ANALISIS ENKRIPSI DAN DESKRIPSI DATA TEXT MENGGUNAKAN ALGORITMA MERKLE HELLMAN

Disusun dan Disajikan Untuk Memenuhi Persyaratan Ujian
Akhir Memperoleh Gelar Sarjana Komputer
Fakultas Sains Dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA : **RINALDI MUHAMMAD PARLINDUNGAN**

N.P.M : **1514370215**

Program Studi : **SISTEM KOMPUTER**

PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019

ABSTRAK

Rinaldi Muhammad Parlindungan
Analisis Enkripsi Dan Deskripsi Data Text Menggunakan Algoritma Merkle Hellman
2019

Keamanan dan kerahasiaan data merupakan suatu aspek yang sangat penting dalam proses pertukaran pesan atau informasi. Suatu pesan yang sifatnya rahasia membutuhkan suatu sistem penyimpanan dan pengiriman data atau file agar tidak mudah terbaca dan diketahui semua orang. Ada berbagai macam cara untuk mengamankan data atau file, salah satunya adalah menggunakan metode kriptografi. Merkle Hellman merupakan algoritma klasik untuk menyandikan sebuah plaintext dengan cara substitusi sehingga dalam memecahkan pesan tersebut akan terasa susah. Algoritma Merkle Hellman merupakan salah satu metode kriptografi berbasis protokol. Protokol adalah aturan yang berisi tentang langkah-langkah yang melibatkan dua kunci yang dibuat untuk menyelesaikan suatu kegiatan. Dalam kriptografi, protokol digunakan oleh orang-orang yang terlibat, seperti untuk proses otentifikasi, pengaktifan bilangan acak, bahkan untuk berbagi dan bertukar informasi yang bersifat rahasia. Dalam penelitian ini untuk penggunaan algoritma Merklee Hellman menyimpulkan Pada proses enkripsi dengan metode Merkell Hellman, kunci public harus di jumlahkan lalu untuk kunci private harus lebih besar nilainya dari kunci public. Kunci yang dipakai pada proses enkripsi dan deskripsi hanya 8 kunci, dan menggunakan kunci public ataupun private.

Kata Kunci : Merklee Hellman, Kriptografi, Pengamanan.

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Judul **Halaman**

DAFTAR TABEL

Judul **Halaman**

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Keamanan dan kerahasiaan data merupakan suatu aspek yang sangat penting dalam proses pertukaran pesan atau informasi. Suatu pesan yang sifatnya rahasia membutuhkan suatu sistem penyimpanan dan pengiriman data atau *file* agar tidak mudah terbaca dan diketahui semua orang. Ada berbagai macam cara untuk mengamankan data atau *file*, salah satunya adalah menggunakan metode kriptografi.

Masalah *knapsack* adalah masalah lengkap NP di optimasi kombinatorial. Yang ditunjuk oleh masalah *knapsack* (ransel) item yang paling berguna dari sejumlah item mengingat bahwa yang ransel atau ransel memiliki kapasitas tertentu. Masalah ransel secara luas digunakan untuk memodelkan solusi masalah industri seperti kriptografi kunci publik. Masalah 0-1 ransel menyatakan bahwa jika ada ransel dengan kapasitas tertentu dan sejumlah item yang perlu dimasukkan ke dalam ransel. Setiap item memiliki nilai dan berat yang terkait dengannya. Yang ditunjuk oleh masalah ransel item yang dapat dimasukkan ke dalam ransel sehingga nilai semua item dimaksimalkan dan berat tidak meningkatkan total kapasitas ransel.

Merkle Hellman merupakan algoritma klasik untuk menyandikan sebuah *plaintext* dengan cara substitusi sehingga dalam memecahkan pesan tersebut akan terasa susah (Aminudin, 2017). Penelitian ini menggunakan pemrograman *Visual Basic.Net* 2010.

Algoritma Merkle Hellman merupakan salah satu metode kriptografi berbasis protokol. Protokol adalah aturan yang berisi tentang langkah-langkah yang melibatkan dua kunci yang dibuat untuk menyelesaikan suatu kegiatan. Dalam kriptografi, protokol digunakan oleh orang-orang yang terlibat, seperti untuk proses otentifikasi, pengaktifan bilangan acak, bahkan untuk berbagi dan bertukar informasi yang bersifat rahasia. Pengirim dan penerima pesan melakukan penukaran public dan rahasia untuk mengekripsikan pesan tersebut. Pada dasarnya, *Algoritma Merkle Hellman* di implementasikan dengan menggunakan satu algoritma enkripsi dan dekripsi yang telah disepakati oleh kedua belah pihak.

Menurut Peneliti sebelumnya, (Murdani, 2017) Algoritma Merkle-Hellman Knapsack menggunakan kunci privat dan kunci publik dalam proses kriptografinya, metode ini memiliki pengamanan ganda sehingga sulit untuk ditembus. Menurut (Aminudin, 2018) Implementasi algoritma knapsack dan kombinasi knapsack dan logaritma diskrit dapat diaplikasikan pada aplikasi chat. Aplikasi chat tersebut dapat melindungi pesan agar pesan tersebut lebih aman dan tidak mudah dibaca oleh orang yang tidak berhak karena pesan tersebut ketika dikirim berbentuk ciphertext yang sudah dienkripsi menggunakan kunci publik dan dapat di deskripsi oleh orang yang memiliki kunci privat.

Berdasarkan latar belakang yang telah penulis uraikan di atas, maka penulis tertarik untuk memilih judul **"*Analisis Enkripsi Dan Deskripsi Data Text Menggunakan Algoritma Merkle Hellman*".**

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas dapat penulis simpulkan bahwa yang menjadi pokok permasalahan dalam pembahasan ini adalah sebagai berikut:

1. Bagaimana merancang sebuah aplikasi keamanan data teks yang dengan algoritma *Merkle Hellman*?
2. Bagaimana menerapkan metode algoritma *Merkle Hellman* dalam proses keamanan data teks yang bersifat rahasia*?*
3. Bagaimana melindungi pesan dengan proses enkripsi *Merkel Hellman?*

1.3 Batasan Masalah

Berdasarkan perumusan masalah diatas maka penulis melakukan pembatasan masalah yang akan dibahas sebagai berikut:

1. Implementasi enkripsi dan dekripsi hanya berupa teks.
2. Program yang dibahas menggunakan pemrograman Visual Basic.Net 2010.
3. Konversi karakter menggunakan bilangan biner dari tabel ASCII.
4. Modulus Invers yang digunakan Modular r^{-1} mod q.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini dengan menggunakan algoritma *Merkle Hellman* ini yang ingin dicapai adalah sebagai berikut:

1. Merancang aplikasi keamanan data teks dengan algoritma *Merkle Hellman*.
2. Memperkuat keamanan data teks yang bersifat rahasia.
3. Untuk membuat sistem aplikasi keamanan data yang bersifat melindungi data.

1.5 Manfaat Penelitian

Adapun manfaat dalam penelitian ini yang diperoleh dari penerapan algoritma *Merkle Hellman* adalah sebagai berikut:

1. Kerahasiaan data yang dikirim dan diterima lebih aman
2. Sebagai media pembelajaran dalam bidang keamanan informasi.

1.6 Metodologi Penelitian

Dalam metodologi penelitian ini peneliti menggunakan beberapa metode dalam pengumpulan data untuk melengkapi hasil penelitian ini. Adapun metode tersebut sebagai berikut:

1. Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan dalam penelitian ini adalah sebagai berikut:

a. Studi Pustaka yaitu pengumpulan data yang diperoleh dari sumber tertulis berupa buku-buku, artikel ilmiah, dan penelitian-penelitian yang berkaitan dengan judul penelitian.

b. Studi literature yaitu pengumpulan data yang diperoleh dari literature, jurnal, paper, dan bacaan-bacaan dari berbagai sumber yang berkaitan dengan judul penelitian.

2. Metode pengembangan dan perancangan sistem

Pada kasus ini menggunakan metode *Merkle Hellman* yang merupakan salah satu contoh metode kriptografi kunci simetris dengan model penggantian karakter. Metode *Merkle Hellman* merupakan metode yang menggunakan kunci berupa angka, sedangkan *Merkle Hellman* merupakan metode yang menggunakan kunci abjad sebagai kunci penyandian untuk penggantian karakter dari pesan rahasia yang dikirim. Pada penelitian ini diharap dapat memberi solusi terhadap masalah keamanan data yang lebih aman.

1.7 Sistematika Penulisan

Secara garis besar sistematika penulisan tugas akhir ini terdiri dari lima bab yaitu sebagai berikut:

BAB I PENDAHULUAN

Pada bab pendahuluan ini, akan menguraikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodelogi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab landasan teori ini, akan memaparkan teori-teori yang digunakan sebagai panduan dalam menyelesaikan skripsi sesuai dengan judul yang diteliti dan dapat diperoleh dari berbagai sumber.

BAB III ANALISA MASALAH DAN RANCANGAN PROGRAM

Pada bab analisa masalah dan rancangan program ini, menjelaskan tentang gambaran pembahasan permasalahan yang terjadi serta perancangan sistem yang ingin diselesaikan.

BAB IV IMPLEMENTASI DAN ANALISA HASIL UJI COBA PROGRAM

Pada bab implementasi dan analisa hasil uji coba program ini, membahas tentang hasil implementasi yang dibuat serta melakukan analisa terhadap hasil tersebut.

BAB V PENUTUP

Pada bab penutup ini, berisi tentang kesimpulan dan saran yang diperoleh dari hasil penelitian untuk pengembangan serta perbaikan yang di perlukan dari hasil penelitian ini.

BAB II

LANDASAN TEORI

2.1 Keamanan Data

Pada zaman teknologi informasi sekarang, data atau informasi merupakan suatu asset yang sangat berharga dan harus dilindungi. Hal ini juga diikuti oleh kemajuan teknologi komputer. Kemajuan teknologi komputer membantu semua aspek kehidupan manusia. Dengan adanya kemajuan dalam teknologi informasi, komunikasi dan komputer maka kemudian muncul masalah baru, yaitu masalah keamanan akan data dan informasi dan dalam hal ini akan membuka peluang bagi orang-orang yang tidak bertanggung jawab untuk menggunakannya sebagai tindak kejahatan. Dan tentunya akan merugikan pihak tertentu. Dalam keamanan data ada beberapa aspek yang berkaitan dengan persyaratan kemanan yaitu (Pabokory, 2015):

1. *Secrecy*. Berhubungan dengan akses membaca data dan informasi. Data dan informasi di dalam suatu sistem komputer hanya dapat diakses dan dibaca oleh orang yang berhak.
2. *Integrity*. Berhubungan dengan akses merubah data dan informasi. Data dan informasi di dalam suatu sistem komputer hanya dapat diubah oleh orang yang berhak.

