IMPLEMENTASI ALGORITMA BEAUFORT CIPHER DALAM PROSES ENKRIPSI DAN DESKRIPSI UNTUK PENGAMANAN PESAN

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	:	**AYU SYAHFITRI**
NPM	:	**1514370378**
PROGRAM STUDI	:	**SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

ABSTRAK

AYU SYAHFITRI

Implementasi Algoritma Beaufort Cipher Dalam Proses Enkripsi Dan Deskripsi Untuk Pengamanan Pesan

2020

Keamanan informasi harus dijaga dengan baik. Hal ini terjadi karena sudah banyaknya pencurian data yang dilakukan oleh pihak yang tidak bertanggung jawab. Pengiriman informasi melalui jaringan bebas seperti *internet* akan bersifat rentan dan dapat sewaktu-waktu dibobol oleh orang lain. Dengan itu diperlukan suatu teknik kriptografi yang dapat menjaga keamanan data tersebut. Data tidak dapat disembunyikan, tetapi data dapat di enkripsi sehingga apabila data tersebut jatuh ke tangan orang lain, informasi yang terkandung di dalamnya tidak akan diketahui. Algoritma *Beaufort Cipher* sangat baik dalam melakukan proses enkripsi. Algoritma ini bekerja dengan sangat cepat dengan cara menggeser posisi karakter sebesar kunci yang telah ditentukan. Hasil *ciphertext* merupakan pertambahan pergeseran dan hasil *plaintext* adalah penguragan pergeseran kunci. Dengan menerapkan algoritma *Beaufort Cipher*, keamanan informasi dapat terjamin.

Kata Kunci: *Algoritma, Ekripsi, Enkripsi, Kriptografi, Beaufort*

DAFTAR ISI

BAB I

PENDAHULUAN

1.1 Latar Belakang

Matematika merupakan ilmu pengetahuan dasar yang sangat berpengaruh dalam perkembangan ilmu pengetahuan lainnya, termasuk perkembangan teknologi informasi. Perkembangan teknologi informasi memiliki pengaruh yang sangat signifikan bagi aspek kehidupan. Tidak terkecuali pada aspek komunikasi dan pengiriman pesan maupun pengiriman isi pesan. Masalah keamanan dan kerahasiaan isi pesan merupakan hal yang penting. Isi pesan yang bersifat rahasia tersebut perlu dibuatkan sistem penyimpanan dan pengirimannya agar tidak terbaca atau diubah oleh orang-orang yang tidak bertanggung jawab, baik itu saat isi pesan tersebut tersimpan di dalam komputer maupun saat isi pesan dikirim melalui *e-mail*.

Untuk menyimpan isi pesan agar benar-benar aman, tentunya dilakukan sistem pengamanan yang baik, dan bebas dari jangkauan orang-orang yang tidak berhak, baik bebas dari jangkauan secara fisik maupun secara sistem. Apalagi jika isi pesan berada dalam suatu jaringan komputer yang terhubung dengan jaringan internet, tentu saja isi *file* tersebut tidak boleh diketahui dan diubah oleh orang yang tidak berhak.

Menurut *Request for Comments* (RFC), kriptografi merupakan ilmu matematika yang berhubungan dengan transformasi data untuk membuat artinya tidak dapat dipahami (untuk menyembunyikan maknanya) dan mencegahnya dari

perubahan tanpa izin, atau mencegahnya dari penggunaan yang tidak sah. Jika transformasinya dapat dikembalikan, kriptografi juga bisa diartikan sebagai proses mengubah kembali data yang terenkripsi menjadi bentuk yang dapat dipahami. Artinya, kriptografi dapat diartikan sebagai proses untuk melindungi data dalam arti yang luas (Oppliger, 2005). Dalam kriptografi terdapat dua konsep utama yaitu enkripsi dan dekripsi. Enkripsi merupakan proses penyandian pesan asli atau *plaintext* menjadi *ciphertext* (teks tersandi). Sedangkan dekripsi adalah proses penyandian kembali *ciphertext* menjadi *plaintext.*

Pada penelitian sebelumnya yang berjudul "Pembuatan Aplikasi Kriptografi *File* Menggunakan Algoritma *Vigenere Cipher*" dijelaskan hasil penelitian menunjukkan bahwa isi *file* dapat terenkripsi menggunakan algoritma *Vigenere Cipher* yang bersifat sistem substitusi multi-alphabet, yaitu dengan sistem sandi *Caesar* tetapi dengan pergeseran alphabet yang berlainan disesuaikan dengan kata kunci. Yang dimaksud sistem sandi substitusi merupakan penyandian dengan cara menggantikan huruf-huruf/teks aslinya dengan huruf-huruf sandi. Cara kerjanya yaitu isi *file* terlebih dahulu akan dienkripsi kemudian isi *file* akan dikirimkan melalui *e-mail,* ketika penerima sudah menerima isi *file* maka akan dideskripsikan sesuai dengan kunci, maka pihak ketiga atau orang yang tidak berhak tidak dapat melihat isi *file* tersebut (Aristia, 2019).

Berdasarkan informasi yang telah dipaparkan, penulis membuat sebuah penerapan enkripsi dan dekripsi dengan menggunakan metode *Beaufort Cipher* dengan mengenkripsi dan mendekripsi isi pesan. Cara kerja enkripsi dan dekripsi ini akan dibuat secara mudah dan efektif. Implementasi ini menerapkan sistem

enkripsi dan dekripsi menggunakan algoritma *Beaufort Cipher* simetris dalam pengamanan isi pesan yang berjudul "**Implementasi Algoritma *Beaufort Cipher* Dalam Proses Enkripsi Dan Deskripsi Untuk Pengamanan Pesan**".

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, maka rumusan masalah dalam penelitian ini sebagai berikut:

1. Bagaimana membuat program aplikasi yang dapat mengamankan pesan?
2. Bagaimana merancang algoritma Beufort cipher dalam mengamankan pesan yang akan dikirim?
3. Bagimana menentukan kunci pada algoritma Beaufort cipher dalam mendukung proses enkripsi dan dekripsi?

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah dipaparkan, maka batasan masalah dalam penelitian ini sebagai berikut:

1. Pesan yang digunakan adalah berbentuk teks.
2. Panjang karakter yang dapat di enkripsi adalah 1000 karakter.
3. Panjang kunci maksimal yang digunakan adalah sepanjang plaintext.
4. Program yang digunakan dalam perancangan aplikasi ini adalah *Visual Basic.Net 2010*.

1.4 Tujuan Penulisan

Tujuan yang ingin dicapai penulis dalam perancangan aplikasi penerapan algoritma *Beaufort* ini adalah:

1 Untuk membuat program aplikasi yang dapat mengamankan pesan.

2 Untuk merancang algoritma Beufort cipher dalam mengamankan pesan yang akan dikirim.

3 Untuk menentukan kunci pada algoritma Beaufort cipher dalam mendukung proses enkripsi dan dekripsi.

1.5 Manfaat Penelitian

Perancangan aplikasi penerapan algoritma *Beaufort* ini bermanfaat bagi masyarakat luas antara lain:

1. Dengan menggunakan aplikasi ini seseorang dapat mengamankan suatu informasi tanpa takut diketahui oleh orang lain.
2. Memberikan kerahasian pada data pribadi pada saat dikirimkan
3. Proses pertukaran data atau informasi menjadi aman.

BAB II

LANDASAN TEORI

2.1 Kebocoran Informasi

Kebocoran Informasi adalah kategori kerentanan perangkat lunak di mana informasi diungkapkan secara tidak sengaja kepada pengguna akhir, yang berpotensi membantu penyerang dalam upaya mereka untuk melanggar keamanan aplikasi. Kriteria kunci untuk Kebocoran Informasi adalah bahwa paparan tidak disengaja dan berguna bagi penyerang (Sun, Zhang, Xiong, & Zhu, 2014).

Kebocoran Informasi berbeda dengan kegagalan melindungi informasi sensitif saat istirahat dan dalam perjalanan. Ini adalah masalah yang sah dan melibatkan pemaparan data sensitif apa pun yang disimpan dan diproses oleh aplikasi. Kebocoran Informasi melibatkan pemaparan informasi yang akan memfasilitasi serangan terhadap aplikasi atau infrastruktur lainnya, seperti wawasan tentang desain aplikasi, penyebaran, atau detail organisasi. Contoh-contoh dari informasi "sensitif aplikasi" yang mungkin tidak disengaja termasuk:

1. Pengidentifikasi Akun: Cara membedakan akun yang ada disediakan, memfasilitasi serangan brute-force pada kontrol akses.
2. Alamat Email: Paparan alamat email internal dapat digunakan dalam serangan Rekayasa Sosial seperti Phishing serta serangan pada kontrol akses.
3. Struktur Sistem File: Eksposur struktur sistem internal melalui referensi jalur terbuka dapat memfasilitasi serangan seperti Path Traversal.

4. Konfigurasi Aplikasi: Eksposur informasi konfigurasi dapat menginformasikan penyerang kesalahan konfigurasi dan membantu memfokuskan strategi serangan. Contoh akan menjadi referensi yang jelas untuk Debug Switches yang dapat memberdayakan penyerang untuk mengaktifkan debugging.
5. Struktur Basis Data: Wawasan tentang skema yang digunakan oleh aplikasi dapat membantu membuat string SQL Injection.
6. Token Sesi dan Otentikasi: Eksposur nilai-nilai ini dapat digunakan untuk melancarkan serangan Pembajakan Sesi.

2.1.1 Terjadinya Kebocoran Informasi

Karena informasi dapat disampaikan dalam berbagai bentuk, ada banyak cara yang memungkinkan informasi yang menarik bagi penyerang(Kelsey, 2002):

1. Pesan Kesalahan Teknis: Manifestasi umum dari kerentanan ini adalah tampilan jejak tumpukan, pesan kesalahan basis data, dan respons lainnya dengan detail teknis yang tidak berarti bagi pengguna akhir.
2. Spanduk: Eksposur versi perangkat lunak Sistem Operasi, Web-Server, database, atau komponen aplikasi lainnya.
3. Penghitungan Akun: Menyediakan mekanisme untuk mengungkapkan nama akun yang ada.
4. Sumber Halaman Web: Dimasukkannya informasi sensitif aplikasi dalam komentar yang dikirimkan kepada pengguna akhir.

5. File Pendukung: Informasi yang terpapar dalam sumber, komentar, atau data konfigurasi ditemukan dalam *Javascript*, *CSS*, atau file tambahan lainnya.
6. Pesan Diagnostik / Debug: Pemaparan informasi melalui data debug dalam Respons.
7. Waktu Acara: Skenario di mana dimungkinkan untuk membedakan wawasan ke dalam operasi internal dengan berapa lama untuk menyelesaikannya. Serangan *Blind Injection* terkadang mengeksploitasi *Event Timing*.
8. Cookie: Paparan token Sesi, token otentikasi, dan informasi status lainnya yang terkandung dalam cookie.
9. Caching: Kegagalan melindungi informasi dari di-*cache* oleh *Web Browser*, berpotensi mengekspos informasi tersebut kepada penyerang dengan akses mesin lokal.

2.1.2 Kebocoran Informasi Adalah Kerentanan

Terkadang ada fokus yang sangat besar untuk melindungi data sensitif yang sedang diproses sehingga kami mengabaikan informasi tentang desain atau penyebaran aplikasi yang harus dianggap “sensitif aplikasi” dan dilindungi. Kegagalan untuk melindungi informasi sensitif aplikasi, meskipun tidak dapat dieksploitasi, dapat memberikan informasi berharga kepada penyerang dan dengan demikian memfasilitasi serangan lain.

2.1.3 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 keamanan yaitu (Pabokory, Astuti, & Kridalaksana, 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.
4. Terdapat lima langkah keamanan komputer yang baik untuk diperhitungkan yaitu; aset, analisis resiko, perlindungan, alat dan prioritas.

2.2 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 (Yakub, 2012).

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. 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.

2.3 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
5. 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

1. Algoritma adalah urutan langkah-langkah berhingga untuk memecahkan masalah logika atau matematika
2. Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan.
3. Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.
4. Algoritma adalah urutan logis untuk pemecahan masalah.

