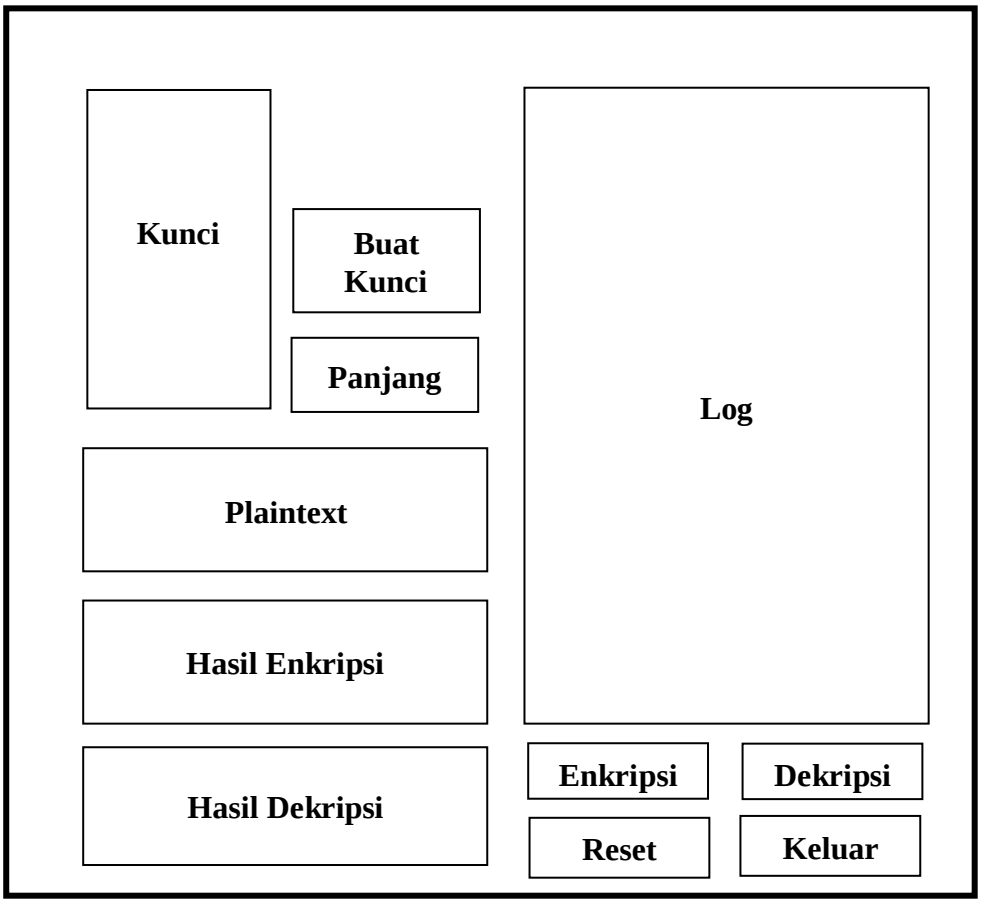
Rancangan halaman menu about berisikan tentang seputar tentang biodata penulis. Informasi ini hanya sebagai tampilan kepemilikan dari program aplikasi yang sudah dibuat. Halaman ini memiliki beberapa komponen. Gambar 3.12 adalah hasil perancangan menu About.



Gambar 3.12 Rancangan Halaman Menu About

3.6.5 Rancangan Halaman Menu Kriptografi

Rancangan halaman menu kriptografi adalah rancangan yang terpenting dari program aplikasi ini. Rancangan ini berfungsi untuk mengolah plaintext agar menghasilkan ciphertext. Proses kriptografi terjadi pada halaman ini. Proses enkripsi dan dekripsi dapat dilakukan secara bersamaan pada halaman yang sama. Hal ini bertujuan agar proses kriptografi menjadi lebih efisien. Gambar 3.13 adalah perancangan menu kriptografi.