3. *Availability*. Berhubungan dengan ketersediaan data dan informasi. Data dan informasi yang berada dalam suatu sistem komputer tersedia dan dapat dimanfaatkan oleh orang yang berhak. (Pabokory, 2015).
4. Lebih lanjut menurut (Pabokory, 2015), terdapat lima langkah kemanan komputer yang baik untuk diperhitungkan yaitu; aset, analisis resiko, perlindungan, alat dan prioritas.

2.2 *Kriptografi*

Kriptografi merupakan kata dari bahasa Yunani yaitu cryptography, terdiri dari dua suku kata yaitu kripto dan graphia. Kripto artinya menyembunyikan, sedangkan graphia artinya tulisan. Sehingga, bila digabungkan akan menjadi kata yang berarti menyembunyikan/merahasiakan tulisan. *Kriptografi* adalah suatu ilmu ataupun seni mengamankan pesan dan dilakukan oleh *cryptographer* (Anonim, 2014).

Menurut (Rhee, 2013). *kriptografi* digunakan untuk memastikan privasi dan autentifikasi data dalam komunikasi antar sistem komputer. Terdapat dua proses dasar dalam *kriptografi* yaitu:

1. *Enkripsi*, adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). (Pabokory, 2015).
2. *Deskripsi*, adalah kebalikan dari *Enkripsi* yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi awal. (Fresly, 2015).

Sebuah pesan atau data yang masih asli dan belum mengalami penyandian dikenal dengan istilah plaintext. Kemudian setelah disamarkan dengan suatu cara penyandian, maka plaintext ini disebut sebagai chipertext. Proses penyamaran dari plaintext ke *cipher*text disebut *Enkripsi* (encryption), dan proses pengembalian dari *cipher*text menjadi plaintext kembali disebut dekripsi (decryption). (Fresly, 2015). *File* yang dapat di*Enkripsi* dapat berupa teks, gambar maupun audio dan video.

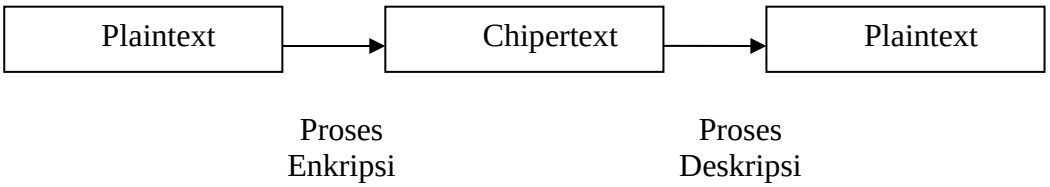
2.3 Macam-Macam *Kriptografi*

Kriptografi dibedakan menjadi 3 bagian yaitu *kriptografi* simetris, *kriptografi* asimetris dan fungsi hash satu arah. *Kriptografi* simetris disebut juga *kriptografi* kunci rahasia merupakan jenis *kriptografi* paling intuitif. Ini termasuk penggunaan kunci rahasia yang dikenal hanya pada pengguna komunikasi yang aman. *Kriptografi* asimetris sendiri berbeda dengan *kriptografi* simetris, dimana *kriptografi* asimetris ini menggunakan dua kunci yang berbeda, yaitu kunci publik dan kunci rahasia atau kunci pribadi. Kunci-kunci tersebut berhubungan secara matematis, tetapi tidak mungkin secara perhitungan untuk menarik kesimpulan satu dengan yang lain.

Fungsi *hash* satu arah, juga dikenal sebagai rangkuman pesan atau fungsi kompresi adalah fungsi matematis yang mengambil input panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap.

2.4 *Enkripsi*

Enkripsi merupakan hal yang sangat penting dalam *kriptografi* supaya keamanan data yang dikirimkan bisa terjaga kerahasiaannya. Pesan asli (plaintext) diubah menjadi kode-kode yang tidak dimengerti. *Enkripsi* bisa diartikan dengan chiper atau kode. Sama halnya dengan kita yang tidak mengerti sebuah kata, kita akan dapat melihatnya di dalam kamus atau daftar istilah-istilah. Berbeda halnya dengan *Enkripsi,* untuk mengubah plaintext ke bentuk chipertext, kita harus menggunakan algoritma yang dapat mengkodekan data yang kita inginkan. Berikut adalah penggambaran proses *Enkripsi*.



Gambar 2.1. Proses *Enkripsi* dan *Deskripsi*

(Sumber: Fresly, 2015)

2.5 *Kriptografi* Klasik

Menurut (Bishop, 2014). *kriptografi* klasik adalah *kriptografi* yang disebut juga sebagai *kriptografi* kunci tunggal atau *kriptografi* simetris yang menggunakan kunci yang sama untuk *Enkripsi* maupun *Deskripsi*. *Kriptografi* klasik merupakan *kriptografi* yang digunakan pada zaman dahulu sebelum komputer ditemukan atau

sudah ditemukan namun belum secanggih sekarang. *Kriptografi* ini melakukan pengacakan huruf pada kata terang / plaintext.

2.6 Algoritma Merkle Hellman

Merupakan salah satu algoritma kriptografi kunci-public awal yang ditemukan oleh Ralph Merkle dan Martin Hellman in 1978. Disebut juga algoritma Merkle-Hellman. Algoritma ini didasarkan pada persoalan 1/0 Knapsack Problem yang berbunyi:

Diberikan bobot knapsack adalah M. Diketahui n buah objek yang masing-masing bobotnya adalah w1, w2, …, wn. Tentukan nilai bi sedemikian sehingga

$$M = b1w1 + b2w2 + \ldots + bnwn$$

yang dalam hal ini, bi bernilai 0 atau 1. Jika bi = 1, berarti objek i dimasukkan ke dalam knapsack, sebaliknya jika bi = 0, objek i tidak dimasukkan. Dalam teori algoritma, persoalan knapsack termasuk ke dalam kelompok NP-complete. Persoalan yang termasuk NP-complete tidak dapat dipecahkan dalam orde waktu polinomial. Ide dasar dari algoritma knapsack adalah mengkodekan pesan sebagai rangkaian solusi dari persoalan knapsack. Setiap bobot wi di dalam persoalan knapsack merupakan kunci rahasia, sedangkan *bit-bit plainteks* menyatakan bi.

Contoh 1: Misalkan n = 6 dan w1 = 1, w2 = 5, w3 = 6, w4 = 11, w5 = 14, dan w6 = 20.

Plainteks: 111001010110000000011000

Plainteks dibagi menjadi blok yang panjangnya n, kemudian setiap *bit* di dalam blok dikalikan dengan wi yang berkorepsonden sesuai dengan persamaan (1):

Blok *plainteks* ke-1 : 111001

Kriptogram : $((1 \times 1) + (1 \times 5) + (1 \times 6) + (1 \times 20) = 32$

Blok *plainteks* ke-2 : 010110

Kriptogram : $(1 \times 5) + (1 \times 11) + (1 \times 14) = 30$

Blok *plainteks* ke-3 : 000000

Kriptogram : 0

Blok *plainteks* ke-4 : 011000

Kriptogram : $(1 \times 5) + (1 \times 6) = 11$

Jadi, *Ciphertek*s yang dihasilkan: 32 30 0 11

2.7 *One Time Pad (OTP)*

Algoritma *One Time Pad* (OTP) merupakan algoritma berjenis *Symetric key* yang artinya bahwa kunci yang digunakan untuk melakukan enkripsi dan deskripsi merupakan kunci yang sama. Dalam proses enkripsi, algoritma ini menggunakan cara *stream Cipher* yang berasal dari hasi XOR antara *bit plaintext* dan *bit key*. Pada metode ini *plain text* diubah kedalam kode ASCII dan kemudian dikenakan operasi XOR terhadap kunci yang sudah diubah ke dalam kode ASCII. (Hamokwarong, 2014).

One-time pad adalah salah satu stream *Cipher* klasik yang secara matematis terbukti sempurna aman. *Cipher* teksnya tidak mungkin dapat dipecahkan. Keamanan algoritma *one-time pad* terletak pada penggunaan barisan bilangan acak sejati (*trully random*) sebagai kunci enkripsi, panjang kunci sama dengan panjang pesan dan tidak ada perulangan kunci sebagaimana pada pada *Vernam Cipher* atau *Vigenere Cipher*. (Munir, 2014).

Sayangnya *one-time pad* tidak dapat diimplementasikan secara praktis sebab pembangkitan bilangan acak sejati tidak dapat diulang kembali di sisi penerima pesan. Oleh karena itu kunci (*pad*) harus dikirim melalui saluran komunikasi yang kedua (misalnya melalui kurir), sayangnya saluran kedua itu umumnya lambat dan ongkosnya mahal. One-time pad masih dapat diterapkan namun kunci yang berupa barisan bilangan acak diganti dengan barisan bilangan semi-acak (*pseudo-random*) dengan syarat barisan kunci itu tidak boleh berulang. (Munir, 2014).

2.8 Algoritma

Penyelesaian permasalahan dengan menggunakan alat bantu system computer paling tidak akan melibatkan lima tahapan, yaitu:

1. Analisis masalah
2. Merancang algoritma
3. Membuat program computer
4. Menguji hasil program computer dan dokumentasi.

Poin kedua menerangkan bahwa dalam perancangan sebuah system computer dibutuhkan adanya perancangan algoritma. Sehingga setelahnya dapat dilanjutkan ke tahap-tahap berikutnya hingga dokumentasi.

Algoritma adalah Sistim kerja komputer memiliki brainware, hardware, dan software. Tanpa salah satu dari ketiga sistim tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada softwarekomputer. Software terbangun atas susunan program (silahkan baca mengenai pengertian program) dan syntax (cara penulisan/pembuatan program). Untuk menyusun program atau syntax, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu software. Maka, Algoritma berperan penting dalam penyusunan program atau syntax tersebut.

Pengertian Algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu. Dalam dunia komputer,Algoritma sangat berperan penting dalam pembangunan suatu software. Dalam dunia sehari-hari, mungkin tanpa kita sadari Algoritma telah masuk dalam kehidupan kita.