Pembuatan algoritma harus selalu dikaitkan dengan:

1. Kebenaran algoritma
2. Kompleksitas (lama dan jumlah waktu proses dan penggunaan memori)

Kriteria Algoritma yang baik:

1. Tepat, benar, sederhana, standar dan efektif
2. Logis, terstruktur dan sistematis
3. Semua operasi terdefinisi
4. Semua proses harus berakhir setelah sejumlah langkah dilakukan

2.3.1 Desain Konseptual

Algoritma adalah serangkaian instruksi, sering disebut sebagai "proses," yang harus diikuti ketika memecahkan masalah tertentu. Meskipun secara teknis tidak dibatasi oleh definisi, kata itu hampir selalu terkait dengan komputer, karena algoritma yang diproses komputer dapat mengatasi masalah yang jauh lebih besar daripada manusia, jauh lebih cepat. Karena komputasi modern menggunakan algoritma jauh lebih sering daripada pada titik lain dalam sejarah manusia, bidang telah tumbuh di sekitar desain, analisis, dan penyempurnaan. Bidang desain algoritma membutuhkan latar belakang matematika yang kuat, dengan gelar ilmu komputer yang sangat dicari kualifikasi. Ini menawarkan semakin banyak pilihan karir yang sangat dikompensasi, karena kebutuhan akan lebih banyak (dan juga lebih canggih) algoritma terus meningkat.

Pada tingkat yang paling sederhana, algoritma pada dasarnya hanya seperangkat instruksi yang diperlukan untuk menyelesaikan tugas. Pengembangan algoritma, meskipun umumnya tidak disebut demikian, telah menjadi kebiasaan yang populer dan pengejaran profesional untuk semua catatan sejarah. Jauh sebelum fajar era komputer modern, orang menetapkan rutinitas yang telah ditentukan untuk bagaimana mereka akan melakukan tugas sehari-hari, sering menuliskan daftar langkah-langkah yang harus diambil untuk mencapai tujuan penting, mengurangi risiko melupakan sesuatu yang penting. Ini, pada dasarnya, adalah apa itu algoritma. Desainer mengambil pendekatan yang mirip dengan pengembangan algoritma untuk tujuan komputasi: pertama, mereka melihat masalah. Kemudian, mereka menguraikan langkah-langkah yang akan diperlukan untuk menyelesaikannya. Akhirnya, mereka mengembangkan serangkaian operasi matematika untuk mencapai langkah-langkah tersebut.

2.3.2 Tugas Algoritma

Tugas sederhana dapat diselesaikan dengan algoritma yang dihasilkan dengan beberapa menit, atau paling banyak pekerjaan pagi. Tingkat kompleksitas menjalankan tantangan yang panjang, namun, sampai pada masalah yang sangat rumit sehingga mereka telah menghalangi matematikawan yang tak terhitung jumlahnya selama bertahun-tahun - atau bahkan berabad-abad. Komputer modern menghadapi masalah pada tingkat ini di bidang-bidang seperti keamanan dunia maya, serta penanganan data besar - penyortiran set data yang efisien dan menyeluruh sedemikian besar sehingga bahkan komputer standar tidak dapat

memprosesnya secara tepat waktu. Contoh data besar mungkin termasuk "setiap artikel di Wikipedia," "setiap halaman web yang diindeks dan diarsipkan akan kembali ke tahun 1998," atau "enam bulan terakhir pembelian online yang dilakukan di Amerika."

2.3.3 Rekayasa Algoritma

Ketika desain algoritma baru diterapkan dalam istilah praktis, disiplin terkait dikenal sebagai rekayasa algoritma. Kedua fungsi tersebut sering dilakukan oleh orang yang sama, meskipun organisasi yang lebih besar (seperti Amazon dan Google) mempekerjakan desainer dan insinyur khusus, mengingat tingkat kebutuhan mereka akan algoritma baru dan khusus. Seperti proses desain, rekayasa algoritma sering kali melibatkan akreditasi sains komputer, dengan latar belakang yang kuat dalam matematika: di mana mereka ada sebagai profesi yang terpisah dan terspesialisasi, insinyur algoritma mengambil ide-ide konseptual dari desainer dan proses kreatif dari mereka yang akan dipahami oleh komputer. Dengan kemajuan teknologi digital yang mantap, para insinyur yang berdedikasi akan terus menjadi semakin umum.

2.4 Algoritma Yang Pernah Ada

2.4.1 Metode Rivest Shamir Adleman (RSA)

Metode *Rivest Shamir Adleman (RSA)* untuk membuat kemanan yang sulit dijebol oleh *hacker*. Karena semua data pelamar dienkripsi kedalam kode *ASCII*. Berikut akan dijelaskan tentang perubahan data yang dilakukan dengan metode *Rivest Shamir Adleman (RSA)*.

Contoh penerapan algoritma *Rivest Shamir Adleman (RSA)*.

Pembangkitan pemasangan kunci

Kunci Publik

n = hasil perkalian dua bilangan prima p dan q

(n) = (p-1)(q-1)

Kunci Pribadi

Enkripsi

$c = m^e \ mod \ n$

Dekripsi

$m = c^d \ mod \ n$

Karakter : Yudi Prima Pratama

Proses Pembuatan Kunci

p = 47

q = 71

$n = p.q$ = 3337

(p-1)(q-1) = 46 * 70 = 3220

e = diperoleh dari bilangan acak = 79

Pasangan Kunci Publik

(n, e) : (3337, 79)

Kunci Private (*Private Key)*

Rumus :

$d = e^{-1} \ mod \ (p-1)(q-1)$

$d = e^{-1} \ mod \ 3220 = 1019$

Pasangan Kunci Private *(Private Key)*

(n, d) : (3337, 1019)

Untuk menyelesaikan soal di atas maka dilakukan langkah-langkah sebagai berikut:

A. Proses mengubah karakter ke *ASCII*

Proses merubah karakter teks terang ke dalam bentuk *ASCII*

Yudi Prima Pratama = 121 117 100 105 112 114 105 109 97 112 114 97 116 97 109 97

B. Proses Perhitungan *Enkripsi*

Rumus :

$c =$ $\quad mod \ n$

Pasangan Kunci Publik

(n, e) : (3337, 79)

maka nilai *m* dibagi menjadi 11 blok

m1 = 1211	m5 = 1410	m9 = 9711
m2 = 1710	m6 = 5109	m10 = 6971
m3 = 0105	m7 = 9711	m11 = 0997
m4 = 1121	m8 = 2114	

Proses enkripsi

$1211^{79} \mod 3337 = 935$

$1710^{79} \mod 3337 = 2938$

$0105^{79} \mod 3337 = 193$

$1121^{79} \mod 3337 = 579$

$1410^{79} \mod 3337 = 423$

$5109^{79} \mod 3337 = 978$

$9711^{79} \mod 3337 = 211$

$2114^{79} \mod 3337 = 140$

$9711^{79} \mod 3337 = 211$

$6971^{79} \mod 3337 = 78$

$0997^{79} \mod 3337 = 1436$

Hasil enkripsi = 935293819357942397821114021178143 6

2.5 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* (Stallings, 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).
2. *Deskripsi*, adalah kebalikan dari *Enkripsi* yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi awal.

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). *File* yang dapat di *Enkripsi* dapat berupa teks, gambar maupun audio dan video (Pabokory et al., 2015).

2.5.1 Kriptografi Klasik

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 plaintext (Bishop, 2002).

Sebelum komputer ada, kriptografi dilakukan dengan menggunakan pensil dan kertas. Algoritma kriptografi (*cipher*) yang digunakan saat itu, dinamakan juga algoritma klasik, adalah berbasis karakter, yaitu enkripsi dan dekripsi dilakukan pada setiap karakter pesan. Semua algoritma klasik termasuk ke dalam sistrm kriptografi simetris dan digunakan jauh sebelum kriptografi kunci publik ditemukan.

Kriptogarfi klasik memiliki beberapa ciri :

1. Berbasis karakter
2. Menggunakan pena dan kertas saja, belum ada komputer
3. Termasuk ke dalam kriptografi kunci simetris.

Tiga alasan mempelajari algoritma klasik :

1. Memahami konsep dasar kriptografi
2. Dasar algoritma kriptografi modern
3. Memahami kelemahan sistem kode

Pada dasarnya, algoritma kriptografi klasik dapat dikelompokkan ke dalam dua macam cipher, yaitu :

1. Cipher substitusi (*substitution cipher*)

   Di dalam cipher substitusi setiap unit plainteks diganti dengan satu unit cipherteks. Satu “unit” di isini berarti satu huruf, pasanga huruf, atau dikelompokkan lebih dari dua huruf. Algoritma substitusi tertua yang diketahui adalah *Caesar cipher* yang digunakan oleh kaisar Romawi , Julius Caesar (sehingga dinamakan juga *casear cipher*), untuk mengirimakan pesan yang dikirimkan kepada gubernurnya.

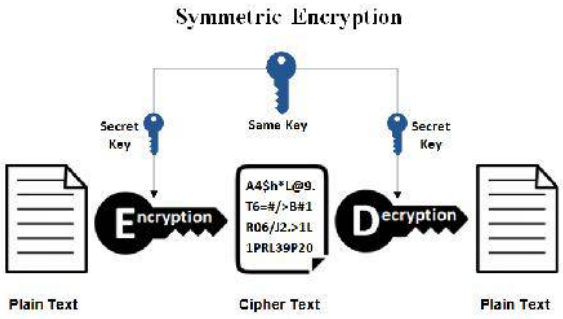
2. Cipher transposisi (*transposition cipher*)

   Pada cipher transposisi, huruf-huruf di dalam plainteks tetap saja, hanya saja urutannya diubah. Dengan kata lain algoritma ini melakukan transpose terhadap rangkaian karakter di dalam teks. Nama lain untuk metode ini adalah permutasi atau pengacakan (*scrambling*) karena transpose setiap karakter di dalam teks sama dengan mempermutasikan karakter-karkater tersebut.

2.5.2 Kriptografi Simetris

Kriptografi kunci simetris adalah setiap algoritma kriptografi yang didasarkan pada kunci bersama yang digunakan untuk mengenkripsi atau mendekripsi teks / *ciphertext*, dalam kontrak dengan kriptografi kunci asimetris, di mana kunci enkripsi dan dekripsi dihubungkan oleh berbeda. Enkripsi simetris umumnya lebih efisien daripada enkripsi asimetris dan karenanya lebih disukai

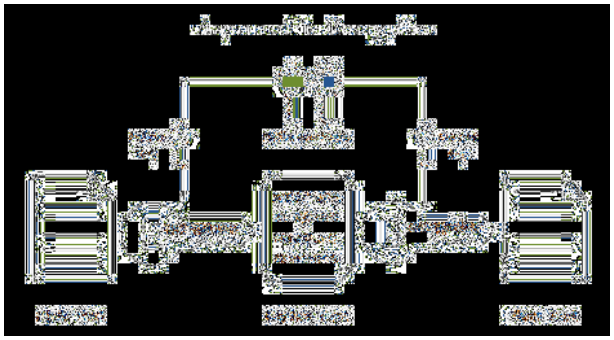
ketika sejumlah besar data perlu dipertukarkan. Membuat kunci bersama sulit menggunakan hanya algoritma enkripsi simetris, sehingga dalam banyak kasus, enkripsi asimetris digunakan untuk membuat kunci bersama antara dua pihak. Contoh untuk kriptografi kunci simetris termasuk AES, DES, dan 3DES. Protokol pertukaran kunci yang digunakan untuk membangun kunci enkripsi bersama termasuk Diffie-Hellman (DH), Eliptic Curve (EC) dan RSA. Gambar berikut adalah skema dari kriptografi simetris(Ayushi, 2010).



Gambar 2.1 Skema kriptografi simetris
Sumber: (Ayushi, 2010)

2.5.3 Kriptografi Asimetris

Dalam versi kriptografi asimetris, pengirim dan penerima memiliki dua kunci, publik dan pribadi. Kunci pribadi dirahasiakan sedangkan kunci publik terbuka ke dunia luar. Set data apa pun, yang dienkripsi dengan kunci publik hanya dapat didekripsi menggunakan kunci pribadi yang sesuai. Ketika datang ke perbandingan, pendekatan simetris lebih cepat daripada yang asimetris. Contoh - tanda tangan digital menggunakan kriptografi asimetris untuk mengenkripsi pesan dalam hash alih-alih pesan lengkap. Berikut ini skema kriptografi asimetris(S., L. Ribeiro, & David, 2012).



Gambar 2.2 Skema kriptografi asimetris
Sumber: (Ayushi, 2010)

2.6 Beaufort Cipher

Algoritma Beaufort, dibuat oleh Sir Francis Beaufort, adalah cipher pengganti yang mirip dengan cipher Vigenère, dengan mekanisme penyandian yang sedikit dimodifikasi dan tablo. Aplikasi yang paling terkenal adalah di mesin sandi berbasis rotor, Hagelin M-209. Algoritma Beaufort didasarkan pada kotak Beaufort yang pada dasarnya sama dengan kotak Vigenère tetapi dalam urutan terbalik dimulai dengan huruf "Z" di baris pertama, di mana baris pertama dan kolom terakhir memiliki tujuan yang sama. Tujuan utama dari *Beaufort cipher* ini adalah menyembunyikan keterhubungan antara *plaintext* dan *Beaufort* dengan menggunakan kata kunci sebagai penentu pergeseran karakternya (Pratama & Tamatjita, 2015).

