APLIKASI ENKRIPSI DAN DEKRIPSI UNTUK MENGAMANKAN PESAN RAHASIA DENGAN METODE GRONSFELD CIPHER

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

SKRIPSI

OLEH:

NAMA	:	**NURAINUN HARAHAP**
NPM	:	**1514370059**
PROGRAM STUDI	:	**SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

NURAINUN HARAHAP
Aplikasi Enkripsi dan Dekripsi Untuk Mengamankan Pesan Rahasia Dengan Metode Gronsfeld Cipher
2020

Pesan merupakan informasi yang akan dipertukarkan dengan orang lain. Pesan akan dikirimkan melalui jaringan komputer. Keamanan pesan belum tentu terjaga dalam proses pengiriman. Teknik pengamanan pesan dapat dilakukan dengan teknik kriptografi. Algoritma Gronsfeld cipher merupakan algoritma yang sangat mudah dilakukan dalam mengamankan pesan. Algoritma ini bekerja dengan cara menggeser plaintext sebesar kunci yang telah ditentukan. Algoritma ini bekerja dengan cepat dan akurat. Kunci yang digunakan adalah angka. Hasil enkripsi merupakan karakter-karakter yang tetap berada pada tabel ASCII. Hasil enkripsi yang melebihi 255 akan dilakukan proses modulo untuk memutar karakter tersebut agar tetap berada pada tabel ASCII. Dengan menerapkan algoritma Gronsfeld pada pengiriman pesan, keamanan pesan akan lebih terjamin.

Kata Kunci: algoritma, keamanan, Gronsfeld, enkripsi, dekripsi

BAB I

PENDAHULUAN

1.1 Latar Belakang

Keamanan komputer sangat diperlukan dalam menjaga kerahasiaan data. Setiap orang memiliki informasi yang akan dikirimkan melalui jaringan komputer. Dalam melakukan pengiriman, sering kali pesan yang dikirim mengalami pencurian. Data-data yang dikirimkan akan bocor ke tangan orang-orang yang tidak bertanggung jawab. Hal ini menyebabkan kerugian bagi pemilik data tersebut. Data yang dicuri akan disalahgunakan dan akan digunakan dalam mencari keuntungan dari pemilik data tersebut.

Dalam mengamankan data, diperlukan suatu teknik yang dapat memastikan data yang dikirimkan pada jaringan komputer akan aman dari serangan pihak-pihak yang tidak bertanggung jawab. Teknik kriptografi adalah salah satu yang dapat dilakukan dalam menjaga kerahasiaan data. Setiap data yang akan dikirim akan terlebih dahulu dienkrip sehingga data tersebut aman dari pencurian data. Ada banyak teknik yang dapat digunakan pada kriptografi ini. Teknik ini juga merupakan salah satu solusi yang dapat digunakan dalam pengamanan pesan sebelum dikirim. Akan tetapi, pengembangan strategi kriptografi pada basis data membutuhkan banyak pertimbangan data, mencakup analisis lingkungan, desain solusi dan persoalan persoalan yang ditemui dalam menentukan desain pengamanan basis data. Pencurian data baik itu terhadap komputer yang terhubung pada suatu jaringan maupun tidak, sudah menjadi hal yang sering terdengar dan tidak asing

lagi bagi kalangan intelektual khususnya dan masyarakat luas pada umumnya. Hal ini menyebabkan pengguna basis data harus menemukan cara untuk mengamankan data tanpa campur tangan orang-orang yang tidak bertanggung jawab. Pengamanan pada pesan dengan teknik kriptografi sangat dibutuhkan sehingga pesan harus dengan enkripsi dan dekripsi pada saat pertukaran data.