Pengertian Algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu.

Algoritma adalah kunci dari bidang ilmu komputer, dan pada dasarnya setiap hari kita melakukan aktivitas algoritma. Kata algoritma berasal dari sebutan Algorizm (Abu Abdullah Muhammad Ibn Musa Al Khwarizmi), ahli matematika Uzbeki

a. Algoritma adalah urutan langkah-langkah berhingga untuk memecahkan masalah logika atau matematika.
b. Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan.
c. Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.
d. Algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah.

Pembuatan algoritma harus selalu dikaitkan dengan:

a. Kebenaran algoritma
b. Kompleksitas (lama dan jumlah waktu proses dan penggunaan memori)

Kriteria Algoritma yang baik:

a. Tepat, benar, sederhana, standar dan efektif
b. Logis, terstruktur dan sistematis
c. Semua operasi terdefinisi
d. Semua proses harus berakhir setelah sejumlah langkah dilakukan
e. Ditulis dengan bahasa yang standar dengan format pemrograman agar mudah untuk diimplementasikan dan tidak menimbulkan arti ganda.

2.9 *Unified Modeling Language* (UML)

1. Pengenalan UML

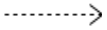
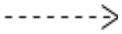
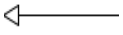
Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain yang berisi sintak dalam memodelkan sistem secara visual (Haviluddin, 2015). Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktur, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan *real time* (waktu nyata).

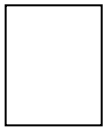
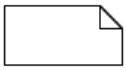
Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML).

2. *Use Case Diagram*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram *use case diagram* (Haviluddin, 2015).

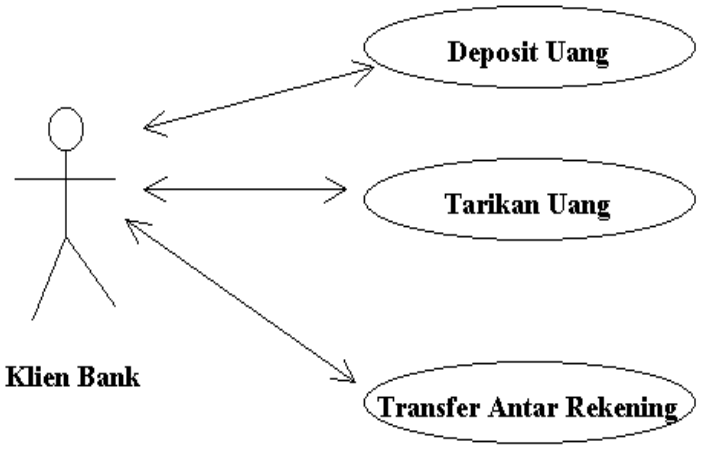
Tabel 2.1 Simbol *Use Case Diagram*

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 (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
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 aktor
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

Sumber : (Gellysa Urva, 2015)

Contoh Use Case Diagram :

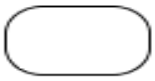


Gambar 2.2. Contoh *Use Case Diagram*
Sumber : (Haviluddin, 2015)

3. *Activity Diagram*

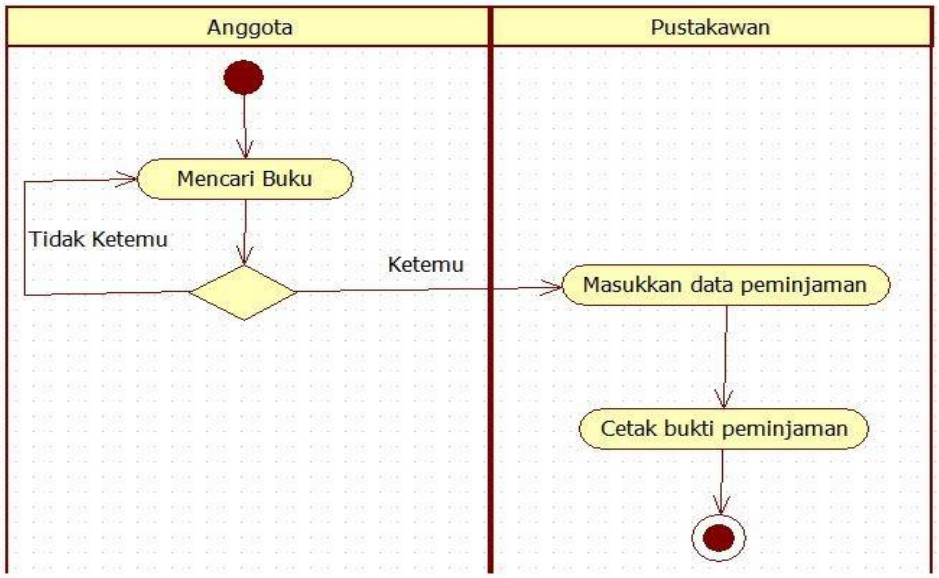
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau *menu* yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.2. Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	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 atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan.
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

Sumber : (Gellysa Urva, 2015)

Contoh Activity Diagram :



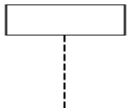
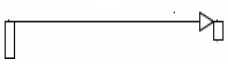
Gambar 2.3. Contoh *Activity Diagram*

Sumber : (Gellysa Urva, 2015)

4. *Sequence Diagram*

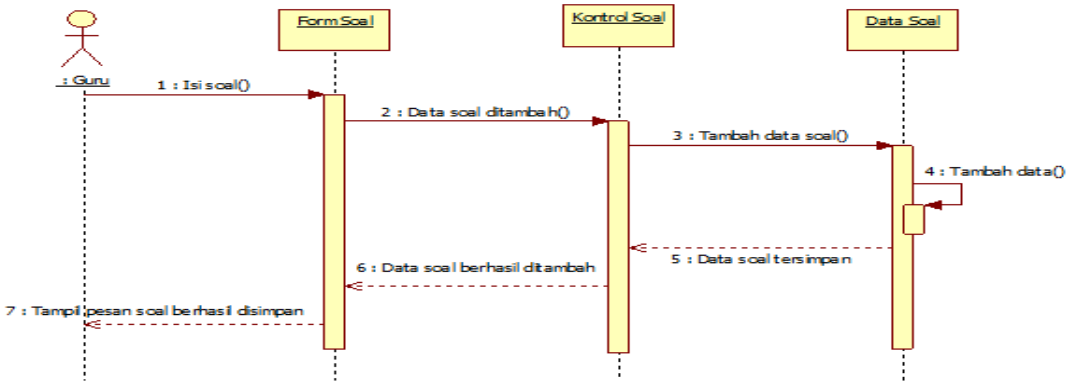
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang 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 : (Gellysa Urva, 2015)

Contoh Squence Diagram :

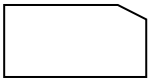


Gambar 2.4. Contoh *Squence Diagram*
Sumber : (Gellysa Urva, 2015)

5. *Class Diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

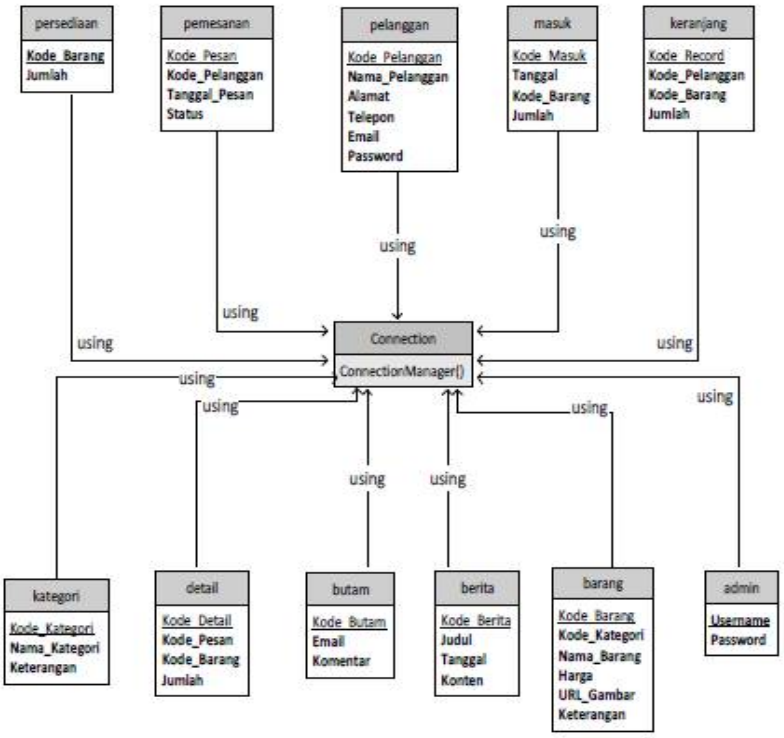
Tabel 2.4. Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2		*dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya
3		*extend*	Menspesifikasikan bahwa use case target memperluas prilaku dari use case sumber pada suatu titik yang diberikan.

Sumber : (Gellysa Urva, 2015)

Contoh *Class* Diagram :



Gambar 2.5. Contoh *Class Diagram*

Sumber : (Gellysa Urva, 2015)

2.10 Pengertian Informasi

Secara Etimologi, kata informasi ini berasal dari kata bahasa Perancis kuno *informacion* (tahun 1387) mengambil istilah dari bahasa Latin yaitu *informationem* yang berarti "konsep, ide atau garis besar". Informasi ini merupakan kata benda dari *informare* yang berarti aktivitas dalam "pengetahuan yang dikomunikasikan".

Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan dan berguna (Yulansari, 2013).

Informasi bisa menjadi fungsi penting dalam membantu mengurangi rasa cemas pada seseorang. Menurut pendapat (Notoatmodjo, 2018) bahwa semakin banyak memiliki informasi dapat memengaruhi atau menambah pengetahuan terhadap seseorang dan dengan pengetahuan tersebut bisa menimbulkan kesadaran yang akhirnya seseorang itu akan berperilaku sesuai dengan pengetahuan yang dimilikinya.

Informasi adalah data yang telah diolah melalui proses tertentu menjadi sesuatu yang menambah pengetahuan atau temuan yang mempunyai arti baru bagi pemakainya (Melina, 2014).

Adapun fungsi-fungsi informasi adalah sebagai berikut:

1. Untuk meningkatkan pengetahuan bagi si pemakai.
2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.

3. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.Informasi yang berkualitas harus akurat, tepat dan relevan.