Tabel yang digunakan merupakan tabel 26 huruf alfabetik standart, yang dimulai dari A sampai Z. Panjang kunci tersebut bisa lebih pendek ataupun sama dengan *plaintext*, maka kunci tersebut akan diulang secara periodik hingga panjang kunci tersebut sama dengan panjang *plaintext*nya. Berikut ini rumus enkripsi dan dekripsi *Beaufort Cipher*:

$$E : C = Pi + k \, mod \, 26$$

$$D : P = Ci - k \, mod \, 26$$

Keterangan:

C : Ciphertext

P : Plaintext

k : Key atau kunci

Bruen (Bruen & Forcinito, 2005) dalam bukunya Cryptography, Information Theory, and Error-Correction, serta Martin (Martin, 2012) dalam bukunya Everyday Cryptography, mengatakan bahwa Beaufort cipher adalah sebuah metode dari enkripsi teks alfabetik menggunakan serangkaian penyandian bebasis caesar pada huruf-huruf dari sebuah kata kunci, dan merupakan bentuk sederhana dari subtitusi polyalphabetic.

Teknik subtitusinya serupa dengan semua penyandian berbasis Caesar. Seperti penyandian berbasis Caesar lainnya, Beaufort cipher sebenarnya juga melakukan pergeseran, tetapi pergeseran dilakukan perhuruf dengan huruf berikutnya pada plaintext berbeda. Dengan demikian jika pada Caesar cipher seseorang dengan mudah menebak kuncinya dengan melakukan pergeseran abjad mulai dari 1 s/d 26 secara cara try and error, sampai ditemukan nilai kunci yang tepat. Maka pada pada Beaufort cipher akan lebih sulit menebak kuncinya dengan try and error mencari nilai pergeseran seperti pada Caesar cipher, karena antara

huruf yang satu dengan huruf berikutnya mempunyai nilai pergeseran yang berbeda.

Beaufort Cipher menggunakan suatu kunci yang memiliki panjang tertentu. Panjang kunci tersebut bias lebih pendek ataupun sama dengan plaintextnya. Jika panjang kunci kurang dari panjang plaintext, maka kunci tersebut akan diulang secara periodic sehingga panjang kunci tersebut akan sama panjang dengan plaintextnya.

Formula atau rumus enkripsi *Beaufort Cipher*

$$Ci = (Pi + Ki)\ mod\ 26$$

Formula atau rumus deskripsi *Beaufort Cipher*

$$Pi = (Ci - Ki)\ mod\ 26\ ;\ untuk\ Ci >= Ki$$

$$Pi = (Ci + 26 - Ki)\ mod\ 26\ ;\ untuk\ Ci <= Ki$$

Keterangan:

Ci = nilai decimal karakter *ciphertext* ke-i

Pi = nilai decimal karakter *ciphertext* ke-i

Ki = nilai decimal karakter *ciphertext* ke-i

Contoh bila kita ingin memberi sandi pada kata ILMU KOMPUTER dan kunci yang diinginkan adalah UNPAB maka enkripsi dilakukan sebagai berikut:

Plaintext : I L M U K O M P U T E R

Kunci : U N P A B

Karena metode yang diterapkan pada *Beaufort Cipher* adalah dengan menyusun kunci yang panjangnya akan disesuaikan dengan panjang plaintextnya, maka kunci akan mengalami perulangan sampai memenuhi banyak karakter yang terdapat pada plaintext.

Plaintext : I L M U K O M P U T E R

Kunci : U N P A B U N P A B U N

Langkah selanjutnya adalah masuk ke proses enkripsi dengan metode *Beaufort* menggunakan formula atau rumus Ci = (Pi + Ki) mod 26.

Plaintext : I L M U K O M P U T E R

Kunci : U N P A B U N P A B U N

Ciphertext : B Y B U L I Z E U U Y E

Ciphertext didapatkan dengan cara menggunakan formula Ci = (Pi + Ki) mod 26.

$$C1 = (I + U)\ mod\ 26\ (8 + 20)\ mod\ 26 = 2 = B$$

$$C2 = (L + N)\ mod\ 26\ (11 + 13)\ mod\ 26 = 24 = Y$$

Setelah seluruh karakter mendapatkan ciphertext, maka proses enkripsi sudah selesai. Proses deskripsi dilakukan untuk memecahkan ciphertext kembali menjadi plaintext. Untuk proses deskripsi digunakan rumus atau formula

$$Pi = (Ci - Ki)\ mod\ 26\ ;\ untuk\ Ci >= Ki$$

$$Pi = (Ci + 26 - Ki)\ mod\ 26\ ;\ untuk\ Ci <= Ki$$

Ciphertext : B Y B U L I Z E U U Y E

Kunci : U N P A B U N P A B U N

Plaintext : I L M U K O M P U T E R

Seperti pada karakter pertama dimana C1 bernilai 2 dan K1 bernilai 20 maka digunakan formula atau rumus Pi = (Ci+26-Ki) mod 26 karena nilai C1 <= K1

$$P1 = (2 + 26 - 20)\ mod\ 26 = 8 = I$$

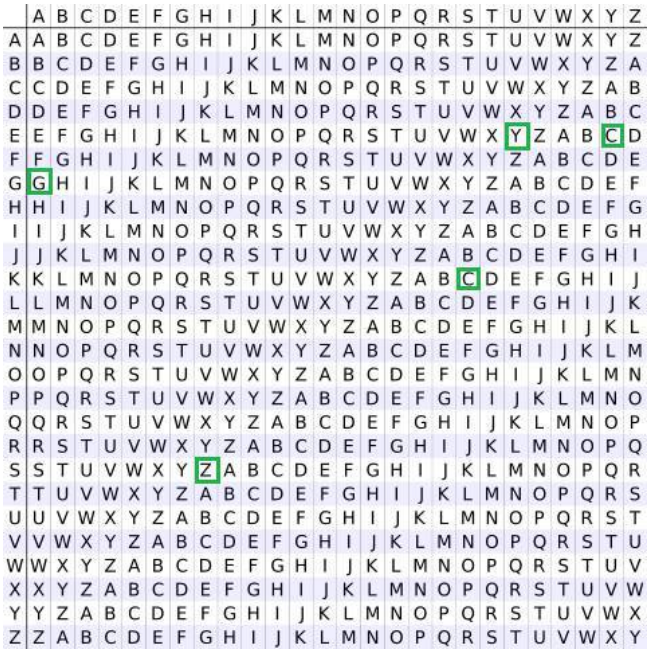
Sementara pada karakter kedua dimana C2 bernilai 24 dan K2 bernilai 13 maka digunakan formula atau rumus Pi = (Ci-Ki) mod 26 karena nilai C2 >= K2

$$P2 = (24 - 13)\ mod\ 26 = 11 = L$$

Selain itu kunci juga dapat berupa angka, berikut adalah table huruf beserta angka yang akan membantu proses enkripsi.

2.6.1 Enkripsi

Huruf pertama dari plaintext, G dipasangkan A, huruf pertama dari kunci. Gunakan baris G dan kolom A dari kotak Beaufort, yaitu G.Untuk huruf kedua dari plaintext, huruf kedua dari kunci digunakan, huruf pada baris E dan kolom Y adalah C. Sisa dari plaintext dienkripsi dengan cara yang serupa.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 2.3 Tabel Beaufort
Sumber: (Ayushi, 2010)

2.6.2 Dekripsi

Dekripsi dilakukan dengan pergi ke baris dalam tabel yang sesuai dengan kunci, menemukan posisi huruf ciphertext di baris ini, dan kemudian menggunakan label kolom sebagai *plaintext*. Misalnya, di baris A (dari AYUSH), *ciphertext* G muncul di kolom G, yang merupakan huruf plaintext pertama. Selanjutnya, kita pergi ke baris Y (dari AYUSH), cari ciphertext C yang ditemukan di kolom E, jadi E adalah huruf plaintext kedua.

2.7 Unified Modelling Language

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain yang berisi sintak dalam memodelkan sistem secara visual. 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* (Technopedia, 2019).

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).

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).

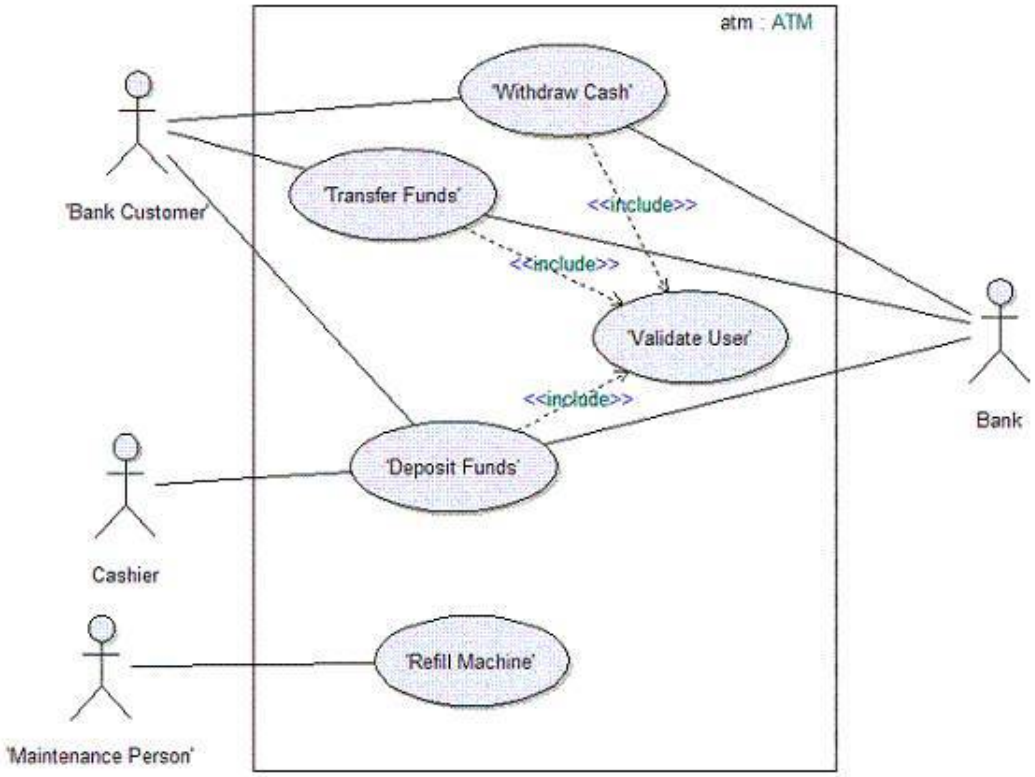
2.7.1 Use Case Diagram

*Use C*ase *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).

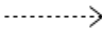
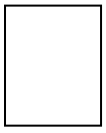
Diagram ini menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram *use case diagram*.



Gambar 2.4 Use-case Diagram ATM
Sumber: (Uml-diagrams.org, 2019)

Gambar 2.4 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* dapat dilihat pada tabel 2.1.

Tabel 2.1 Simbol Use Case Diagram

No	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpunan 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		*Generalizatio n*	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).

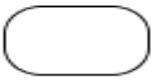
Sumber: (Kurniawan, 2018)

2.7.2 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 (Ladjamudin, 2005).

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 ActivityDiagram

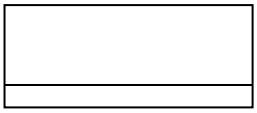
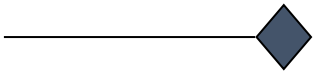
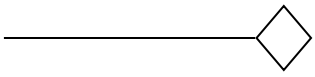
No	Gambar	Nama	Keterangan
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk /diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk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

Sumber: (Kurniawan, 2018)

2.7.3 Class Diagram

Class diagram digunakan untuk menggambarkan perbedaan yang mendasar antara *class*, hubungan antara *class*, dan di mana *sub-sistem class* tersebut (Jogiyanto, 2006). Simbol yang digunakan dalam *class diagram* adalah sebagai berikut:

Tabel 2.3 Simbol Class Diagram

Simbol	Nama	Fungsi
	*Class*	Menggambarkan *Class* baru pada diagram.
	Association	Menggambarkan relasi antar asosiasi
	*Composition*	Jika sebuah *class* tidak bisa berdiri sendiri dan harus merupakan bagian dari *class* yang lain, maka class tersebut memiliki relasi *Composition* terhadap *class* tempat dia bergantung tersebut.
	Depedency	Umumnya penggunaan *dependency* digunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain.
	*Aggregation*	*Aggregation* mengindikasikan keseluruhan bagian *relationship* dan biasanya disebut sebagai relasi.