Gambar 3.13 Rancangan Halaman Menu Kriptografi

Halaman menu kriptografi terdiri dari beberapa objek, antara lain:

1. Plaintext
2. Kunci
3. Hasil Enkripsi
4. Hasil Dekripsi
5. Panjang Teks
6. Tombol Enkripsi dan Dekripsi
7. Tombol Reset dan Keluar

BAB IV

HASIL DAN PEMBAHASAN

Hasil dan pembahasan merupakan uji coba apa yang sudah penulis selesaikan menurut perancangan bab sebelumnya. Dalam menyelesaikan program aplikasi, ada kebutuhan yang menjadi syarat utama. Kebutuhan tersebut adalah dimana penulis membutuhkan sistem yang akan membantu penulis dalam melaksanakan kegiatan penelitian.

4.1 Kebutuhan Sistem

Kebutuhan sistem merupakan kebutuhan yang paling utama dalam menyelesaikan program aplikasi. Kebutuhan ini adalah merupakan sarana dalam melakukan proses pembuatan program aplikasi.

4.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam penelitian ini antara lain adalah:

1. Processor i3 2.0 GHz
2. RAM 4 GB
3. HDD 320 GB
4. Keyboard dan Mouse
5. Monitor 14

4.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak juga tidak kalah pentingnya dengan perangkat keras. Hal ini dibutuhkan untuk mendukung penelitian. Berikut ini adalah daftar perangkat lunak yang digunakan, antara lain:

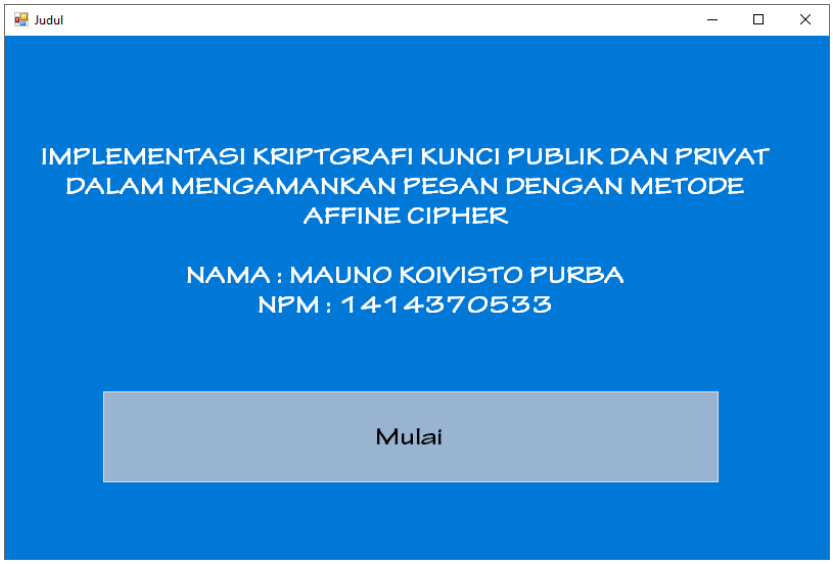
1. Microsoft Windows 7 SP1 sebagai sistem operasi
2. Mozilla Firefox sebagai browser
3. Microsoft Excel sebagai data editor
4. Microsoft Word sebagai pengolah kata
5. Microsoft Visual Studio 2010 sebagai editor programming
6. Adobe Photoshop sebagai pengolah gambar
7. Snipping Tool sebagai alat tangkap gambar

4.2 Implementasi Tampilan Antarmuka

Hasil program aplikasi yang terpenting adalah bagaimana program aplikasi tersebut dapat berinteraksi dengan pengguna. Program aplikasi berinteraksi dengan menampilkan Graphic User Interface sehingga pengguna dapat mengerti maksud dan tujuan si pembuat program. Beberapa bagian dalam implementasi antarmuka akan dijelaskan berikut ini.

4.2.1 Tampilan Halaman Judul

Halaman judul tampil pada saat program Affine Cipher dimuat dalam memori komputer. GUI akan ditampilkan dengan bentuk grafis berwarna biru. Gambar 4.1 adalah tampilan halaman judul.