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dan sebagainya.

2.11 Pengertian Visual Studio

Visual Studio .Net merupakan salah satu *tool development Microsoft* yang dapat digunakan untuk membuat aplikasi di lingkungan kerja berbasis sistem operasi *Windows*. *Visual Studio .NET* menyediakan tools bagi para *developer* untuk membangun aplikasi yang berjalan di *.Net Framework* (Safik, 2015).

Visual Studio (*Beginners All-Purpose Symbolic Instruction Code*) merupakan Bahasa pemrograman *Integrated Development Environment* (IDE), yaitu bahasa pemrograman *visual* yang digunakan untuk membuat program aplikasi atau *software* berbasis sistem operasi *Microsoft Windows*, dengan menggunakan model pemrograman "*Common Object Model* (COM)".

Visual Studio merupakan turunan bahasa pemrograman *STUDIO* yang menawarkan pengembangan perangkat lunak komputer bebasis grafik dengan cepat.

Dengan menggunakan bahasa pemrograman VB, para progammer dapat membangun aplikasi dengan menggunakan komponen-komponen yang di sediakan VB.

Microsoft Visual Studio (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 Studio* merupakan turunan bahasa pemrograman *STUDIO* dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat, Beberapa bahasa skrip seperti *Visual Studio for Applications* (*VBA*) dan *Visual Studio Scripting Edition* (*VBScript*), mirip seperti halnya *Visual Studio*, tetapi cara kerjanya yang berbeda.

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

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

1. Komponen kerja

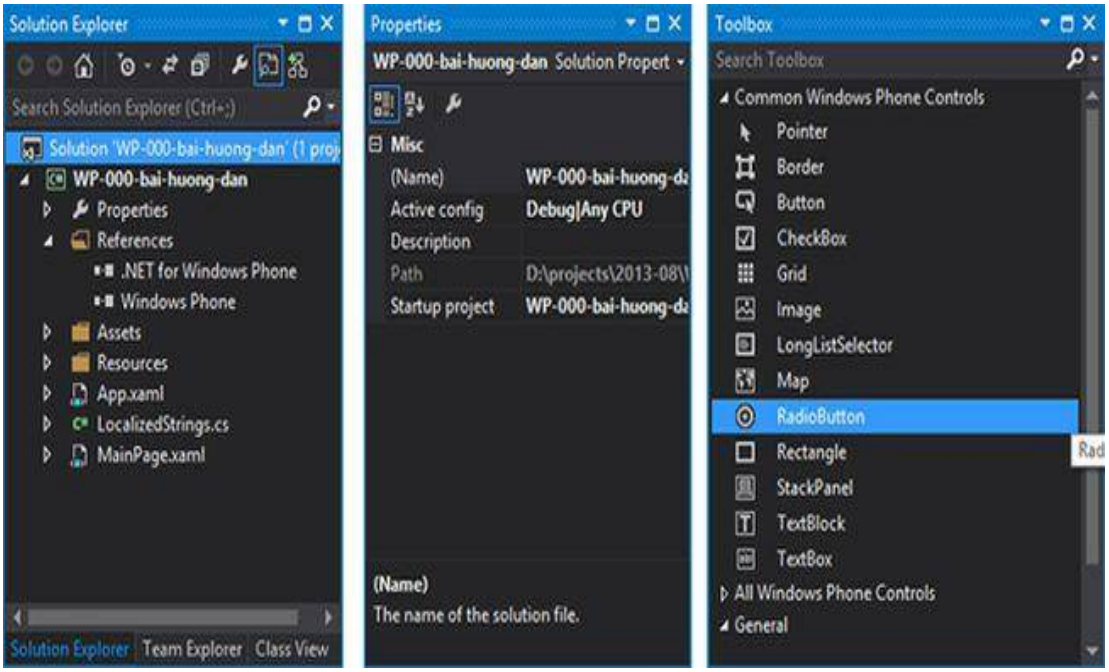
Beberapa komponen kerja program *visual Studio 2015* telah ditampilkan sebagai tampilan standard. Masih banyak lagi komponen yang masih tersembunyi

sehingga memerlukan perintah tertentu untuk menampilkannya. Kita dapat mengatur komponen di dalam program visual Studio 2015 sesuai dengan yang kita butuhkan. Berikut ini adalah beberapa komponen kerja dari visual Studio 2015 adalah :

a. Toolbox

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 Studio 2015* :

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



Gambar 2.6. Tampilan *Toolbox*
Sumber : (Safik, 2015)

Table 2.5. *Toolbox Visual Studio*

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 control waktu dan interval yang diperlukan
Image	Menampilkan gambar pada form dalam format *bit*map, icone, atau metafile
Picturebox	Menampilkan gambar dari sebuah file
Textbox	Membuat teks, dimana teks tersebut dapat diubah oleh pembuat program
Button	Membuat tombol perintah
Combobox	Menambahkan control kotak combo yang merupakan control gabungan antara textbox dan listbox

Sumber : (Safik, 2015)

2.12 Tabel ASCII

ASCII merupakan kepanjangan dari (American Standard Code for Information Interchange), dan pengertian dari ASCII sendiri adalah suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal.

TABEL ASCII

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	sp
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,
010 1101	055	45	2D	-
010 1110	056	46	2E	.
010 1111	057	47	2F	/
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
011 0011	063	51	33	3
011 0100	064	52	34	4
011 0101	065	53	35	5
011 0110	066	54	36	6
011 0111	067	55	37	7
011 1000	070	56	38	8
011 1001	071	57	39	9
011 1010	072	58	3A	:
011 1011	073	59	3B	;
011 1100	074	60	3C	<
011 1101	075	61	3D	=
011 1110	076	62	3E	>
011 1111	077	63	3F	?

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1100	134	92	5C	\
101 1101	135	93	5D	]
101 1110	136	94	5E	^
101 1111	137	95	5F	_

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q
111 0010	162	114	72	r
111 0011	163	115	73	s
111 0100	164	116	74	t
111 0101	165	117	75	u
111 0110	166	118	76	v
111 0111	167	119	77	w
111 1000	170	120	78	x
111 1001	171	121	79	y
111 1010	172	122	7A	z
111 1011	173	123	7B	{
111 1100	174	124	7C	|
111 1101	175	125	7D	}
111 1110	176	126	7E	~

Gambar 2.7. Tabel ASCII

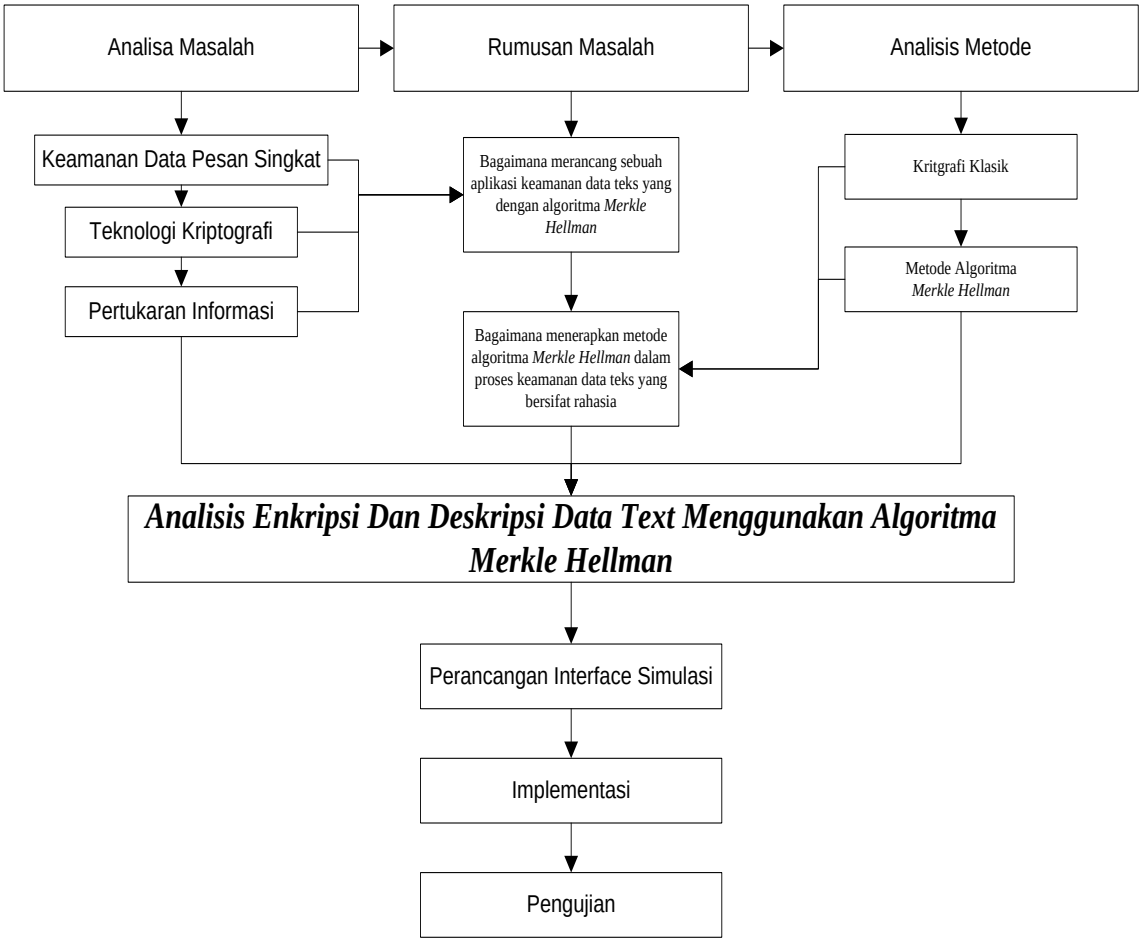
https://www.asciitable.com/

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Tahapan Penelitian

Adapun tahapan penelitian yang dilakukan oleh penulis ini dengan judul Analisis Enkripsi Dan Deskripsi Data Text Menggunakan Algoritma Merkle Hellman adalah sebagai berikut:



Gambar 3.1 Tahapan Penelitian

3.2 Metode Pengumpulan Data

Pengumpulan data adalah pencarian terhadap susuatu kerena ada perhatian dan keinginan terhadap hasil suatu aktivitas. Metode pengumpulan data dalam penulisan ini dibagi menjadi 2, yaitu :

1. Pengamatan (*Observation*)

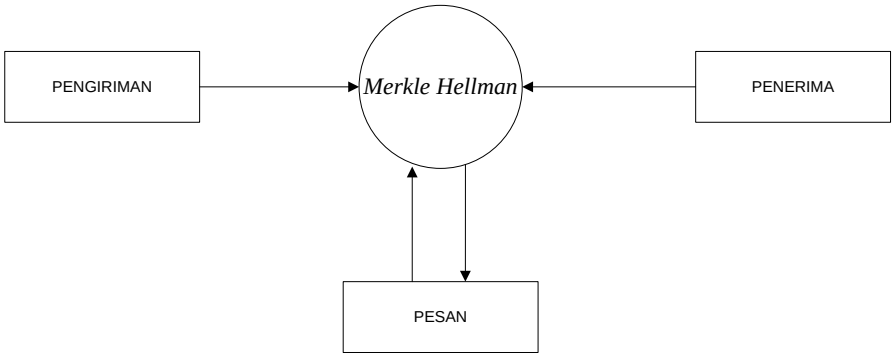
   Penulis melakukan pengamatan langsung pada setiap penggunan aplikasi chating yang sudah ada seperti WA, BBM dan Line untuk mengamati proses keamanan yang sudah dibuat sebelumnya.