Sumber: (Kurniawan, 2018)

2.8 *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.

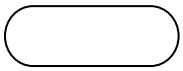
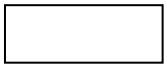
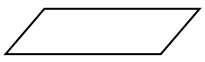
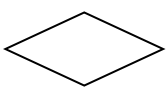
Diagram alir menggambarkan aspek-aspek tertentu dari proses dan biasanya dilengkapi dengan jenis diagram lainnya. Misalnya, Kaoru Ishikawa, mendefinisikan diagram alir sebagai salah satu dari tujuh alat dasar kendali mutu, di sebelah histogram, diagram Pareto, lembar periksa, diagram kontrol, diagram

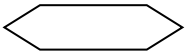
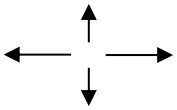
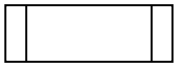
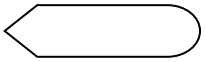
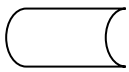
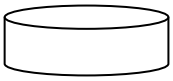
sebab-akibat, dan diagram sebaran. Demikian pula, di UML, notasi pemodelan konsep standar yang digunakan dalam pengembangan perangkat lunak, diagram aktivitas, yang merupakan jenis diagram alur, hanyalah salah satu dari banyak jenis diagram yang berbeda.

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 untuk di simpan dengan semantara waktu itu di flowchart yang bentuk seperti ini
12		Menunjukkan input / Output Hardisk (media penyimpanan)

Sumber: (Kurniawan, 2018)

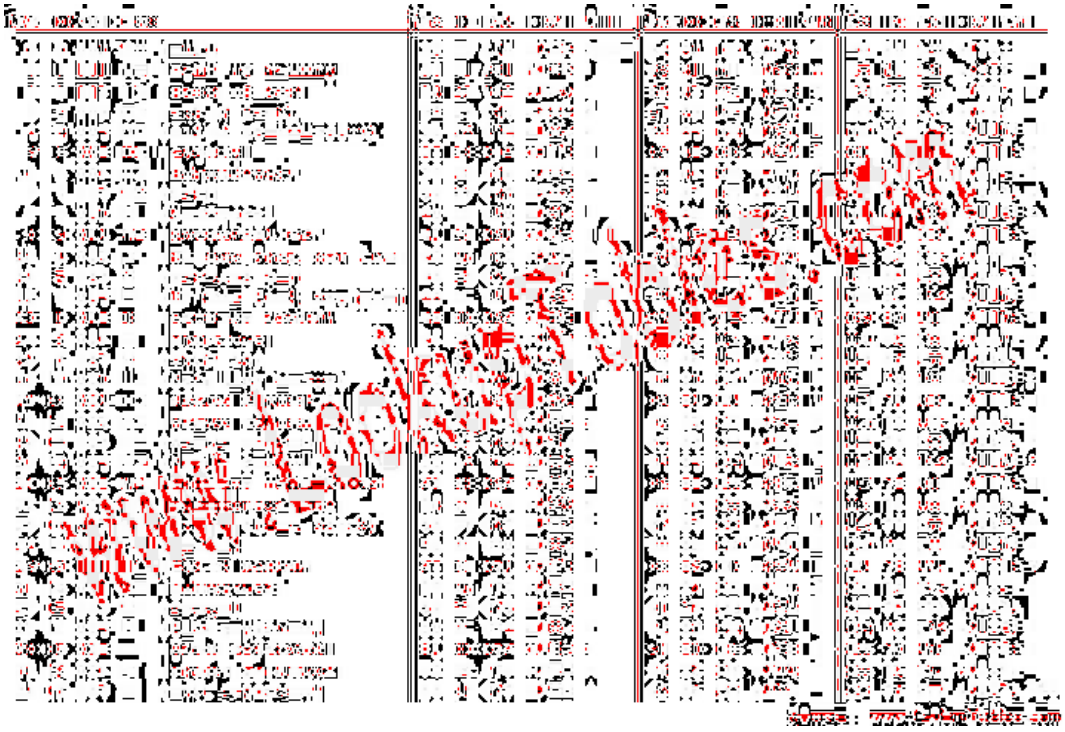
2.9 Tabel ASCII

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, contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. sedangkan fungsi dari kode ASCII ialah digunakan untuk mewakili karakter-karakter angka maupun huruf di dalam komputer, sebagai contoh dapat kita lihat pada karakter 1, 2, 3, A, B, C, dan sebagainya.

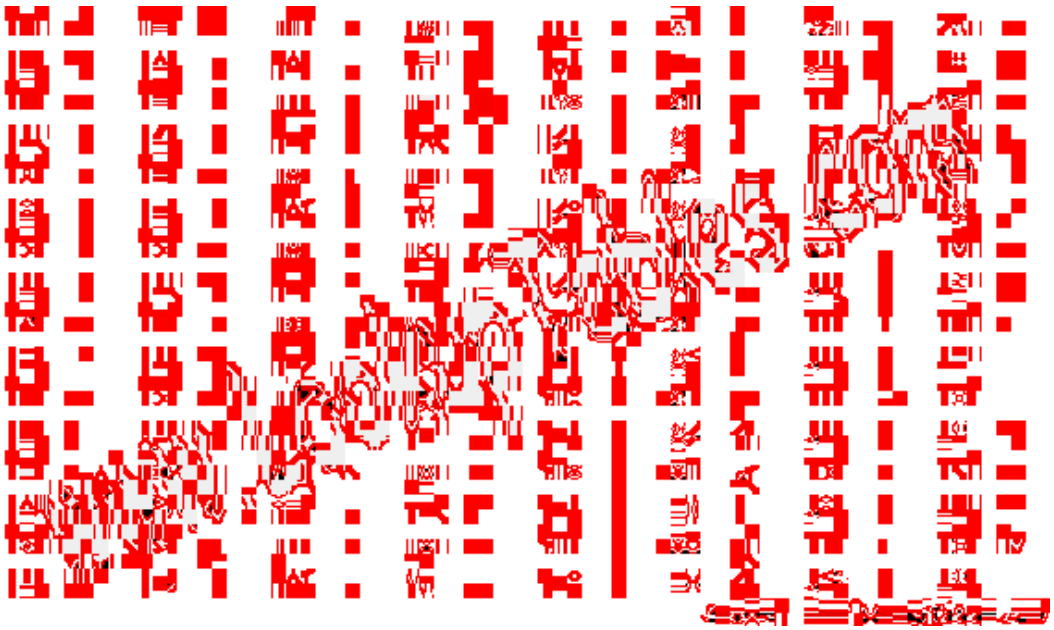
File teks yang disimpan dalam format ASCII kadang-kadang disebut file ASCII. Editor teks dan pengolah kata biasanya mampu menyimpan data dalam format ASCII, meskipun format ASCII tidak selalu merupakan format penyimpanan standar. Sebagian besar file data, terutama jika mengandung data numerik, tidak disimpan dalam format ASCII. Program yang dapat dieksekusi tidak pernah disimpan dalam format ASCII. Set karakter ASCII standar hanya menggunakan 7bit untuk setiap karakter. Ada beberapa set karakter yang lebih besar yang menggunakan 8bit, yang memberi mereka 128 karakter tambahan. Karakter tambahan digunakan untuk mewakili karakter non-Inggris, simbol grafik, dan simbol matematika. Beberapa perusahaan dan organisasi telah mengusulkan ekstensi untuk 128 karakter ini. Sistem operasi DOS menggunakan superset ASCII yang disebut extended ASCII atau ASCII tinggi. Standar yang lebih universal adalah set karakter ISO Latin 1, yang digunakan oleh banyak sistem operasi, serta

browser Web. Seperangkat kode yang digunakan pada komputer IBM besar adalah EBCDIC (Beal, 2019).

Jika seseorang mengatakan mereka menginginkan CV Anda dalam format ASCII, semua ini berarti mereka menginginkan teks 'polos' tanpa format seperti tab, tebal atau garis bawah - format mentah yang dapat dipahami oleh komputer mana pun. Ini biasanya agar mereka dapat dengan mudah mengimpor file ke aplikasi mereka sendiri tanpa masalah. Notepad.exe membuat teks ASCII, atau dalam MS Word dapat menyimpan file sebagai 'teks saja' (Spencer, 2019). Gambar berikut ini adalah daftar tabel ASCII yang standard dan extended.



Gambar 2.5 Daftar ASCII Standard
Sumber:(Spencer, 2019)



Gambar 2.6 Daftar ASCII Extended
Sumber: (Spencer, 2019)

2.10 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*

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.

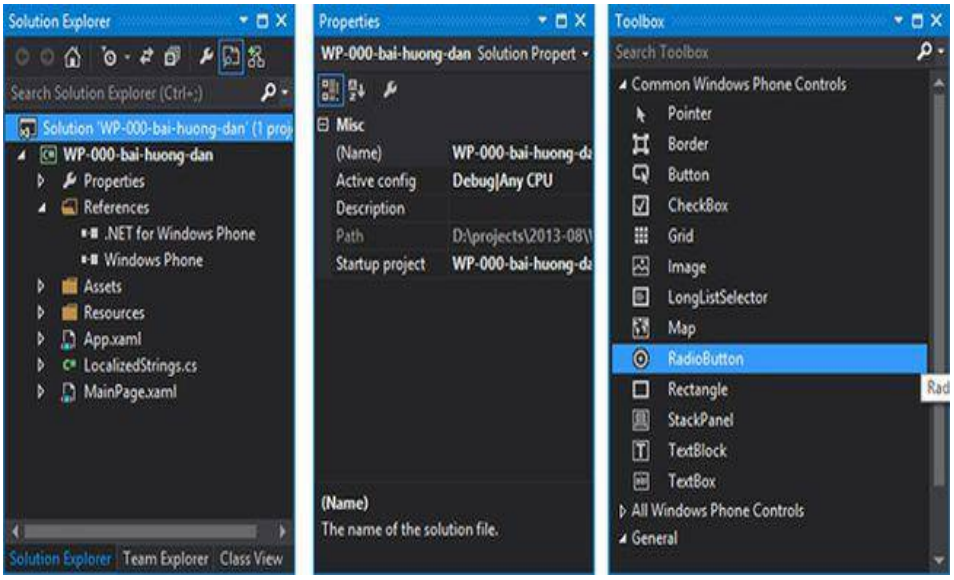
Dalam pemrograman untuk bisnis, *Visual Studio* memiliki pangsa pasar yang sangat luas. Dalam sebuah survey yang dilakukan pada tahun 2005, 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk *Visual Studio*, yang diikuti oleh *C++*, *JavaScript*, *C#*, dan *Java*.

2.10.1 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. Komponen dapat disesuaika di dalam program visual Studio 2015 sesuai dengan kebutuhan.

2.10.2 Toolbox

Toolbox adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain mulai dari tombol *label, pointer, button*, dan lain-lain. Gambar berikut adalah *table* yang berisi nama tombol yang terdapat didalam *toolbox* beserta fungsinya.