Gambar 4.1 Tampilan Halaman Judul

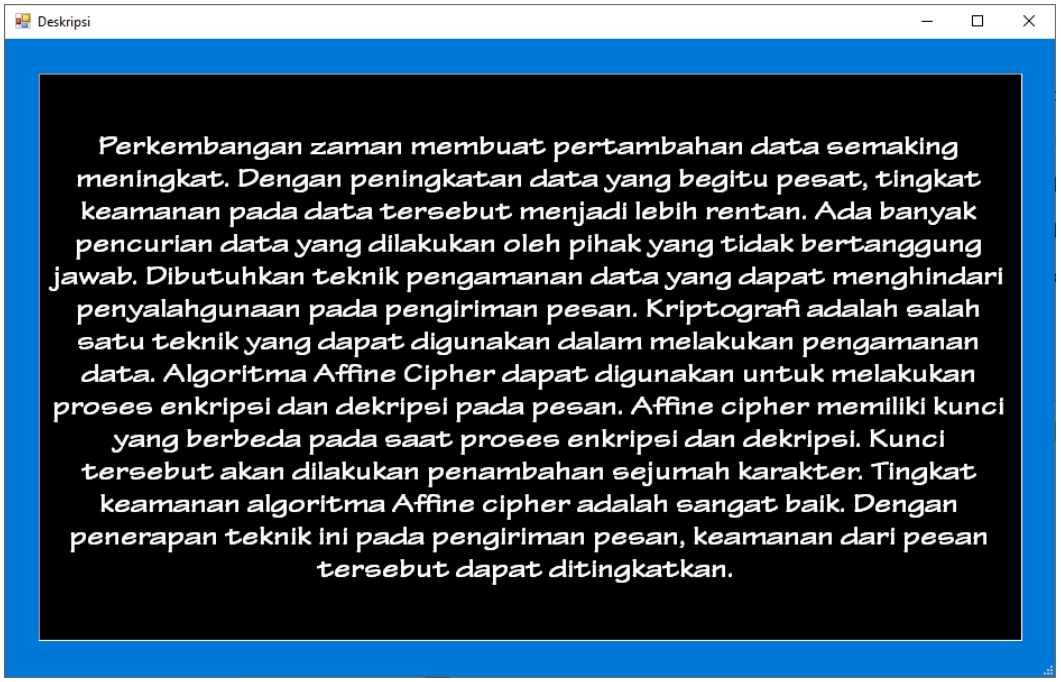
4.2.2 Tampilan Halaman Menu Utama

Implementasi Menu Utama merupakan menu memiliki beberapa fungsi lain yang pengguna dapat masuk. Gambar 4.1 adalah hasil implementasi Menu Utama.

Gambar 4.2 Tampilan Halaman Menu Utama

4.2.3 Tampilan Halaman Menu Deskripsi

Halaman menu deskripsi adalah halaman yang berisi abstrak dari penelitian yang penulis lakukan. Pada menu ini akan diceritakan tentang rumusan masalah, hasil dan sedikit kesimpulan dari penelitian. Penjelasan singkat akan algoritma Affine Cipher turut disertakan pada halaman menu abstrak untuk memberikan pengguna kemudahaan mengetahui maksud dan tujuan dari penelitian ini. Gambar **4.3** adalah hasil tampilan dari halaman deskripsi.



Gambar 4.3 Tampilan Halaman Menu Deskripsi

4.2.4 Tampilan Halaman Menu About

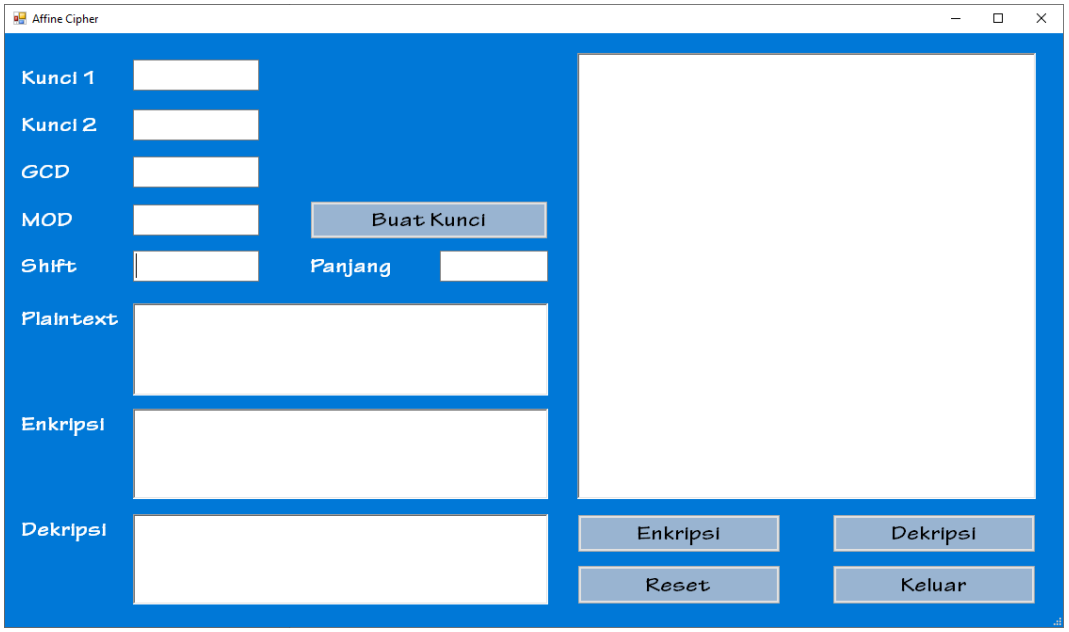
Halaman about menampilkan informasi jati diri penulis dan logo Unpab. Halaman ini memiliki sebuah objek *label* dan sebuah objek *picturebox*. Gambar 4.4 adalah tampilan dari halaman menu tentang.