2. Penelitian Kepustakaan (*Library Research*)

   Merupakan cara untuk mencari referensi dengan mengumpulkan bahan-bahan pustaka yang dilakukan di perpustakaan kampus, maupun perpustakaan umum, juga melakukan pencarian lewat internet, dengan mengunjungi situs-situs seperti *google Book online* yang dapat membantu pembahasan materi.

3.3 Analisa Permasalahan yang Berjalan

Pertukaran data dalam hal ini pesan rahasia berbentuk teks dengan menggunakan metode tradisional yaitu dengan cara bertukar kata kunci tunggal. Diagram dibawah adalah penggambaran bagaimana pertukaran pesan rahasia menggunakan kunci tunggal terjadi.



Gambar 3.2 Skema Pengiriman Pesan

Pemberitahuan kata kunci dari pengirim ke penerima menggunakan media yang umum digunakan oleh banyak orang.

1. Analisa Kelemahan yang Berjalan

1. Penggunaan kata kunci tunggal berpotensi terjadinya salah pemahaman. Dalam hal ini kemungkinan penerima salah mengartikan kunci yang diberikan oleh pengirim adalah hal yang dapat terjadi.
2. Pemberitahuan atau pertukaran kata kunci yang dikirimkan oleh pengirim ke penerima memiliki potensi dapat diketahui oleh orang lain sehingga pesan rahasia dapat terbongkar.

2. Solusi Pemecahan Masalah

Pemecahan masalah yang penulis lakukan adalah dengan melakukan penerapan metode ini yang didalamnya terdapat Algoritma *Merkle Hellman*. Penggunaan metode ini dapat digunakan sebagai solusi agar pengirim dan penerima tidak lagi harus bertukar kunci tunggal untuk membuka pesan melainkan dapat memiliki kata kunci masing-masing.

Tabel 3.1 Tabel Perencanaan Rancangan

No	Sistem yang Berjalan	Sistem yang Diusulkan	Hasil yang Ingin Dicapai
1.	Penggunaan kunci tunggal yang harus diketahui oleh pengirim dan penerima untuk membuka pesan.	Pengirim dan penerima memiliki kunci masing-masing untuk membuka pesan	Tidak ada lagi kesalahan pemahaman atau salah tafsir kunci tunggal karena pengirim dan penerima memiliki kunci yang dapat ditetapkan masing-masing pihak.
2.	Pertukaran kunci tunggal menggunakan media komunikasi yang rentan untuk dapat diketahui orang lain.	Pengirim dan penerima dapat menentukan sendiri kunci yang ingin digunakan untuk membuka pesan.	Kemungkinan bocornya kunci saat proses pertukaran informasi kunci tunggal dapat dihindari.

3.4 Analisa Kebutuhan Sistem

Analisis kebutuhan sistem merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen – komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukkan yan diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukkan sehingga menghasilkan suatu keluaran yang diinginkan.

1. Analisis Perangkat Keras (Hardware)

Perangkat keras minimum yang digunakan untuk membangun Sistem Informasi Penjualan ini adalah

a. Processor berkecepatan 2.0 Ghz
b. RAM 2 Gb
c. Hardisk minimal 10 Gb untuk menyimpan data
d. LAN Card
e. Keyboard dan Mouse
f. Monitor 14.

2. Analisis Perangkat Lunak (Software)

Untuk mendukung dalam penyimpanan informasi, dibutuhkan suatu fasilitas yang memadai. Yaitu berupa perangkat lunak (software) yang dirancang untuk

memudahkan dalam pembangunan dan menjalankan sisten nantinya. Adapun perangkat lunak yang digunakan adalah sebagai berikut :

a. Microsoft Windows 7 , Windows XP sebagai sistem operasi
b. Mozila Firefox version 3.5 sebagai browser
c. Dreamwever CS 3 Server 10 sebagai Web Server

3.5 Analisa Proses Sistem Yang Berjalan

Visual basic 2010 akan menjadi sarana untuk menciptakan perangkat lunak ini. Pada analisa proses ini penggunaan digunakan sebagai metode yang didalamnya terdapat kombinasi dari algoritma *Merkle Hellman*. Algoritma *Merkle Hellman* digunakan oleh pengirim untuk mengenkripsi pesan yang akan dikirimkan..

Perhitungan secara matematis dilakukan sebagai penggambaran proses yang akan terjadi pada metode ini yang didalamnya terdapat algoritma *Merkle Hellman*. Berikut tahapannya:

Diberikan Private key :

s = (1,2,5,11,32,87,141)

a (Kunci Publik) = 279

p (Kunci Private) = 407

Plaintext (x) : RINALDI

Enkripsi Plaintext : RINALDI

Enkripsi :

R = 1010010

I = 1001001

N = 1001110

A = 1000001

L = 1001100

D = 1000100

I = 1001001

Kunci Publik = 279

Kunci Private = 407

Proses =

R = 82 + 407 Mod 279

I = 73 + 407 Mod 279

N = 78 + 407 Mod 279

A = 65 + 407 Mod 279

L = 76 + 407 Mod 279

D = 68 + 407 Mod 279

I = 73 + 407 Mod 279

Hasil Enkripsi = 210,201,206,193,204,196,201

Proses Deskripsi:

182 - 407 Mod 279 = 82

173 - 407 Mod 279 = 73

178 - 407 Mod 279 = 78

165 - 407 Mod 279 = 65

176 - 407 Mod 279 = 76

168 - 407 Mod 279 = 68

173- 407 Mod 279 = 73

Hasil Deskripsi :

82, 73, 78, 65, 76, 68, 73

Dirubah Menjadi Plaintext Menjadi RINALDI.

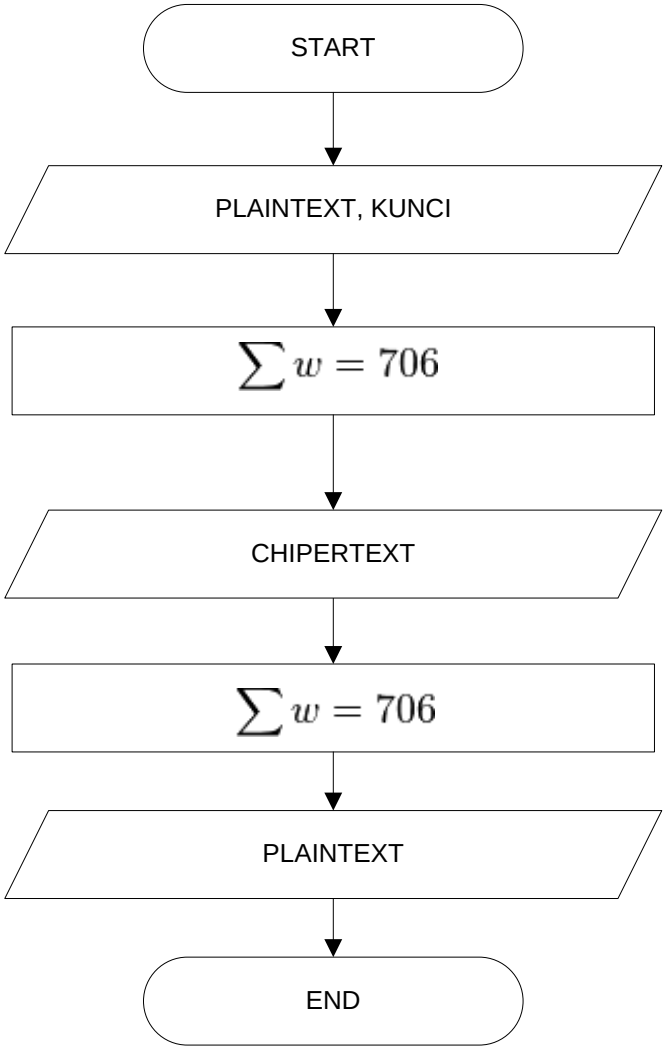
3.6 Flowchart Sistem

Flowchart merupakan langkah awal pembuatan program. Dengan adanya flowchart urutan proses kegiatan menjadi lebih jelas. Bila terdapat penambahan proses maka dapat dilakukan lebih mudah. Setelah flowchart selesai disusun,selanjutnya pemrogram (programmer) menerjemahkannya ke bentuk program dengan bahasa pemrograman.