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

Tabel 2.5 Toolbox Visual Studio

Nama	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: (Rahmel, 2008)

BAB III

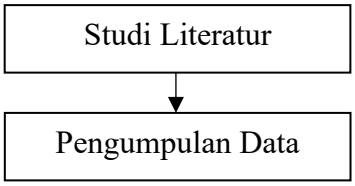
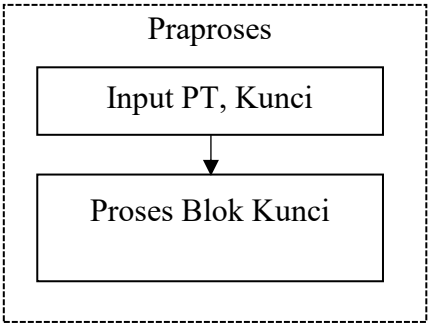
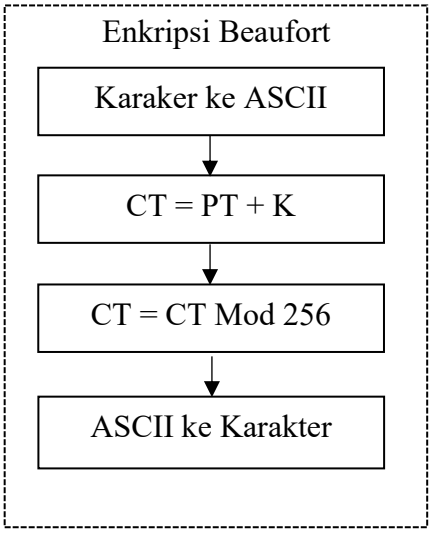
METODE PENELITIAN

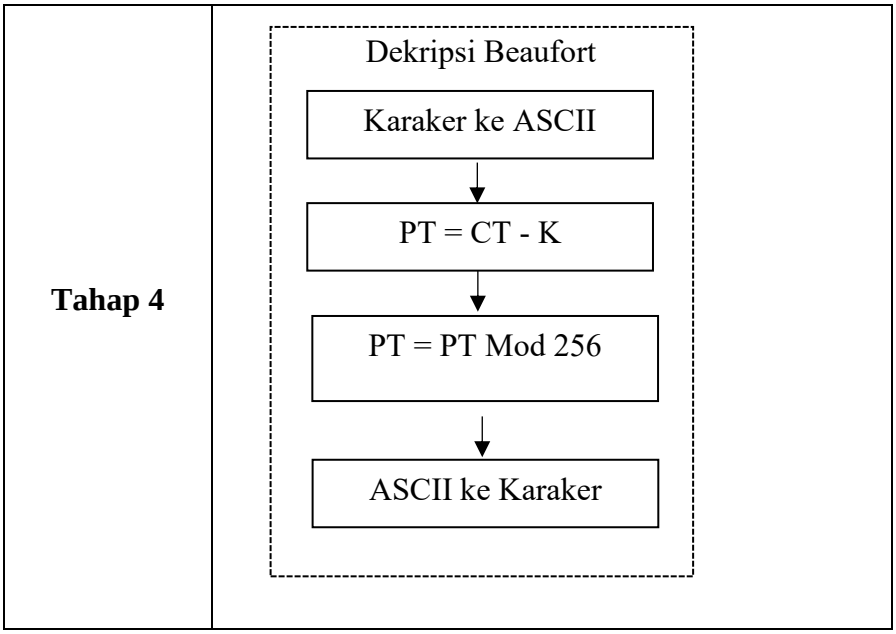
3.1 Kerangka Algoritma Beaufort

Kerangka algortima ini adalah bagaimana tahapan dilakukan untuk menentukan hasil ciphertext dan plaintext dengan algoritma Beaufort Cipher. Kerangka algoritma dengan jelas menggambarkan struktur rencana penelitian dan membantu peneliti dalam membuat program aplikasi dengan baik dan benar. Dalam melakukan penelitian, kerangka ini akan dibagi menjadi empat tahap yaitu:

1. Studi literatur dan pengumpulan data.
2. Praproses, menyiapkan plaintext dan kunci.
3. Enkripsi, proses transformasi plaintext ke ciphertext.
4. Dekripsi, proses transformasi ciphertext ke plaintext.

Untuk menjelaskan bagaimana alur kerja dari penelitian ini, akan digambarkan sebuah kerangka dalam menentukan batasan kerja dari yang akan dibahas. Gambar 3.1 adalah kerangka dari algoritma Beaufort Cipher yang dilakukan.

Tahap 1	
Tahap 2	
Tahap 3	



Gambar 3.1 Kerangka Algoritma Beaufort

Berikut adalah tahapan penelitian yang dilakukan:

1. Studi Literatur dan pengumpulan data

   Studi literatur dan pengumpulan data bertujuan untuk mendapatkan teori-teori tentang ilmu kriptografi khususnya algoritma Beaufort Cipher serta teknik pengumpulan data agar tujuan program aplikasi tercapai.

2. Praproses

   Bagian ini menjelaskan proses persiapan plaintext dan kunci yang akan dijadikan sebagai input pada program aplikasi.

3. Enkripsi

   Bagian ini membahas proses enkripsi plaintext menjadi ciphertext dengan algoritma Beaufort Cipher.

4. Dekripsi

   Bagian ini membahas proses pengembalian ciphertext menjadi plaintext dengan algoritma Beaufort Cipher.

5. Implementasi dan pengujian

   Bagian ini dilakukan pengujian kebenaran output yang dihasilkan oleh program aplikasi Microsoft Visual Basic.Net 2010. Hasil program aplikasi akan dibandingkan dengan perhitungan manual.

3.2 Analisa Perbandingan

Algoritma Beaufort lebih baik dari algorima klasik lainnya. Pada penelitian sebelumnya yang membahas algoritma Caesar Cipher. Algoritma ini sangat rentan terhadap percobaan pemecahan ciphertext (Leighton, 1969). Algoritma ini dapat dipecahkan dengan memutar ciphertext terebut dengan hanya 26 atau 256 putaran sehingga plaintext dengan utuh dapat dipecahkan kembali. Algoritma Beaufort memperbaiki dengan cara memberikan pasangan karakter plaintext dengan kunci dengan kunci yang berebeda sementara algoritma Caesar hanya memberikan angka yang sama untuk tiap karakter plaintext tersebut sehingga apabila ada satu karakter yang berhasil dipecahkan maka ini berlaku untuk semua plaintext.

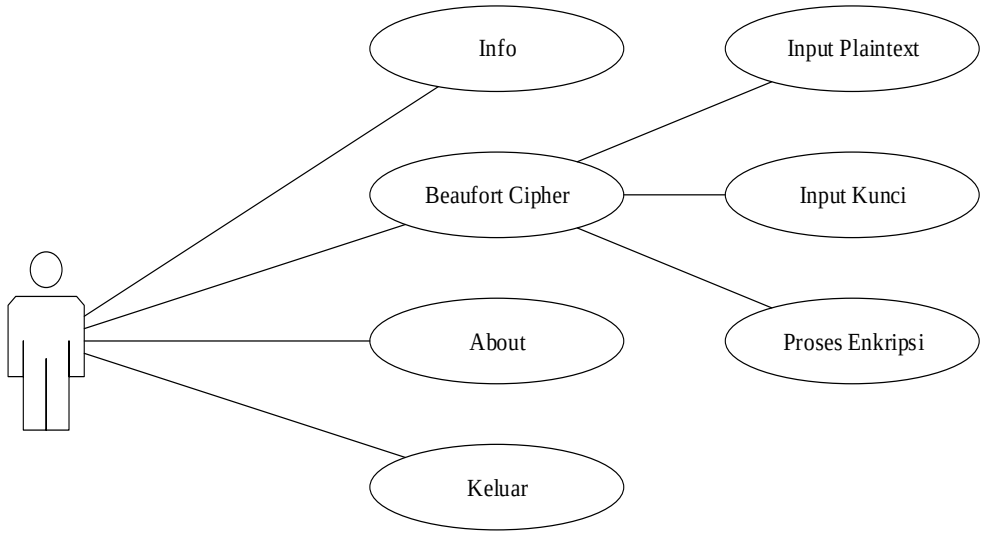
3.3 Algoritma yang diusulkan

Perancangan penelitian adalah bagaimana suatu penelitian dimodelkan dalam suatu alur atau diagram. Banyak cara yang dapat dilakukan untuk merancang penelitian agar lebih terarah dan terstruktur. Perancangan ini membutuhkan

ketelitian yang tinggi agar tidak menyalahi aturan yang ada. Hal ini bertujuan agar program aplikasi yang telah dibuat dapat bekerja secara efisien dan efektif. Desain penelitian dapat digambarkan dengan bentuk Unified Modelling Language (UML). Pada UML, setiap arah penelitian tergambarkan dengan jelas dan terstruktur. Hal ini dapat memudahkan peneliti dalam menghasilkan output yang benar.

3.3.1 Use Case Diagram

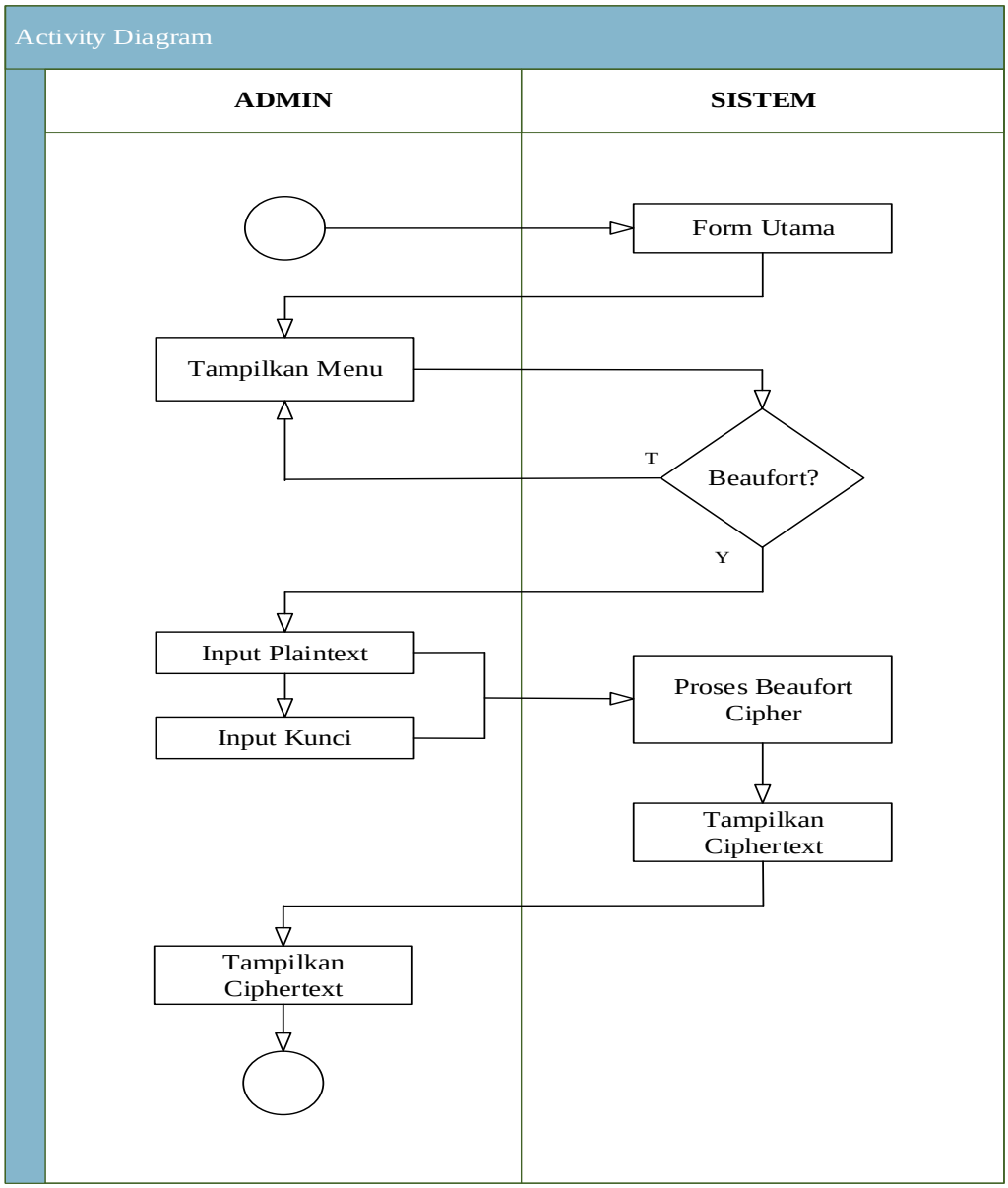
Use Case diagram didefinisikan sebagai diagram yang menangkap fungsionalitas dan persyaratan sistem dalam UML. Use-cases adalah konsep inti dari pemodelan bahasa Unified Modeling. Use Case terdiri dari use case, orang, atau berbagai hal yang menggunakan fitur yang disebut sebagai aktor dan elemen yang bertanggung jawab untuk mengimplementasikan use case. Use case diagram menangkap perilaku dinamis dari sistem live. Ini memodelkan bagaimana entitas eksternal berinteraksi dengan sistem untuk membuatnya bekerja. Use case diagram bertanggung jawab untuk memvisualisasikan hal-hal eksternal yang berinteraksi dengan bagian dari sistem. Gambar 3.2 adalah perancangan *Use Case* untuk algoritma Beaufort Cipher.