Ada banyak algoritma kriptografi yang ada. Salah satu adalah dengan menggunakan teknik *cipher* substitusi polyalphabetic yang paling sederhana. Algoritma yang digunakan adalah Gronsfeld Cipher. Algoritma ini mirip dengan dari Vigenere Cipher. Gronsfeld Cipher berasal dari nama penemunya, yaitu Johann Franz Graf Gronsfeld-Bronkhorst. Dia adalah panglima kekaisaran dalam pemberontakan nasional Bavarian 1705-1706. Gronsfeld identik dengan Vigenere. Perbedaan yang mendasar adalah kunci Gronsfeld menggunakan pergeseran angka. Kunci yang digunakan pada algoritma Gronsfeld Cipher adalah numerik. Cipher Gronsfeld sendiri umumnya lebih banyak digunakan di Jerman dan beberapa negara lain di Eropa.

Penelitian ini bertujuan untuk mengamankan pesan dengan menggunakan algoritma Gronsfeld Cipher. Berdasarkan latar belakang yang sudah dipaparkan, penulis mengambil penelitian dengan judul "**APLIKASI ENKRIPSI DAN DEKRIPSI UNTUK MENGAMANKAN PESAN RAHASIA DENGAN METODE GRONSFELD CIPHER**".

1.2 Rumusan Masalah

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

1. Bagaimana merancang aplikasi pengamanan isi di dalam pesan menggunakan metode Gronsfeld cipher?
2. Bagaimana mengetahui cara kerja proses enkripsi dan dekripsi algoritma Gronsfeld cipher?
3. Bagaimana menentukan pergeseran kunci pada algoritma Gronsfeld cipher?

1.3 Batasan Masalah

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

1. Kunci enkripsi yang digunakan pada algoritma Gronsfeld hanya menggunakan angka dengan rentang 1 – 100.
2. Pesan yang digunakan adalah bertipe teks yang diinputkan langsung pada textbox.
3. Program aplikasi yang digunakan adalah menggunakan Microsoft Visual Basic.Net 2010.

1.4 Tujuan Penelitian

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

1. Untuk merancang aplikasi pengamanan isi di dalam pesan menggunakan metode Gronsfeld cipher.

2. Untuk mengetahui cara kerja proses enkripsi dan dekripsi algoritma Gronsfeld cipher.
3. Untuk menentukan pergeseran kunci pada algoritma Gronsfeld cipher.

1.5 Manfaat Penelitian

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

1. Meningkatkan keamanan pesan sebelum dikirimkan.
2. Memberikan pengetahuan tentang cara kerja algoritma Gronsfeld Cipher.

BAB II

LANDASAN TEORI

2.1 Pengertian Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanankan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan (Sopyan, Supriyadi, & Kurniadi, 2016).

Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer.

2.2 Data

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi (Jogiyanto, 2006). Kegiatan suatu perusahaan, misalnya transaksi penjualan oleh sejumlah *salesman*, dihasilkan sejumlah faktur-faktur yang

merupakan data dari penjualan pada suatu periode tertentu. Faktur-faktur penjualan tersebut masih belum dilaporkan secara terperinci kepada manajemen. Untuk keperluan pengambilan keputusan, maka faktur-faktur tersebut perlu diolah lebih lanjut untuk menjadi suatu informasi (Sun, Zhang, Xiong, & Zhu, 2014).

2.2.1 Bagaimana Data Disimpan

Komputer mewakili data, termasuk video, gambar, suara dan teks, sebagai nilai biner menggunakan pola hanya dua angka: 1 dan 0. Sedikit adalah unit data terkecil dan hanya mewakili nilai tunggal. Satu byte terdiri dari delapan digit biner. Penyimpanan dan memori diukur dalam megabit dan gigabit.