Gambar 4.4 Tampilan Halaman Menu About

4.2.5 Tampilan Halaman Menu Kriptografi

Menu kriptografi mengizinkan pengguna untuk melakukan proses enkripsi dan dekripsi terhadap plaintext. Plaintext yang diproses adalah teks yang langsung diinputkan pada textbox tersebut. Proses algoritma Afiine Cipher menggunakan beberapa textbox dalam pembentukan kunci yang akan digunakan pada proses enkripsi dan dekripsi. Gambar 4.5 adalah tampilan dari halaman menu kriptografi.

.



Gambar 4.5 Tampilan Halaman Menu Kriptografi

4.2.6 Hasil Enkripsi

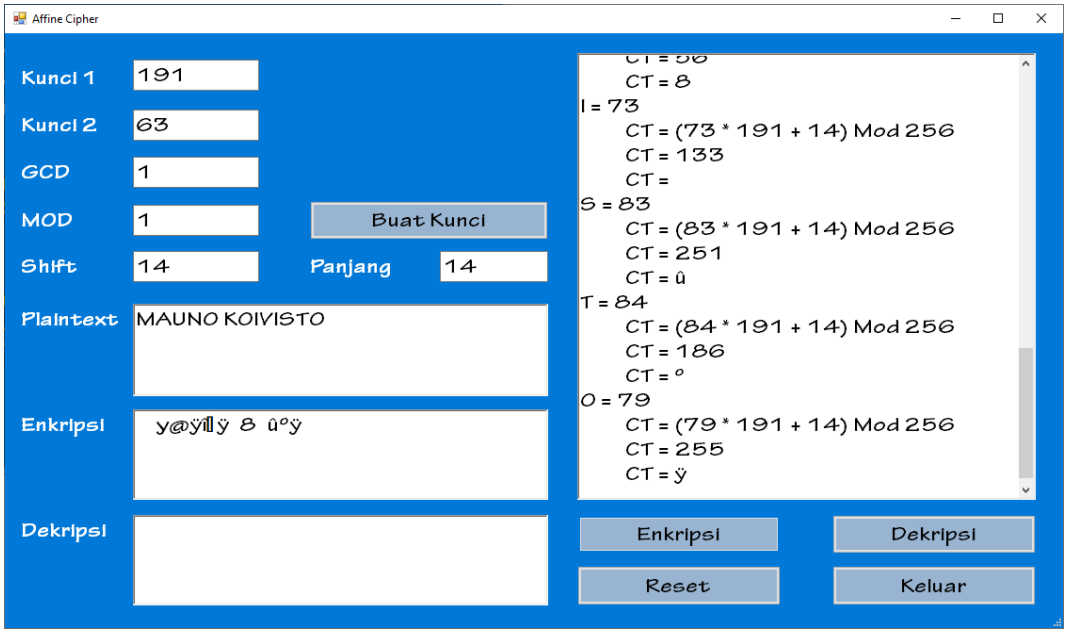
Ada dua bagian yang harus diselesaikan sebelum mendapatkan ciphertext, antara lain:

1. Pembentukan Kunci
2. Proses Enkripsi

Pembentukan kunci melibatkan tiga buah parameter yang berguna untuk mendapatkan kunci yang pas, antara lain:

1. Kunci 1 (Enkripsi)
2. Kunci 2 (Dekripsi)
3. Shift (Pergeseran)

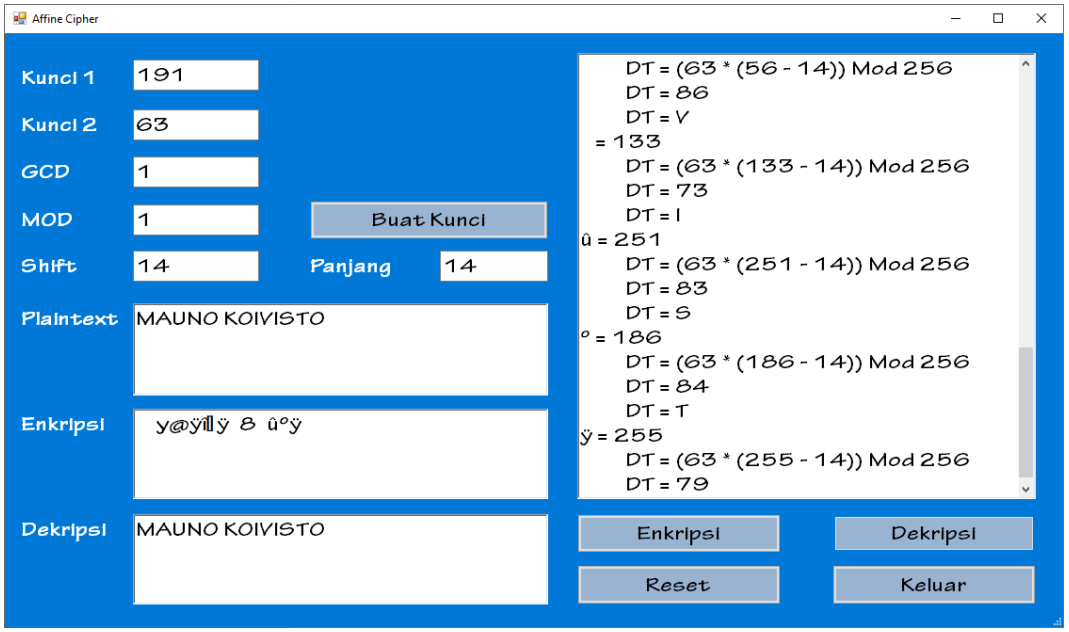
Gambar 4.6 adalah tampilan dari hasil perhitungan proses enkripsi dengan algoritma Beaufort dan Gronsfeld Cipher.



Gambar 4.6 Tampilan Halaman Enkripsi

4.2.7 Hasil Dekripsi

Hasil dekripsi dapat terjadi apabila pembentukan kunci sebelumnya sudah dilakukan. Proses enkripsi menggunakan Kunci 2 sebagai kunci dekripsi. Ciphertext akan diproses karakter demi karakter hingga menghasilkan deretan ASCII. ASCII tersebut akan diproses menggunakan formula Affine Cipher sehingga menghasilkan plaintext yang berbentuk byte. Hasil tersebut akan dikonversikan kembali menjadi karakter dan digabungkan sehingga menghasilkan plaintext kembali. Gambar 4.7 adalah tampilan dari hasil perhitungan dekripsi dari kombinasi kedua algoritma.



Gambar 4.7 Tampilan Halaman Dekripsi

4.3 Perhitungan Manual

Berikut ini adalah perhitungan manual yang penulis paparkan untuk menjelaskan cara kerja algoritma Affine Cipher. Proses enkripsi dan dekripsi dilakukan secara terpisah untuk melihat cara kerja algoritma Affine Cipher. Dalam melakukan perhitungan manual, kunci yang dibentuk harus benar agar pengembalian cihertext ke plaintext tidak mengalami kendala. Diasumsikan kunci yang diberikan berikut ini sudah bernilai benar.