Gambar 3.2 Use Case Diagram

3.3.2 Activity Diagram

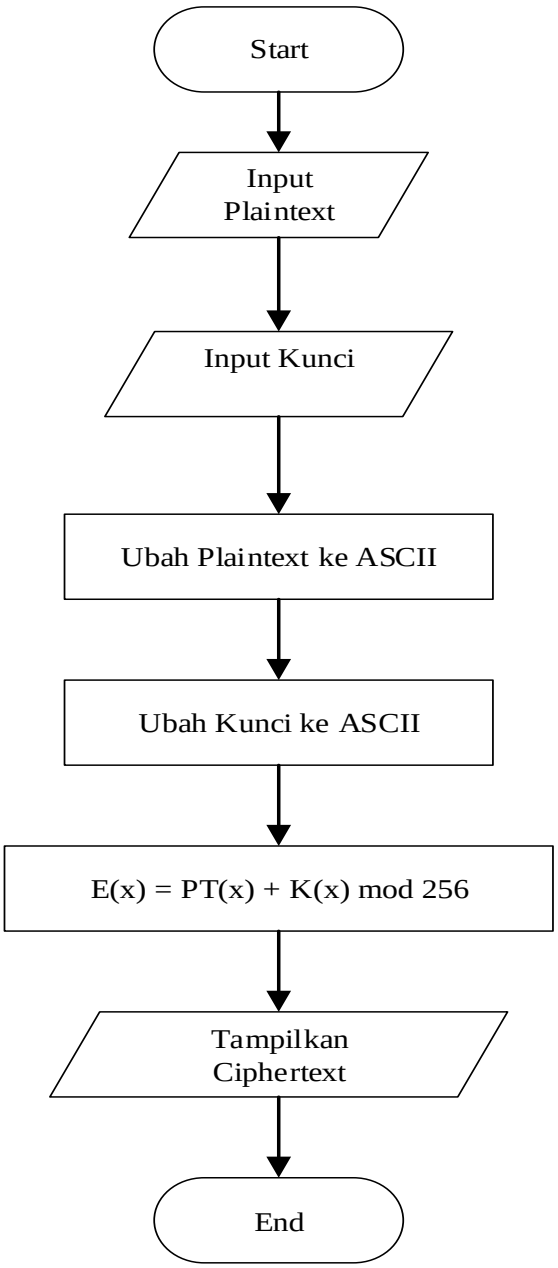
Activity diagram menggambarkan perilaku alur kerja aktual suatu sistem dalam Teknologi Informasi. Diagram ini menggambarkan keadaan aktual dari suatu sistem dengan menunjukkan semua urutan kegiatan yang dilakukan. Juga, diagram ini dapat menunjukkan aktivitas yang kondisional atau paralel. Gambar berikut ini akan menjelaskan *Activity diagram* tersebut.



Gambar 3.3 Activity Diagram

3.3.3 Flowchart Enkripsi

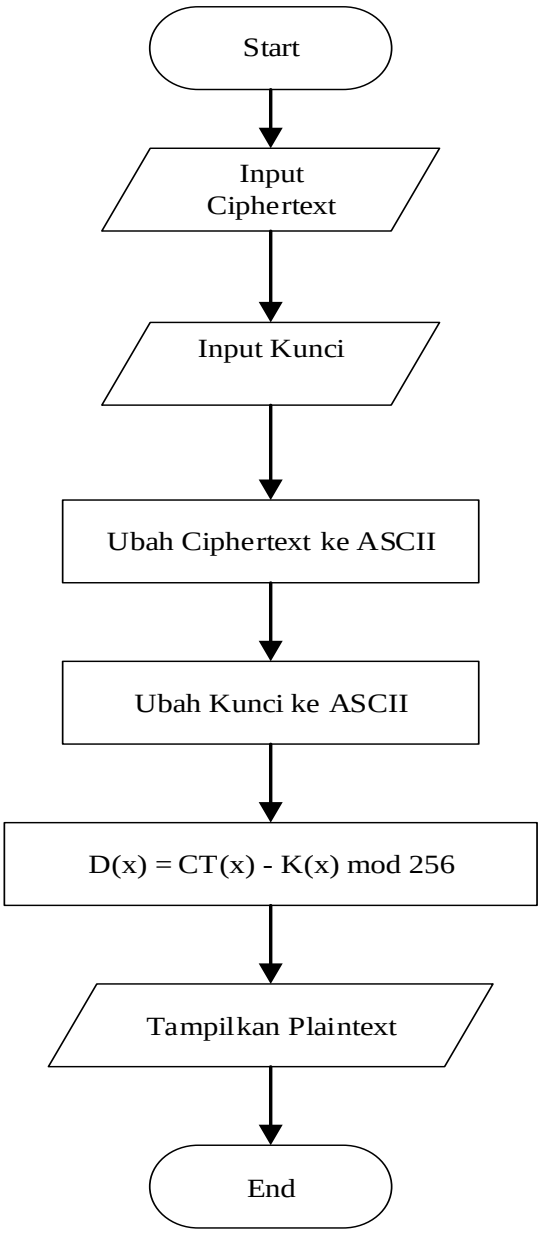
Flowchart enkripsi akan menerangkan cara kerja algoritma Beaufort Cipher dengan proses enkripsi. *Flowchart* enkripsi dapat dilihat pada gambar 3.4.



Gambar 3.4 Flowchart enkripsi algoritma Beaufort

3.3.4 Flowchart Dekripsi

Flowchart dekripsi akan menjelaskan cara kerja algoritma Beaufort Cipher dengan proses dekripsi. *Flowchart* dekripsi dapat dilihat pada gambar 3.5.



Gambar 3.5 Flowchart dekripsi algoritma Beaufort

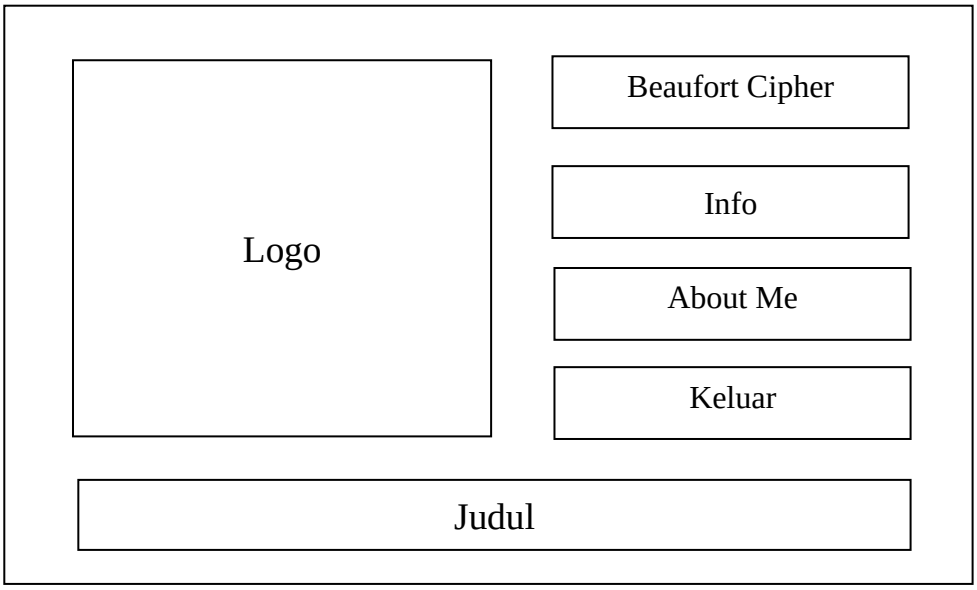
3.4 Desain Interface

Desain interface atau perancangan antarmuka adalah bagaimana suatu tampilan pada program aplikasi akan dibuat. Perancangan ini melibatkan beberapa objek yang akan disematkan pada program aplikasi tersebut. Dalam perancangan

ini dapat dilihat apa-apa saja yang akan digunakan dalam program aplikasi. Perancangan ini memudahkan penulis dalam menentukan dan memodifikasi apabila terjadi perubahan pada program aplikasi algoritma Beaufort Cipher.

3.4.1 Desain Interface Menu Utama

Menu Utama adalah tampilan yang pertama sekali muncul pada saat progam aplikasi dijalankan. Gambar 3.6 adalah hasil perancangan menu utama yang memiliki beberapa komponen lainnya.



Gambar 3.6 Desain Menu Utama

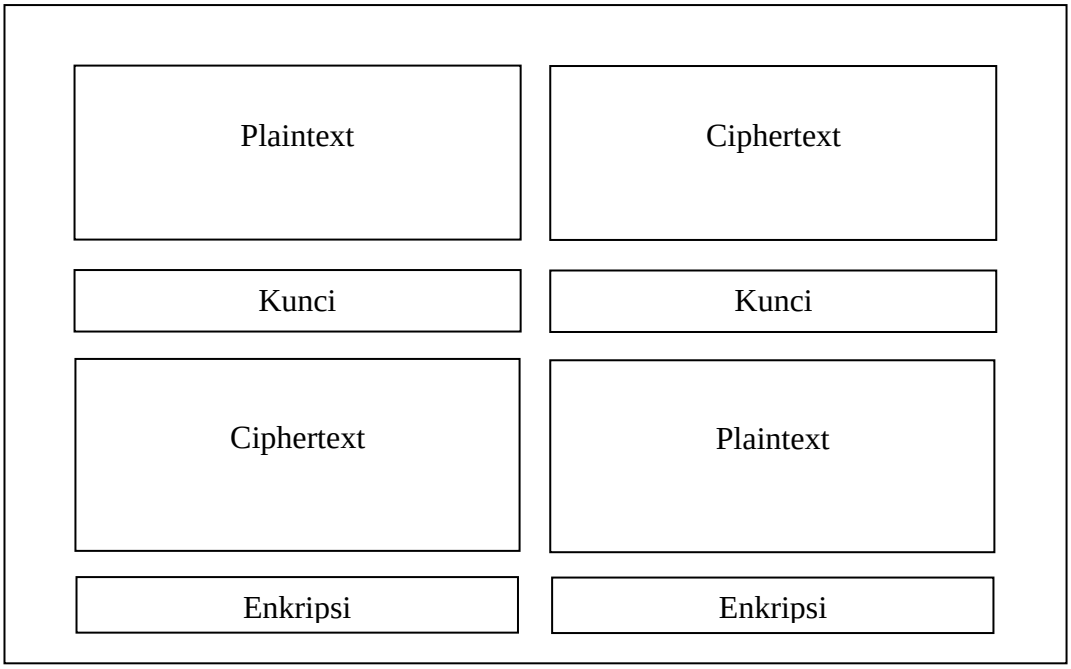
Desain ini memiliki berapa sub-menu antara lain:

- Logo
- Beaufort Cipher
- Info
- About Me

- Keluar
- Judul

3.4.2 Desain Interface Menu Beaufort Cipher

Menu ini adalah perancangan program aplikasi utama yang berfungsi menjalankan algoritma Beaufort Cipher. Tampilan ini adalah tempat untuk melakukan proses enkripsi dan dekripsi. Gambar 3.7 adalah tampilan menu ini.



Gambar 3.7 Desain Menu Beaufort Cipher

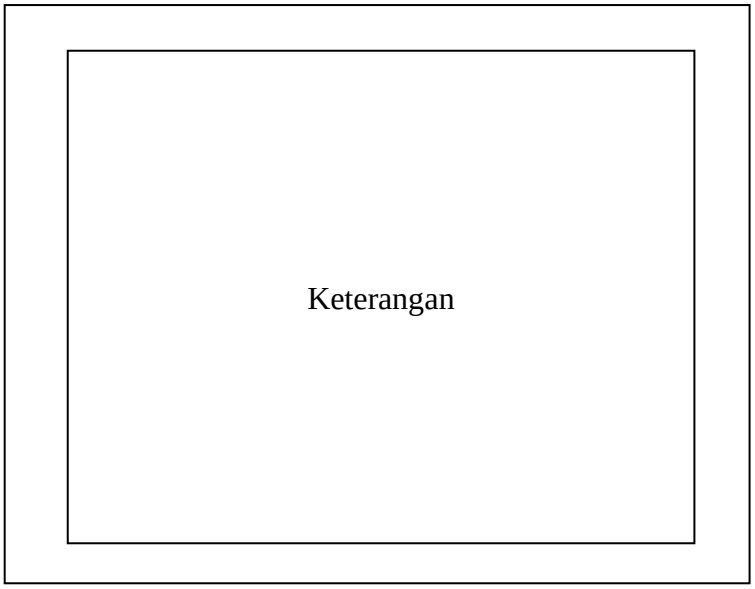
Tampilan algoritma Beaufort Cipher memiliki beberapa bagian antara lain:

- Plaintext
- Ciphertext
- Kunci
- Tombol Enkripsi

- Tombol Dekripsi
- Log

3.4.3 Desain Interface Menu Info

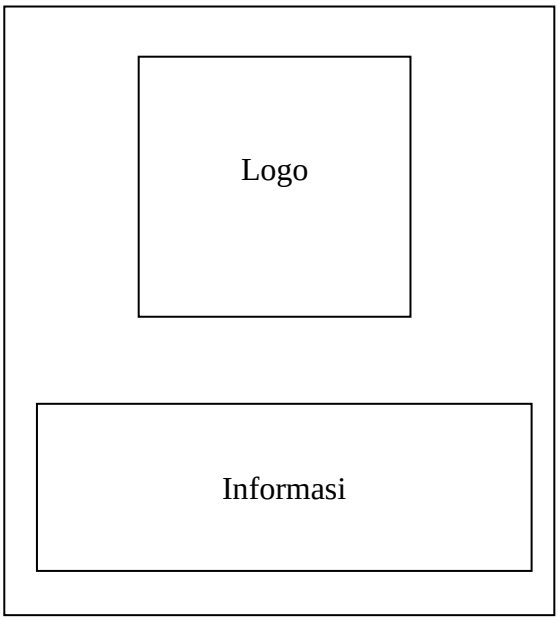
Menu ini menampilkan informasi tentang algoritma Beaufort Cipher. Tampilan ini terdiri dari objek gambar dan keterangan. Gambar 3.8 adalah hasil perancangan menu Info.