Flowchart merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol - simbol yang disusun secara sistematis.(Iswandy, 2015)

3.7 Flowchart *Merkle Hellman*

Flowchart Merkle Hellman yang digunakan oleh pengirim untuk mengenkripsi dan mendeskripsi plaintext hingga mendapatkan ciphertext digambarkan sebagai berikut:

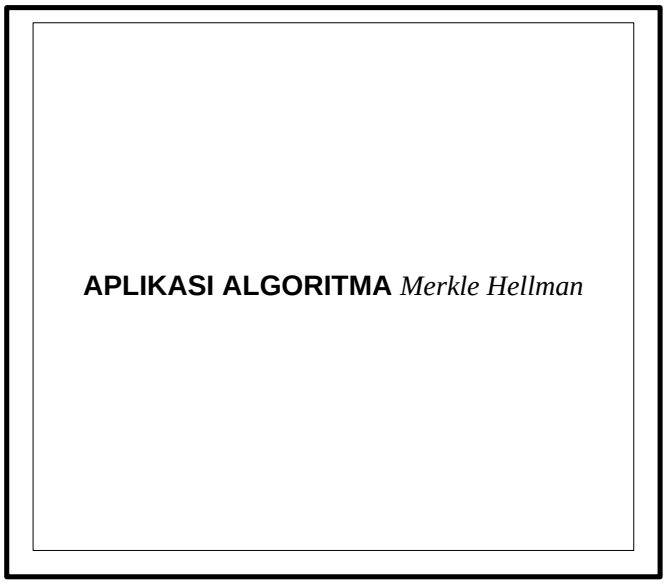


Gambar 3.3 Flowchart Merkle Hellman

3.8 Perancangan Antarmuka

1. Rancangan Halaman Judul

Halaman judul merupakan halaman yang pertama muncul pada saat program dijalankan

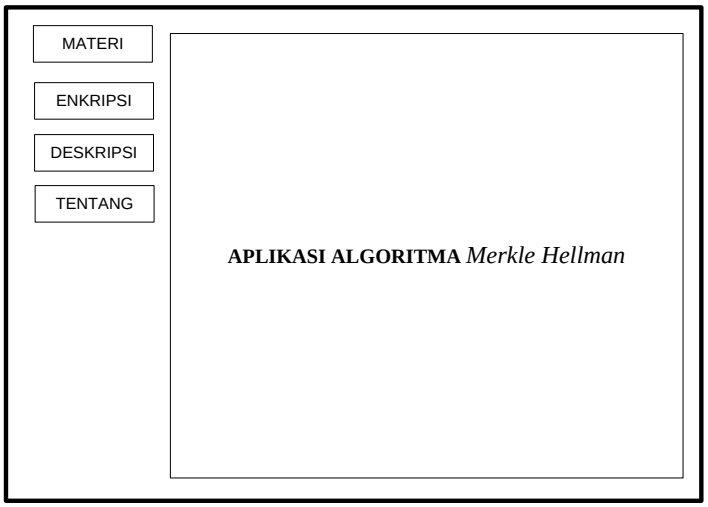


Gambar 3.4 Rancangan Halaman Judul

Pada rancangan di atas akan menampilkan judul yang kemudian akan pindah ke form menu utama dengan menggunakan timer.

2. Rancangan Halaman Menu Utama

Form ini berisi tombol-tombol seperti menu Materi, Enkripsi, Deskipsi, tentang, dan Keluar.



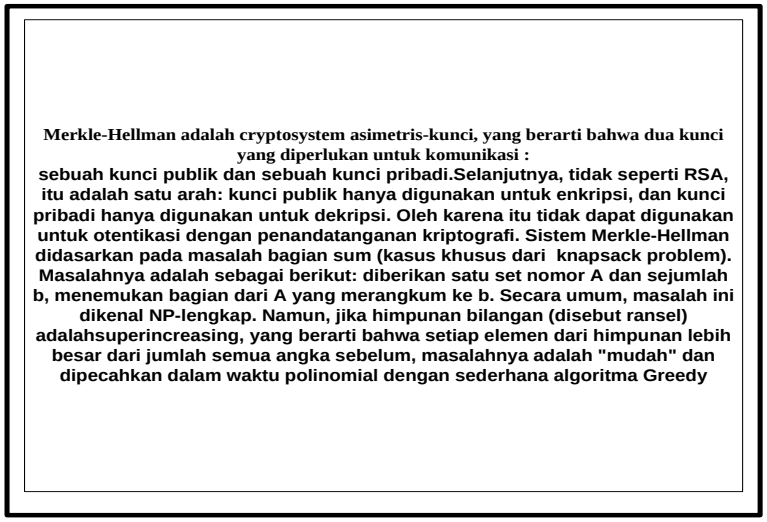
Gambar 3.5 Rancangan Halaman Menu Utama

Pada tampilan di atas terdapat 5 tombol yaitu Materi, Enkripsi, Deskripsi, Tabel Affine, Tentang dan keluar.

a. Tombol Materi berfungsi untuk menghubungkan pengguna ke form materi.
b. Tombol Enkripsi berfungsi untuk menghubungkan pengguna ke form Enkripsi.
c. Tombol Deskripsi berfungsi untuk menampilkan form Deskripsi.
d. Tombol Tentang berfungsi untuk menghubungkan pengguna ke form tentang.
e. Tombol Keluar berfungsi untuk keluar dari program.

3. Rancangan Halaman Materi

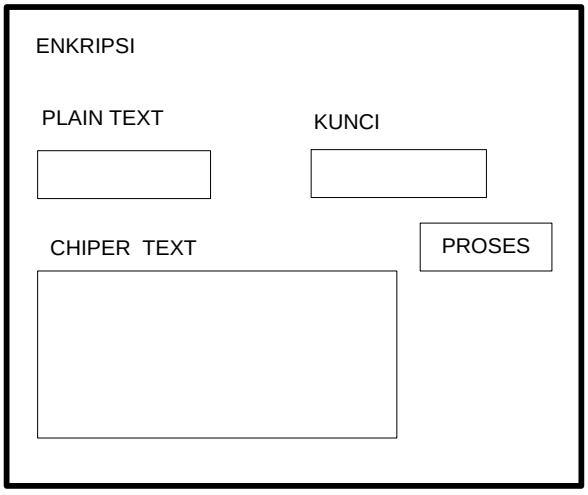
Form ini digunakan untuk menjelaskan cara kerja penyandian, dimulai dari plaintext kemudian kunci yang dikonversikan dalam bentuk angka. Setelah itu dilakukan proses penjumlahan dan jika hasil penjumlahan maka akan dikurangi 6 lalu hasilnya akan dikembalikan lagi ke dalam bentuk huruf.



Gambar 3.6 Rancangan Halaman Materi

4. Rancangan Halaman Enkripsi

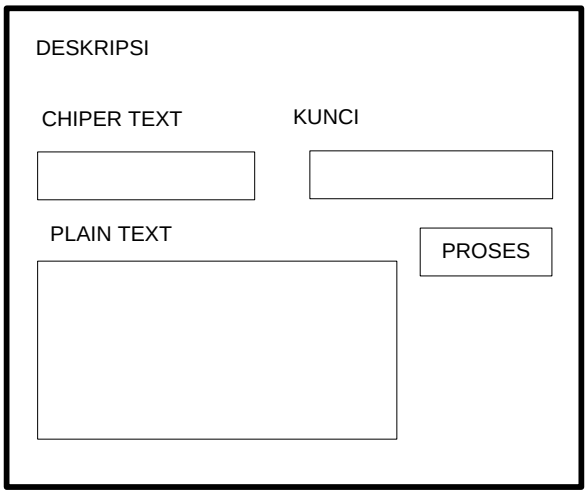
Berisi penjelasan mengenai Enkripsi. Pengguna memasukkan tulisan asli atau *plaintext* ke dalam tombol masukan *plaintext* kemudian dimasukkan juga kunci. Setelah itu, ditekan tombol Proses Enkripsi yang kemudian akan menampilkan ciphertext atau tulisan yang telah disandikan.



Gambar 3.7 Rancangan Halaman Enkripsi

5. Rancangan Halaman Deskripsi

Berisi penjelasan mengenai Enkripsi. Pengguna memasukkan tulisan asli atau *plaintext* ke dalam tombol masukan *plaintext* kemudian dimasukkan juga kunci. Setelah itu, ditekan tombol Proses Enkripsi yang kemudian akan menampilkan ciphertext atau tulisan yang telah disandikan.



Gambar 3.8 Rancangan Halaman Deskripsi

Pada gambar di atas terdapat kotak input Deskripsi berfungsi untuk memasukkan tulisan yang telah disandikan. Kemudian terdapat tombol Proses Deskripsi untuk mengembalikan ke tulisan asli jika kunci yang dimasukkan sama dengan kunci pada saat penggunaan plaintext.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisa Kebutuhan Sistem

Analisis kebutuhan sistem merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen – komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukkan yan diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukkan sehingga menghasilkan suatu keluaran yang diinginkan.

1. Analisis Perangkat Keras (Hardware)

Perangkat keras minimum yang digunakan untuk membangun Sistem Informasi Penjualan ini adalah

a. Processor berkecepatan 2.0 Ghz
b. RAM 2 Gb
c. Hardisk minimal 10 Gb untuk menyimpan data
d. LAN Card
e. Keyboard dan Mouse
f. Monitor 14.

2. Analisis Perangkat Lunak (Software)

Untuk mendukung dalam penyimpanan informasi, dibutuhkan suatu fasilitas yang memadai. Yaitu berupa perangkat lunak (software) yang dirancang untuk memudahkan dalam pembangunan dan menjalankan sisten nantinya. Adapun perangkat lunak yang digunakan adalah sebagai berikut :

a. Microsoft Windows 7 , Windows XP sebagai sistem operasi
b. Microsoft Visual Studio 2010

4.2 Implementasi Sistem

Pengujian dilakukan dengan memasukkan karakter atau huruf dari file berformat txt selanjutnya diproses oleh aplikasi apakah aplikasi tersebut dapat memberikan hasil yang sesuai. Proses yang akan dilakukan pengujian dalam aplikasi ini adalah simulasi pengiriman pesan dengan menggunakan metode Algoritma Merkle Hellman antara pengirim kepada penerima dengan kunci yang dimiliki masing-masing pihak tanpa perlu bertukar kunci tunggal hingga pada akhirnya pesan asli yang dikirimkan oleh pengirim dapat dibaca oleh penerima .