Plaintext = MAUNO KOIVISTO

Kunci 1 = 191

Kunci 2 = 63

Shift = 14

Hasil Enkripsi

M = 77

CT = (77 * 191 + 14) Mod 256

CT = 129

CT = •

A = 65

CT = (65 * 191 + 14) Mod 256

CT = 141

CT = •

U = 85

CT = (85 * 191 + 14) Mod 256

CT = 121

CT = y

N = 78

CT = (78 * 191 + 14) Mod 256

CT = 64

CT = @

O = 79

CT = (79 * 191 + 14) Mod 256

CT = 255

CT = ÿ

= 32

CT = (32 * 191 + 14) Mod 256

CT = 238

CT = î

K = 75

CT = (75 * 191 + 14) Mod 256

CT = 3

CT = ———————————

O = 79

CT = (79 * 191 + 14) Mod 256

CT = 255

CT = ÿ

I = 73

CT = (73 * 191 + 14) Mod 256

CT = 133

CT = …

V = 86

CT = (86 * 191 + 14) Mod 256

CT = 56

CT = 8

I = 73

CT = (73 * 191 + 14) Mod 256

CT = 133

CT = …

S = 83

CT = (83 * 191 + 14) Mod 256

CT = 251

CT = û

T = 84

CT = (84 * 191 + 14) Mod 256

CT = 186

CT = º

O = 79

CT = (79 * 191 + 14) Mod 256

CT = 255

CT = ÿ

Ciphertext = •• y@ÿîÿ…8…ûºÿ

Kunci 1 = 191

Kunci 2 = 63

Shift = 14

Hasil Dekripsi

• = 129

DT = (63 * (129 - 14)) Mod 256

DT = 77

DT = M

• = 141

DT = (63 * (141 - 14)) Mod 256

DT = 65

DT = A

y = 121

DT = (63 * (121 - 14)) Mod 256

DT = 85

DT = U

@ = 64

DT = (63 * (64 - 14)) Mod 256

DT = 78

DT = N

ÿ = 255

DT = (63 * (255 - 14)) Mod 256

DT = 79

DT = O

î = 238

DT = (63 * (238 - 14)) Mod 256

DT = 32

DT =

= 3

DT = (63 * (3 - 14)) Mod 256

DT = 75

DT = K

ÿ = 255

DT = (63 * (255 - 14)) Mod 256

DT = 79

DT = O

… = 133

DT = (63 * (133 - 14)) Mod 256

DT = 73

DT = I

8 = 56

DT = (63 * (56 - 14)) Mod 256

DT = 86

DT = V

… = 133

DT = (63 * (133 - 14)) Mod 256

DT = 73

DT = I

û = 251

DT = (63 * (251 - 14)) Mod 256

DT = 83

DT = S

º = 186

DT = (63 * (186 - 14)) Mod 256

DT = 84

DT = T

ÿ = 255

DT = (63 * (255 - 14)) Mod 256

DT = 79

DT = O

Plaintext = MAUNOKOIVISTO

BAB V

PENUTUP

5.1 Kesimpulan

Ada beberapa kesimpulan yang dapat ditarik setelah melakukan penelitian pada algoritma Affine Cipher, antara lain:

1. Algoritma Affine Cipher memiliki dua buah kunci yang digunakan pada proses enkripsi dan dekripsi.
2. Pergeseran/shift merupakan teknik tambahan yang ada pada algoritma Affine Cipher untuk meningkatkan keamanan.
3. Masing-masing pengirim dan penerima memiliki kunci masing-masing sehingga kunci untuk enkripsi berbeda dengan kunci untuk dekripsi.
4. Ciphertext yang dihasilkan masih dalam ruang lingkup tabel ASCII karena hasil enkripsi mengalami proses modulo 256.

5.2 Saran

Adapun beberapa saran yang dapat dikemukakan untuk pengembangan program aplikasi adalah sebagai berikut:

1. Algoritma Affine Cipher dapat ditingkatkan keamanannya dengan menerapkan skema Three-pass Protocol.
2. Program aplikasi dapat digunakan secara online dan mobile.

DAFTAR PUSTAKA

Buckbee, M. (2019). Data Security: Definition, Explanation and Guide. Retrieved November 20, 2019, from Varonis website: https://www.varonis.com/blog/data-security/

Fachri, barany, agus perdana windarto, and ikhsan parinduri. "penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik." jepin (jurnal edukasi dan penelitian informatika) 5.2 (2019): 202-208.

Fachri, b., windarto, a. P., & parinduri, i. (2019). Penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik. Jepin (jurnal edukasi dan penelitian informatika), 5(2), 202-208.