Gambar 3.8 Desain Menu Info

3.4.4 Desain Interface Menu About Me

Menu ini menampilkan informasi penulis dan institusi dimana penulis melakukan penelitian. Tampilan ini teridiri dari logo Universitas Pembangunan Panca Budi dan biodata. Gambar 3.9 adalah hasil tampilan dari menu About Me.



Gambar 3.9 Desain Menu About Me

3.5 Analisis Algoritma Beaufort

Algoritma Beaufort bekerja dengan cara menambahkan karakter dengan kunci. Tiap karakter pada plaintext akan dikonversi kan menjadi nilai desimal menurut tabel ASCII. Proses enkripsi akan menambahkan nilai desimal plaintext tersebut dengan nilai desimal kunci. Proses dekripsi akan mengurangkan nilai desimal dari ciphertext dengan kunci. Berikut ini akan dijelaskan bagaimana proses enkripsi dan dekripsi algoritma Beaufort.

PT	U	N	P	A	B		I	N	D	O	N	E	S	I	A
K	M	E	D	A	N	M	E	D	A	N	M	E	D	A	N
ASCII PT	85	78	80	65	66	32	73	78	68	79	78	69	83	73	65
ASCII K	77	69	68	65	78	77	69	68	65	78	77	69	68	65	78

ASCII CT	162	147	148	130	144	109	142	146	133	157	155	138	151	138	143
CT	¢	“	”	,	•	m	Ž	’	…	•	›	Š	—	Š	•
ASCII PT2	85	78	80	65	66	32	73	78	68	79	78	69	83	73	65
PT2	U	N	P	A	B		I	N	D	O	N	E	S	I	A

Keterangan:

PT = Plaintext

K = Kunci

ASCII PT = Nilai desimal plaintext sesuai tabel ASCII

ASCII K = Nilai desimal kunci sesuai tabel ASCII

ASCII CT = Hasil enkripsi dari ASCII PT ditambah ASCII K

CT = Karakter ASCII CT menurut tabel ASCII

ASCII PT2 = Hasil dekripsi dari ASCII CT dikurang ASCII K

PT2 = Karakter ASCII PT2 menurut tabel ASCII

Algoritma Beufort memiliki input dengan dua parameter yaitu plaintext dan kunci. Plaintext diisi dengan teks asli yang dapat dibaca secara sempurna. Kunci adalah password yang digunakan untuk proses enkripsi dan dekripsi. Proses dari algoritma Beaufort adalah dengan melakukan pertambahan karakter plaintext dengan kunci setelah dikonversi ke desimal sesuai dengan tabel ASCII. Hasil enkripsi adalah hasil pertambahan plaintext dengan dan hasil dekripsi adalah hasil pengurangan ciphertext dengan kunci. Output dari algoritma ini adalah cipertext dan plaintext.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas hasil dan pembahasan dari hasil perancangan sistem yang sudah penulis lakukan. Pembahasan yang akan dilakukan terbagi dua yaitu sistem, implementasi dan pengujian manual. Pada sistem akan dijelaskan kebutuhan alat bantu dalam mendukung penelitian ini. Implementsi adalah program aplikasi yang telah dibuat dan bagaimana cara penggunaan program aplikasi tersebut dalam melaksanakan pengujian algoritma Beaufort Cipher. Pengujian manual akan membuktikan kebenaran hasil dari program aplikasi apakah sudah sesuai dengan perhitungan Beaufort secara manual.

4.1 Spesifikasi Sistem

Spesifikasi sistem menjelaskan persyaratan operasional dan kinerja suatu sistem, seperti komputer. Ini dianggap sebagai dokumen tingkat tinggi yang menentukan fungsi global. Spesifikasi sistem membantu untuk menentukan pedoman operasional dan kinerja untuk suatu program aplikasi. Spesifikasi sistem dapat menguraikan bagaimana sistem diharapkan untuk melakukan, dan apa yang mungkin termasuk. Spesifikasi utama dapat mencakup definisi antarmuka, aturan desain dokumen, dan area fungsional. Spesifikasi dapat menentukan akses keamanan. Spesifikasi pada penelitian ini terdiri dari perangkat keras dan lunak.

4.1.1 Spesifikasi Perangkat Keras

Perangkat keras sangat dibutuhkan dalam melaksanakan pembuatan program aplikasi algoritma Beaufort Cipher. Hal ini sebagai sarana pendukung utama. Tabel 4.1 adalah spesifikasi perangkat keras yang digunakan pada penelitian ini.

Tabel 4.1 Spesifikasi perangkat keras

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

4.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak memiliki tujuan, deskripsi kebutuhan dan persiapan validasi aplikasi perangkat lunak. Deskripsi kebutuhan memunculkan file spesifikasi aplikasi perangkat lunak. Perangkat lunak merupakan kode program yang sudah dikemas menjadi beberapa program aplikasi pendukung dalam membantu pembuatan program aplikasi algoritma Beaufort Cipher. Tabel 4.2 adalah spesifikasi perangkat lunak yang digunakan pada penelitian ini.

Tabel 4.2 Spesifikasi perangkat lunak

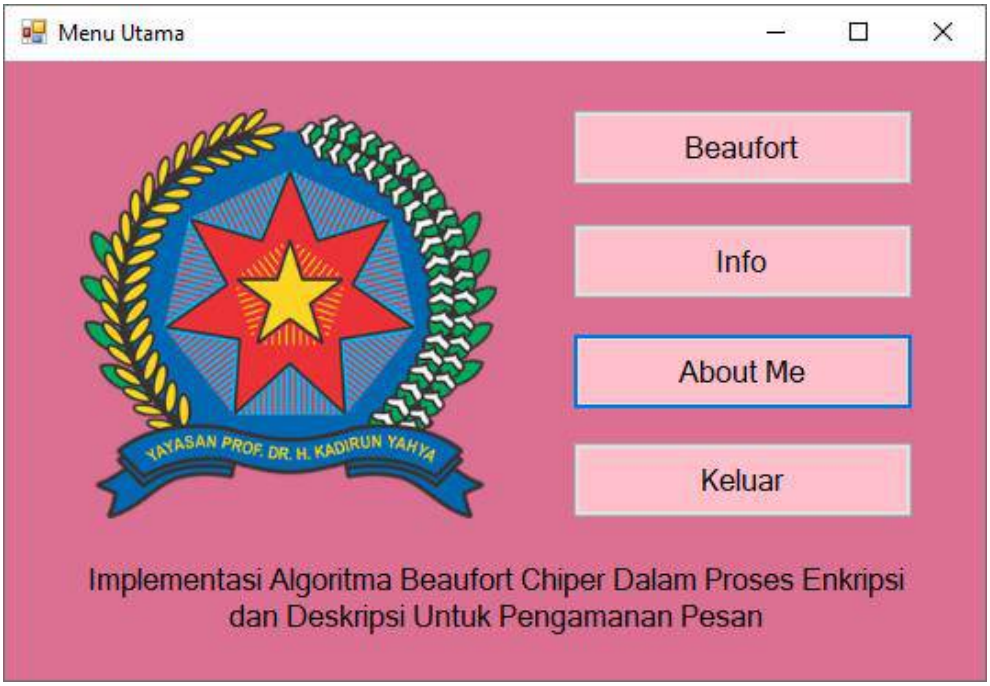
No.	Komponen	Spesifikasi
1	Sistem Operasi	Windows 10 64 Bit
2	IDE Pemrograman	Microsoft Visual Basic.NET 2010
3	Tangkap Gambar	Snipping Tool
4	Data Editor	Microsoft Excel

4.2 Implementasi Interface

Implementasi inferface adalah penerapan antarmuka yang telah dibuat berdasarkan perancangan yang dilakukan pada bab sebelumnya. Program aplikasi terbagi menjadi beberapa halaman menu. Tiap menu memiliki fungsi dan tugas masing-masing.

4.2.1 Halaman Menu Utama

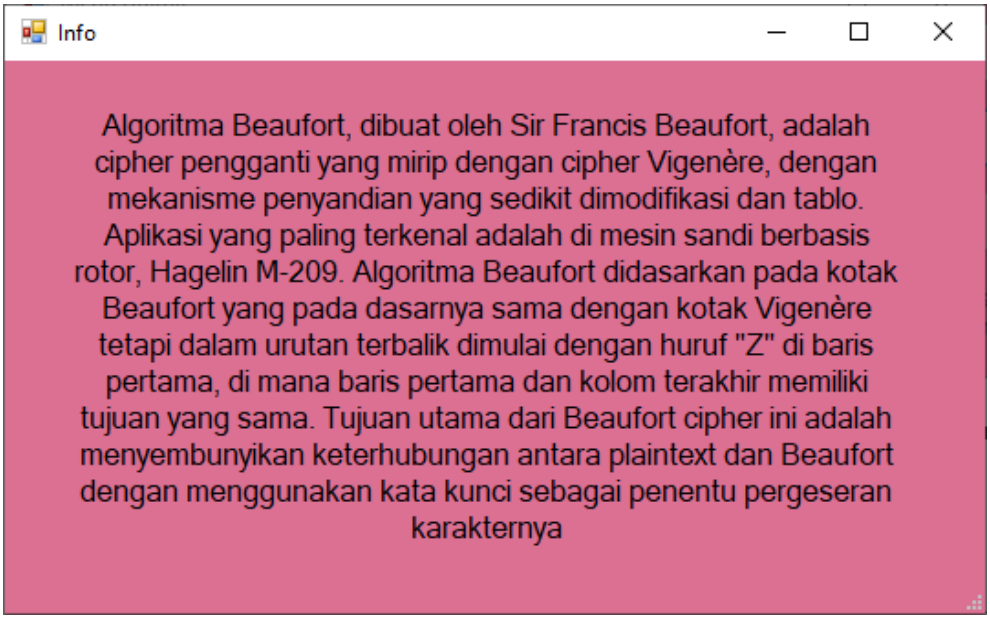
Halaman Menu Utama merupakan halaman utama sebuah program aplikasi. Halaman ini berfungsi untuk mengakses menu-menu lainnya. Pada halaman Menu Utama, ada beberapa tombol navigasi yang berfungi untuk membuka submenu dan satu buah tombol untuk keluar dari program aplikasi tersebut. Gambar 4.1 adalah hasil tampilan menu utama.



Gambar 4.1 Halaman Menu Utama

4.2.2 Halaman Info

Halaman info adalah menu yang menampilkan sejarah singkat tentang algoritma Beaufort Cipher. Gambar 4.2 adalah hasil tampilan dari halaman info.



Gambar 4.2 Halaman Info

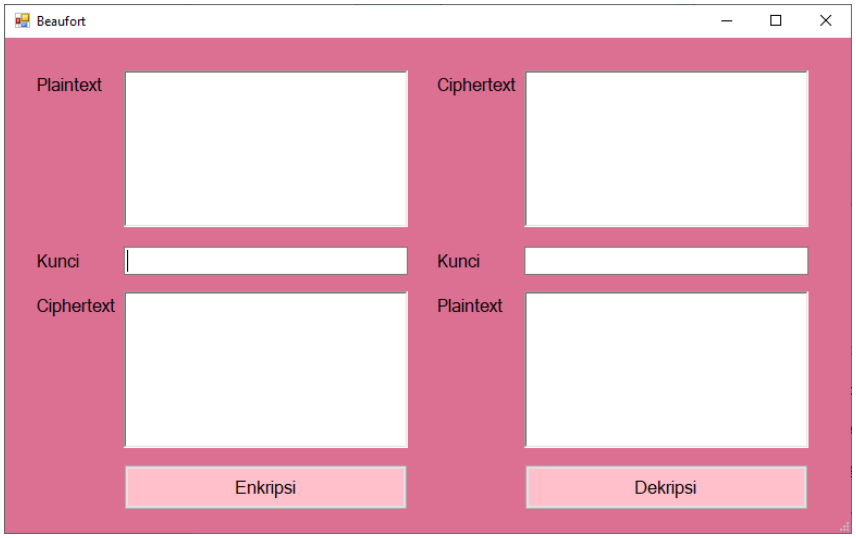
4.2.3 Halaman About Me

Halaman About Me menampilkan status dari penulis dalam lingkungan universitas. Form ini memiliki sebuah objek label dan picturebox. Pada halaman ini, akan ditampilkan sebuah logo universitas yaitu logo Universitas Pembangunan Panca Budi dan beberapa untaian kata yang berhubungan dengan biodata penulis. Gambar 4.3 adalah tampilan dari halaman About Me.