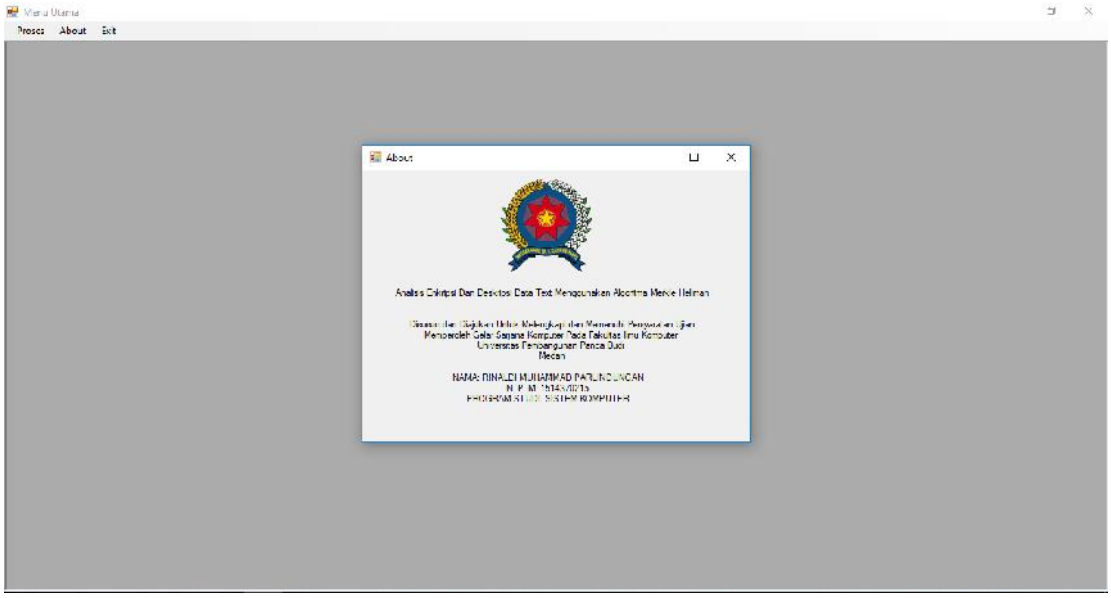
1. Tampilan Awal/ Home

Tampilan pada gambar 4.1 merupakan tampilan awal ketika aplikasi dijalankan. Pada form ini pengguna dapat memilih untuk membuka beberapa form lainnya seperti tombol tentang yang akan mengarahkan pengguna menuju form yang menjelaskan profil aplikasi ini, tombol *read me!* yang akan mengarahkan pengguna ke form yang menjelaskan tata cara penggunaan dari aplikasi ini.

Gambar 4.1 Tampilan Awal/ Home

2. Tampilan Halaman Tentang

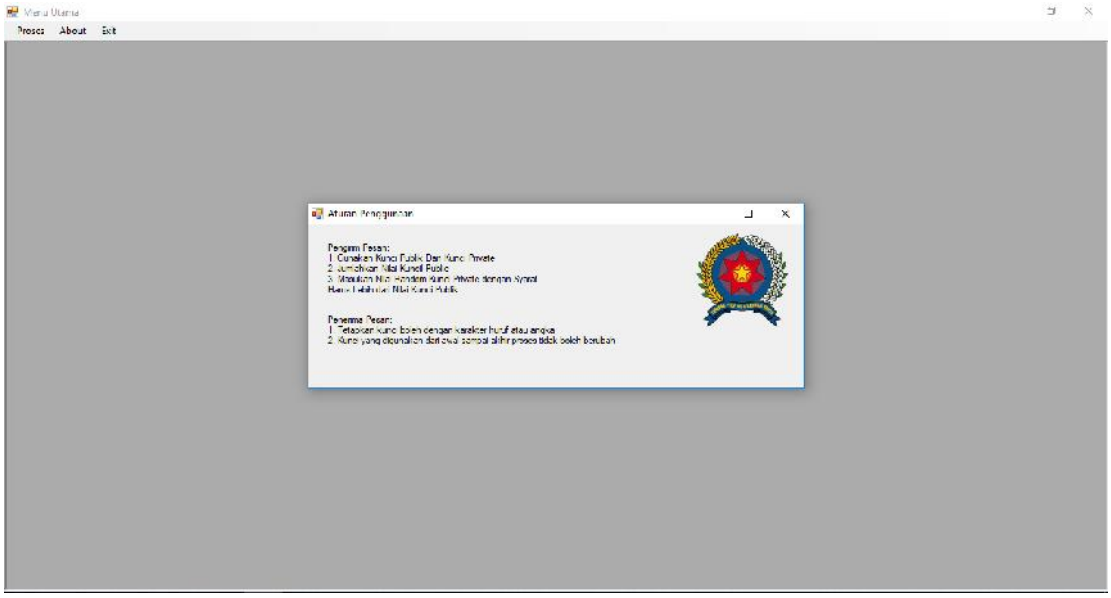
Tampilan berikut ini menampilakan halaman atau form yang berisi tentang profil dari aplikasi ini. Didalamnya terdapat judul dari aplikasi beserta maksud dari pembuatannya beserta nama dan nomor pokok mahasiswa penulis.



Gambar 4.2 Tampilan Halaman Tentang

3. Tampilan Aturan Penggunaan Aplikasi

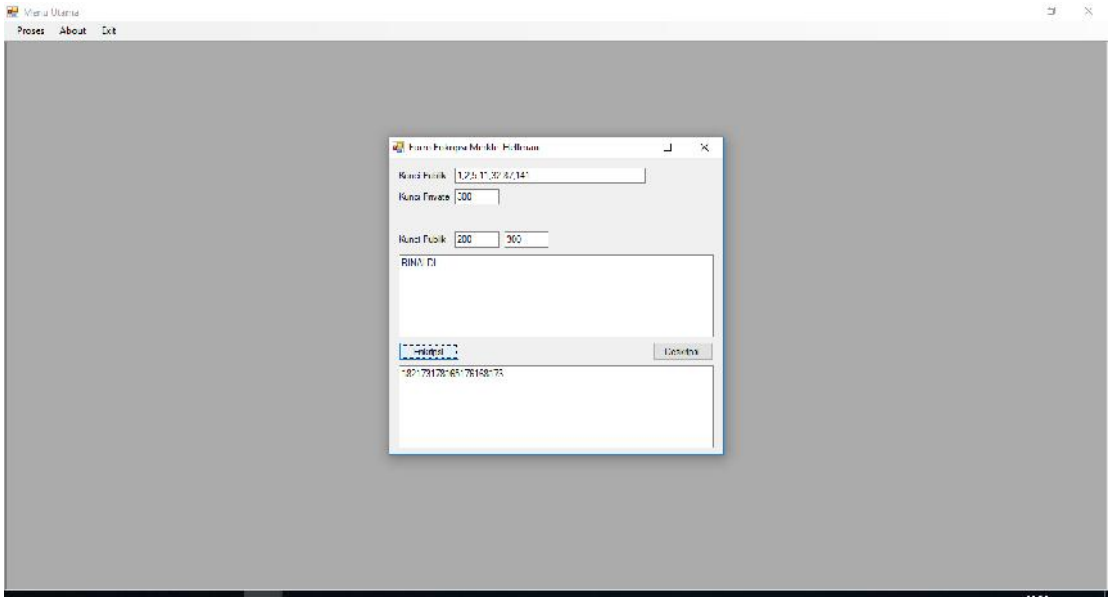
Tampilan aturan penggunaan aplikasi merupakan tampilan halaman atau form yang berisi tentang tata cara penggunaan aplikasi yang dijalankan. Pada halaman tersebut dijelaskan apa-apa saja yang menjadi kewajiban bagi pengirim dan penerima saat menjalankan simulasi Algoritma Merkle Hellman .



Gambar 4.3 Tampilan Aturan Penggunaan Aplikasi

4. Tampilan Halaman Pengirim Pesan

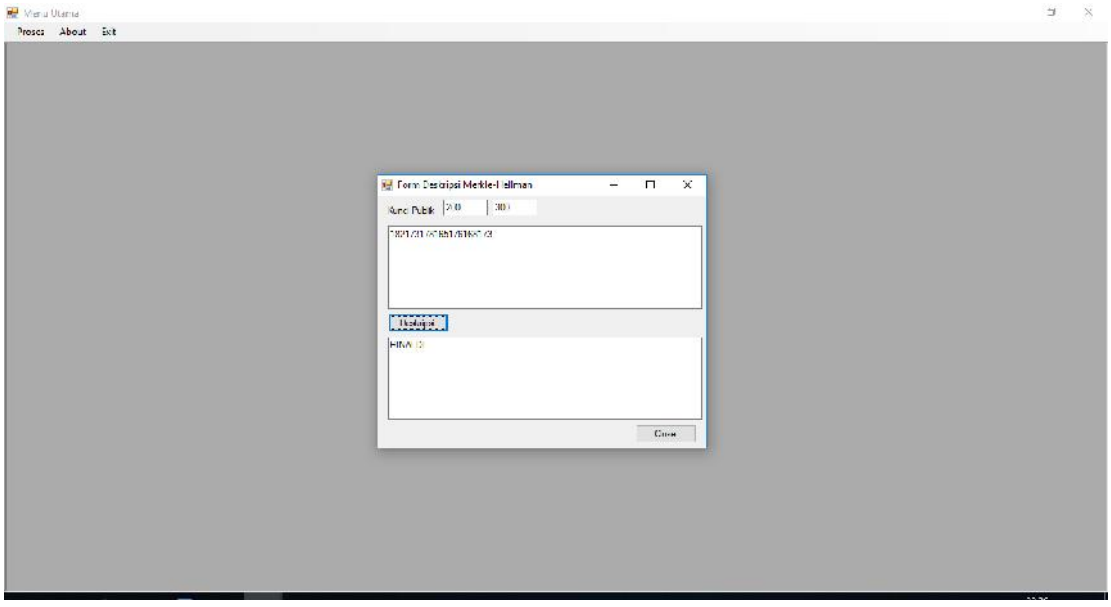
Tampilan berikut merupakan tampilan pengiriman pesan pada aplikasi ini. Algoritma Merkle Hellman merupakan protokol yang menjamin tidak adanya pertukaran kunci antara pihak-pihak yang melakukan enkripsi dan dekripsi. Kedua belah pihak menggunakan kunci mereka masing-masing untuk mengenkripsi pesan dan kemudian untuk mendekripsi pesan tanpa perlu mengetahui kunci yang lainnya



Gambar 4.4 Tampilan Halaman Pengirim Pesan

5. Tampilan Halaman Penerima Pesan

Tampilan berikut merupakan tampilan penerima pesan pada aplikasi ini.



Gambar 4.5 Tampilan Halaman Penerima Pesan

4.3 Pengujian Sistem

Perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, perancangan, dan pengkodean. Pengujian yang digunakan untuk menguji sistem ini adalah metode pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak.

1. Rencana Pengujian

Pengujian fungsi Penerapan Matrix Persegi Pancajang Dalam Pengembangan Algoritma Hill Chiper dilakukan dengan menggunakan metode Black Box. Pengujian dilakukan pada fungsi-fungsi sistem untuk menentukan apakah fungsi tersebut telah berjalan sesuai dengan yang diharapkan.