Fachri, barany; windarto, agus perdana; parinduri, ikhsan. Penerapan backpropagation dan analisis sensitivitas pada prediksi indikator terpenting perusahaan listrik. Jepin (jurnal edukasi dan penelitian informatika), 2019, 5.2: 202-208.

Gabbrielli, M., & Martini, S. (2010). *Programming Languages: Principles and Paradigms*. https://doi.org/10.1007/978-1-84882-914-5

Hamdi, nurul. "model penyiraman otomatis pada tanaman cabe rawit berbasis programmable logic control." jurnal ilmiah core it: community research information technology 7.2 (2019).

Information, S. (2019). Symmetric vs. Asymmetric Encryption – What are differences? Retrieved November 20, 2019, from https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

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. (2017). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.

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

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

Permana, aminuddin indra. "kombinasi algoritma kriptografi one time pad dengan generate random keys dan vigenere cipher dengan kunci em2b." (2019).

Putra, randi rian. "sistem informasi web pariwisata hutan mangrove di kelurahan belawan sicanang kecamatan medan belawan sebagai media promosi." jurnal ilmiah core it: community research information technology 7.2 (2019).

Putra, randi rian, et al. "decision support system in selecting additional employees using multi-factor evaluation process method." (2019).

Putra, randi rian. "implementasi metode backpropagation jaringan saraf tiruan dalam memprediksi pola pengunjung terhadap transaksi." jurti (jurnal teknologi informasi) 3.1 (2019): 16-20.

Rouse, M., Rosencrance, L., & Cobb, M. (2019). What is Asymmetric Cryptography? Retrieved November 20, 2019, from TechTarget website: https://searchsecurity.techtarget.com/definition/asymmetric-cryptography

Saputra, muhammad juanda, and nurul hamdi. "rancang bangun aplikasi sejarah kebudayaan aceh berbasis android studi kasus dinas kebudayaan dan pariwisata aceh." journal of informatics and computer science 5.2 (2019): 147-157

Sidik, a. P., efendi, s., & suherman, s. (2019, june). Improving one-time pad algorithm on shamir's three-pass protocol scheme by using rsa and elgamal algorithms. In journal of physics: conference series (vol. 1235, no. 1, p. 012007). Iop publishing.

Sitepu, n. B., zarlis, m., efendi, s., & dhany, h. W. (2019, august). Analysis of decision tree and smooth support vector machine methods on data mining. In journal of physics: conference series (vol. 1255, no. 1, p. 012067). Iop publishing.

Smirnoff, P., & Turner, D. M. (2019). Symmetric Key Encryption - why, where and how it's used in banking. Retrieved November 20, 2019, from Cryptomathic website: https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking

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

Stallings, William. (2005). *Cryptography and Network Security Principles and Practices* (4th ed.). Prentice Hall.

Sukmawati, R., & Priyadi, Y. (2019). Perancangan Proses Bisnis Menggunakan UML Berdasarkan Fit/Gap Analysis Pada Modul Inventory Odoo. *INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, *3*(2), 104. https://doi.org/10.29407/intensif.v3i2.12697

Tasril, v., wijaya, r. F., & widya, r. (2019). Aplikasi pintar belajar bimbingan dan konseling untuk siswa sma berbasis macromedia flash. Jurnal informasi komputer logika, 1(3).

Technopedia. (2019). Unified Modeling Language (UML). Retrieved from Technopedia website: https://www.techopedia.com/definition/3243/unified-modeling-language-uml

Uml-diagrams.org. (2019). Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors). Retrieved November 3, 2019, from https://www.uml-diagrams.org/use-case-diagrams.html

UTM. (2019). Concept: Use-Case Model. Retrieved September 19, 2019, from

Univesidad Technologica de la Mixteca website: http://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/concepts/use_case_model_CD178AF9.html

Wasserkrug, S., Dalvi, N., Munson, E. V., Gogolla, M., Sirangelo, C., Fischer-Hübner, S., … Snodgrass, R. T. (2019). Unified Modeling Language. In

Encyclopedia of Database Systems (pp. 3232–3239). https://doi.org/10.1007/978-0-387-39940-9_440

Zwass, V. (2019). Information System. Retrieved November 20, 2019, from Britannica website: https://www.britannica.com/topic/information-system