Gambar 4.3 Halaman About Me

4.2.4 Halaman Beaufort Cipher

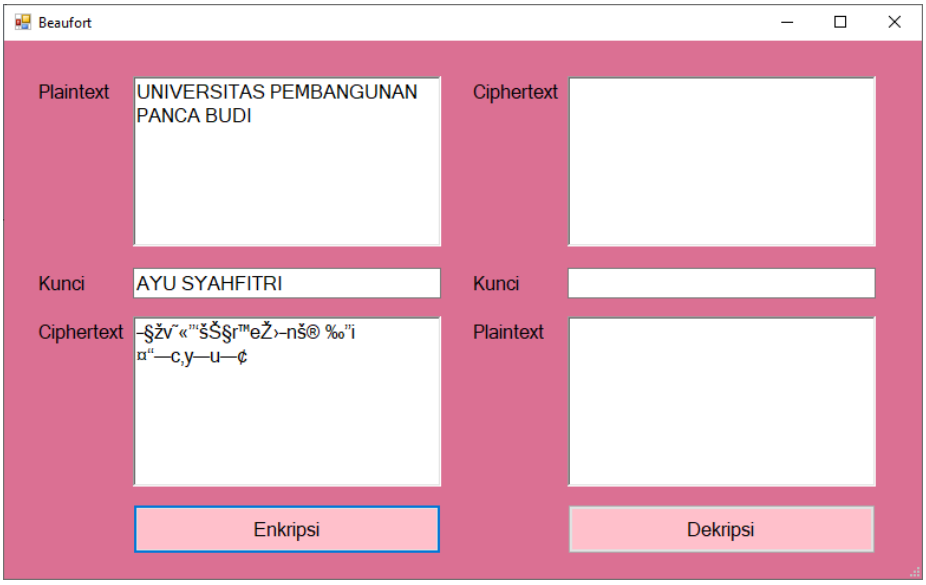
Halaman ini adalah bagian algoritma Beaufort Cipher dari program aplikasi. Halaman ini berupa proses kriptografi substitusi untuk mengubah plaintext ke ciphertext dan juga sebaliknya chipertext ke plaintext. Bagian ini terdiri beberapa objek textbox yaitu dua buah textbox plaintext, sebuah textbox kunci dan dua buah textbox ciphertext yang termasuk dari bagian input dan output. Proses enkripsi dan dekripsi dilakukan dengan menekan objek tombol enkripsi dan dekripsi. Gambar 4.4 adalah hasil tampilan dari halaman Beaufort Cipher.



Gambar 4.4 Halaman kriptografi algoritma Beaufort

4.2.5 Proses Enkripsi

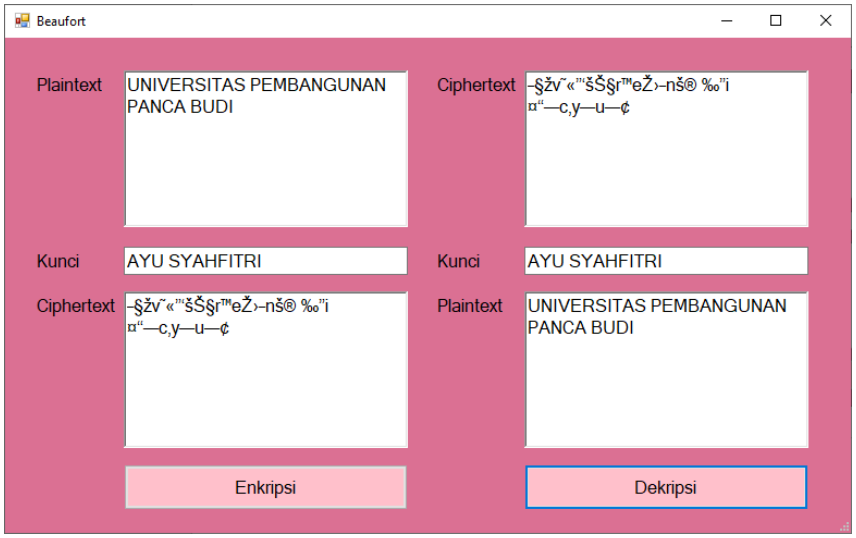
Bagian ini berisi tentang hasil proses enkripsi dengan plaintext dan kunci yang telah dimasukkan pada textbox. Plaintext dan Kunci adalah dua bagian yang harus dipenuhi agar ciphertext dapat ditentukan. Blok kunci dibentuk sesuai dengan jumlah panjang Kunci dan Plaintext. Blok kunci harus memiliki panjang yang sama dengan plaintext agar setiap karakter pada plaintext dapat dilakukan proses enkripsi. Gambar 4.5 adalah tampilan dari hasil perhitungan proses enkripsi algoritma Beaufort Cipher.



Gambar 4.5 Halaman enkripsi algoritma Beaufort Cipher

4.2.6 Proses Dekripsi

Setelah proses enkripsi, ciphertext dihasilkan merupakan pesan terenkripsi. Pesan ini tidak dapat difahami oleh pembaca sehingga pesan ini harus dikembalikan ke bentuk plaintext agar dapat dikenali. Dekripsi harus menggunakan kunci yang digunakan pada saat enkripsi. Blok kunci kembali dibentuk agar setiap karakter pada ciphertext dapat dilakukan proses dekripsi. Hasil dekripsi akan salah jika ada satu kunci yang berbeda sehingga plaintext yang dihasilkan berbeda dengan pesan aslinya. Gambar 4.6 adalah tampilan dari hasil perhitungan proses dekripsi algoritma Beaufort Cipher.



Gambar 4.6 Halaman dekripsi algoritma Beaufort Cipher

4.3 Pengujian Manual

Pengujian manual dilaksanakan untuk melihat kemampuan program aplikasi dalam menghasilkan program kriptografi. Pada pengujian ini akan dilihat perhitungan yang dilakukan secara manual dengan hasil yang ditampilkan pada program aplikasi algoritma Beaufort tersebut. Proses yang dilakukan terdiri dari dua tahap, yaitu enkripsi dan proses dekripsi. Berikut ini adalah penjelasan dan perhitungan pada algoritma Beaufort Cipher dengan memberikan parameter input.

Parameter Input

Plaintext = UNIVERSITAS PEMBANGUNAN PANCA BUDI

Kunci Beaufort = AYU SYAHFITRI

Hasil Enkripsi

Tabel 4.3 Hasil enkripsi pengujian manual

PT	PT ASCII	KUNCI	KUNCI ASCII	CT ASCII	CT
U	85	A	65	150	–
N	78	Y	89	167	§
I	73	U	85	158	ž
V	86		32	118	v
E	69	S	83	152	˜
R	82	Y	89	171	«
S	83	A	65	148	”
I	73	H	72	145	‘
T	84	F	70	154	š
A	65	I	73	138	Š
S	83	T	84	167	§
	32	R	82	114	r
P	80	I	73	153	™
E	69		32	101	e
M	77	A	65	142	Ž
B	66	Y	89	155	›
A	65	U	85	150	–
N	78		32	110	n
G	71	S	83	154	š
U	85	Y	89	174	®
N	78	A	65	143	•
A	65	H	72	137	‰
N	78	F	70	148	”
	32	I	73	105	i
P	80	T	84	164	¤
A	65	R	82	147	“
N	78	I	73	151	—
C	67		32	99	c
A	65	A	65	130	,
	32	Y	89	121	y
B	66	U	85	151	—
U	85		32	117	u
D	68	S	83	151	—
I	73	Y	89	162	¢

Hasil Dekripsi

Tabel 4.4 Hasil Dekripsi Pengujian Manual

CT	CT ASCII	KUNCI	KUNCI ASCII	PT ASCII	PT
–	150	A	65	85	U
§	167	Y	89	78	N
ž	158	U	85	73	I
v	118		32	86	V
˜	152	S	83	69	E
«	171	Y	89	82	R
”	148	A	65	83	S
‘	145	H	72	73	I
š	154	F	70	84	T
Š	138	I	73	65	A
§	167	T	84	83	S
r	114	R	82	32	
™	153	I	73	80	P
e	101		32	69	E
Ž	142	A	65	77	M
›	155	Y	89	66	B
–	150	U	85	65	A
n	110		32	78	N
š	154	S	83	71	G
®	174	Y	89	85	U
•	143	A	65	78	N
‰	137	H	72	65	A
”	148	F	70	78	N
i	105	I	73	32	
¤	164	T	84	80	P
“	147	R	82	65	A
—	151	I	73	78	N
c	99		32	67	C
,	130	A	65	65	A
y	121	Y	89	32	
—	151	U	85	66	B
u	117		32	85	U
—	151	S	83	68	D
¢	162	Y	89	73	I

BAB V

PENUTUP

5.1 Kesimpulan

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

1. Beaufort Cipher bekerja dengan cara melakukan pergeseran pada karakter.
2. Beaufort Cipher memiliki kunci yang dapat ditentukan sesuai dengan jumlah kunci yang diinginkan.
3. Beaufort Cipher harus menggunakan modulo agar karakter hasil enkripsi tidak melewati batas karakter pada tabel ASCII.

5.2 Saran

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

1. Sebaiknya algoritma Beaufort Cipher dapat digunakan secara online.
2. Beaufort Cipher akan lebih jika dikombinasikan dengan algoritma lain agar meningkat keamanan.
3. Algoritma Beaufort dapat dipecahkan dengan percobaan pergeseran. Sebaiknya kunci dapat dikembangkan menjadi lebih baik.

DAFTAR PUSTAKA

Ayushi, M. (2010). A Symmetric Key Cryptographic Algorithm. *International Journal of Computer Applications*, *1*(15), 1–6. https://doi.org/10.5120/331- 502

Bishop, D. (2002). *Introduction to Cryptography*. Jones and Batrlet Publisher. Bruen, A. A., & Forcinito, M. A. (2005). *Cryptography, Information Theory, and*

Error Correction: A Handbook for the 21st Century. New Jersey: John Wiley & Sons.

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.

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

Jogiyanto, H. M. (2006). *Analisis Dan Desain Sistem Informasi, Pendekatan Terstruktur Teori Dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Offset.

Kelsey, J. (2002). *Compression and Information Leakage of Plaintext*. https://doi.org/10.1007/3-540-45661-9_21

Kurniawan, T. A. (2018). Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *5*(1), 77. https://doi.org/10.25126/jtiik.201851610

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

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

Leighton, A. C. (1969). Secret Communication among the Greeks and Romans. *Technology and Culture*, *10*(2), 139. https://doi.org/10.2307/3101474

Martin, K. M. (2012). Everyday Cryptography: *Fundamental Principles and Applications*. London: Oxford University Press.

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

Oppliger, R. (2005). *Contemporary Cryptography*. Boca Raton, Florida: CRC Press.

Pabokory, F. N., Astuti, I. F., & Kridalaksana, A. H. (2015). Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *10*, 22. https://doi.org/10.30872/jim.v10i1.23

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

Pratama, G. M., & Tamatjita, E. N. (2015). Modifikasi algoritma vigenere cipher menggunakan metode catalan number dan double columnar transposition. *Compiler*, *4*(1), 31–40.

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.

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

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

S., G., L. Ribeiro, A. R., & David, E. (2012). Asymmetric Encryption in Wireless Sensor Networks. In *Wireless Sensor Networks - Technology and Protocols*. https://doi.org/10.5772/48464

Spencer, M. (2019). ASCII Table and Description. Retrieved November 23, 2019, from ASCII Table website: http://www.asciitable.com/

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

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

Sun, Y., Zhang, J., Xiong, Y., & Zhu, G. (2014). Data Security and Privacy in Cloud Computing. *International Journal of Distributed Sensor Networks*, *10*(7), 190903. https://doi.org/10.1155/2014/190903

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 Technopediawebsite:https://www.techopedia.com/definition/3243/unifie d-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

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