1) Bangkitkan Kunci

Tabel 4.1 . Rencana Pengujian Tombol Cari

Menu yang diuji	**Detail pengujian**	**Kesimpulan**
Kunci Public	Melakukan penjumlahan kunci public.	*Diterima*

2) Proses Enkripsi

Tabel 4.2. Rencana Pengujian Pengguna (*User*)

Menu yang diuji	**Detai pengujian**	**Jenis uji**
Enkripsi	Melakukan proses enkripsi	*Diterima*
Kirim	Proses pengiriman file enkripsi	*Diterima*
Kunci Public	Melakukan proses Kunci Public	*Diterima*
Kunci Private	Melakukan proses Kunci Private	*Diterima*

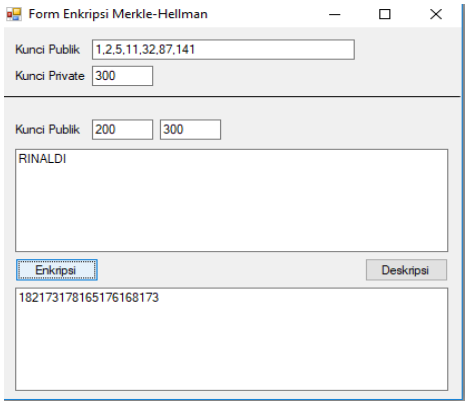
3) Proses Deskripsi

Tabel 4.3. Rencana Pengujian Pengguna (*User*)

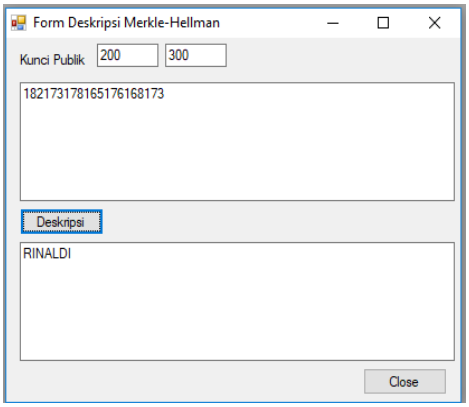
Menu yang diuji	**Detai pengujian**	**Jenis uji**
Dekripsi	Melakukan proses deskripsi atau pengembalian pesan asli	*Diterima*
Close	Menutup semua program	*Diterima*

2. Pengujian Proses

Pengujian proses yang telah disusun, maka dapat dilakukan pengujian sebagai berikut :



Gambar 4.6. Proses Pengujian Enkripsi (*User*)



Gambar 4.7. Proses Pengujian Deskripsi (*User*)

3. Kesimpulan Dan Hasil Pengujian Sistem

Hasil pengujian dari pengujian alpha telah selesai, menunjukkan bahwa sistem sudah memenuhi syarat fungsional. Secara fungsional sistem yang sudah dibangun sudah dapat menghasilkan keluaran sesuai yang diharapkan.

Tabel 4.4. Kesimpulan Pengujian Alpha

Nama fungsi	**Hasil**
File	Fungsi berjalan dengan baik
Enrkipsi	Fungsi berjalan dengan baik
Deskripsi	Fungsi berjalan dengan baik
Kunci Private	Fungsi berjalan dengan baik
Kunci Public	Fungsi berjalan dengan baik

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan dalam Analisis Enkripsi Dan Deskripsi Data Text Menggunakan Algoritma Merkle Hellman, maka dapat diambil kesimpulan sebagai berikut :

1. Dengan adanya aplikasi enkripsi dan deskripsi dari metode Merkell Hellman, penulis dapat merancang dan memahami metode dalam proses pengaman sebuah data text.
2. Pada proses enkripsi dengan metode Merkell Hellman, kunci public harus di jumlahkan lalu untuk kunci private harus lebih besar nilainya dari kunci public.
3. Kunci yang dipakai pada proses enkripsi dan deskripsi hanya 8 kunci, dan menggunakan kunci *public* ataupun *private*.

5.2 Saran

Adapun saran-saran yang dapat dilakukan penelitian ataupun pengembangan selanjutnya adalah sebagai berikut:

1. Perangkat lunak ini dapat dikembangkan dengan menggunakan andriod pada aplikasi seperti WhatApp, BBM, ataupun akun pada sosial media dan LINE.
2. Perangkat lunak ini dapat dikembangkan menggunakan Artificial Intelligence (AI).

DAFTAR PUSTAKA

Andrian, Yudhi, and Purwa Hasan Putra. "Analisis Penambahan Momentum Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network." Seminar Nasional Informatika (SNIf). Vol. 1. No. 1. 2017.

Azmi, Fadhillah, And Winda Erika. "Analisis Keamanan Data Pada Block Cipher Algoritma Kriptografi Rsa." Cess (Journal Of Computer Engineering, System And Science) 2.1: 27-29.

Bishop, D. V. (2014). Uncommon Understanding (Classic Edition): Development and disorders of language comprehension in children. Psychology Press.

Erika, Winda, Heni Rachmawati, and Ibnu Surya. "Enkripsi Teks Surat Elektronik (E-Mail) Berbasis Algoritma Rivest Shamir Adleman (RSA)." Jurnal Aksara Komputer Terapan 1.2 (2012).

Hafni, Layla, And Rismawati Rismawati. "Analisis Faktor-Faktor Internal Yang Mempengaruhi Nilai Perusahaan Pada Perusahaan Manufaktur Yang Terdaftar Di Bei 2011-2015." Bilancia: Jurnal Ilmiah Akuntansi 1.3 (2017): 371-382.

Hamdi, Muhammad Nurul, Evi Nurjanah, And Latifah Safitri Handayani. "Community Development Based Onibnu Khaldun Thought, Sebuah Interpretasi Program Pemberdayaan Umkm Di Bank Zakat El-Zawa." El Muhasaba: Jurnal Akuntansi (E-Journal) 5.2 (2014): 158-180.

Haviluddin, H., & Jawahir, A. (2015). Comparing of ARIMA and RBFNN for short-term forecasting. International Journal of Advances in Intelligent Informatics, 1(1), 15-22.

Indra Permana, Aminuddin "Sistem Pakar Mendeteksi Hama Dan Penyakit Tanaman Kelapa Sawit Pada Pt. Moeis Kebun Sipare-Pare Kabupaten Batubara." (2013).

Juliarta, F. (2015). Penerapan Pembelajaran Kontekstual (Ctl) Pada Materi Aritmatika Sosial Untuk Meningkatkan Hasil Belajar Siswa Kelas Vii-3 Smp Negeri 17 Medan Ta 2014/2015 (Doctoral dissertation, UNIMED).

JURTEKSI ROYAL Edisi2.

Munir, R. (2011). Algoritma enkripsi citra dengan pseudo One-Time Pad yang menggunakan sistem chaos. konferensi nasional informatika, 12-16.

Muttaqin, Muhammad. "Analisa Pemanfaatan Sistem Informasi E-Office Pada Universitas Pembangunan Panca Budi Medan Dengan Menggunakan Metode Utaut." Jurnal Teknik Dan Informatika 5.1 (2018): 40-43.

Muttaqin, Muhammad. "Portal Academic Portal Innovation Based On Website In The Era Of Digital 4.0 Technology Now."

Nawaz, A., Aminuddin, A., Kado, T., Takikawa, A., Yamamoto, S., Tsuneyama, K.

Pabokory, F. N., & Astuti, I. & Kridalaksana, AH (2015). Implementasi kriptografi pengamanan data pada pesan teks, isi file dokumen, dan file dokumen menggunakan algoritma advanced encryption standard. Jurnal Informatika Mulawarman.

Permana, A. I., and Z. Tulus. "Combination of One Time Pad Cryptography Algorithm with Generate Random Keys and Vigenere Cipher with EM2B KEY." (2020).

Permana, Aminuddin Indra. "Kombinasi Algoritma Kriptografi One Time Pad dengan Generate Random Keys dan Vigenere Cipher dengan Kunci EM2B." (2019).

Perwitasari, I. D. (2018). Teknik Marker Based Tracking Augmented Reality untuk Visualisasi Anatomi Organ Tubuh Manusia Berbasis Android. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 8-18.

Puspita, Khairani, and Purwa Hasan Putra. "Penerapan Metode Simple Additive Weighting (SAW) Dalam Menentukan Pendirian Lokasi Gramedia Di Sumatera Utara." Seminar Nasional Teknologi Informasi Dan Multimedia, ISSN. 2015.

Rhee, H. W., Zou, P., Udeshi, N. D., Martell, J. D., Mootha, V. K., Carr, S. A., & Ting, A. Y. (2013). Proteomic mapping of mitochondria in living cells via spatially restricted enzymatic tagging. Science, 339(6125), 1328-1331.

Rizal, Chairul. "Pengaruh Varietas dan Pupuk Petroganik Terhadap Pertumbuhan, Produksi dan Viabilitas Benih Jagung (Zea mays L.)." ETD Unsyiah (2013).

Santosa, R. A. (2014). Penerapan Algoritma AES Guna Keamanan Transmisi Data pada Aplikasi Client-Server: studi kasus sistem jejaring klaster (Doctoral dissertation, Program Studi Teknik Informatika FTI-UKSW).

Sari, C. A., & Rachmawanto, E. H. (2014). Gabungan Algoritma Vernam Chiper Dan End Of File Untuk Keamanan Data. Techno. Com, 13(3), 150-157.

Syahputra, Rizki, And Hafni Hafni. "Analisis Kinerja Jaringan Switching Clos Tanpa Buffer." Journal Of Science And Social Research 1.2 (2018): 109-115.

Syahrizal, M., Murdani, M., Nasution, S. D., Mesran, M., Rahim, R., & Siahaan, A. P. U. (2017). Modified Playfair Cipher Using Random Key Linear Congruent Method. J. Online Jar. COT POLIPD, 10(2), 45-49.

& Takatsu, K. (2017). CD206+ M2-like macrophages regulate systemic glucose metabolism by inhibiting proliferation of adipocyte progenitors. Nature communications, 8(1), 286.

Urva, G., & Siregar, H. F. (2015). Pemodelan UML E-Marketing Minyak Goreng.

Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." Jurnal Abdi Ilmu 10.2 (2018): 1899-1902.