Unit-unit pengukuran data terus bertambah seiring dengan meningkatnya jumlah data yang dikumpulkan dan disimpan. Istilah "brontobyte" yang relatif baru, misalnya, adalah penyimpanan data yang setara dengan 10 hingga 27 byte. Data dapat disimpan dalam format file, seperti pada sistem mainframe menggunakan ISAM dan VSAM. Format file lain untuk penyimpanan, konversi, dan pemrosesan data termasuk nilai yang dipisah koma. Format ini terus menemukan kegunaan di berbagai jenis mesin, bahkan ketika pendekatan yang lebih berorientasi data terstruktur memperoleh pijakan dalam komputasi perusahaan. Spesialisasi yang lebih besar dikembangkan sebagai basis data, sistem manajemen basis data, dan kemudian teknologi basis data relasional muncul untuk mengatur informasi (Zhang et al., 2009).

2.2.2 Jenis data

Pertumbuhan web dan telepon pintar selama dekade terakhir menyebabkan peningkatan dalam penciptaan data digital. Data sekarang termasuk informasi teks, audio dan video, serta catatan aktivitas log dan web. Banyak dari itu adalah data yang tidak terstruktur.

Istilah big data telah digunakan untuk menggambarkan data dalam kisaran petabyte atau lebih besar. Tulisan singkat menggambarkan data besar dengan 3V - volume, variasi, dan kecepatan. Ketika e-commerce berbasis web telah menyebar, model bisnis berbasis data besar telah berevolusi yang memperlakukan data sebagai aset. Tren semacam itu juga telah menimbulkan keasyikan yang lebih besar dengan penggunaan sosial data dan privasi data.

Data memiliki makna di luar penggunaannya dalam aplikasi komputasi yang berorientasi pada pemrosesan data. Misalnya, dalam interkoneksi komponen elektronik dan komunikasi jaringan, istilah data sering dibedakan dari "informasi kontrol," "bit kontrol," dan istilah serupa untuk mengidentifikasi konten utama dari unit transmisi. Selain itu, dalam sains, istilah data digunakan untuk menggambarkan kumpulan fakta. Itu juga terjadi di bidang-bidang seperti keuangan, pemasaran, demografi dan kesehatan.

2.2.3 Pengelolaan dan Penggunaan Data

Dengan semakin banyaknya data dalam organisasi, penekanan tambahan telah ditempatkan pada memastikan kualitas data dengan mengurangi duplikasi dan menjamin yang paling akurat, catatan saat ini digunakan. Banyak langkah yang

terlibat dengan manajemen data modern termasuk pembersihan data, serta mengekstrak, mengubah dan memuat (ETL) proses untuk mengintegrasikan data. Data untuk diproses telah dilengkapi dengan metadata, kadang-kadang disebut sebagai "data tentang data," yang membantu administrator dan pengguna memahami database dan data lainnya.

Analisis yang menggabungkan data terstruktur dan tidak terstruktur menjadi bermanfaat, karena organisasi berupaya memanfaatkan informasi tersebut. Sistem untuk analitik semacam itu semakin berupaya untuk kinerja waktu-nyata, sehingga mereka dibangun untuk menangani data yang masuk yang dikonsumsi dengan tingkat konsumsi tinggi, dan untuk memproses aliran data untuk penggunaan langsung dalam operasi.

Seiring waktu, gagasan basis data untuk operasi dan transaksi telah diperluas ke basis data untuk pelaporan dan analitik data prediktif. Contoh utama adalah gudang data, yang dioptimalkan untuk memproses pertanyaan tentang operasi untuk analis bisnis dan pemimpin bisnis. Meningkatnya penekanan pada menemukan pola dan memprediksi hasil bisnis telah mengarah pada pengembangan teknik penambangan data (Barone, Williams, & Micklos, 2017).

2.3 Logika dan Algoritma

Pengertian algoritma sangat lekat dengan kata logika, yaitu kemampuan seorang manusia untuk berfikir dengan akal tentang suatu permasalahan menghasilkan sebuah kebenaran, dibuktikan dan dapat diterima akal, logika

seringkali dihubungkan dengan kecerdasan, seseorang yang mampu berlogika dengan baik sering orang menyebutnya sebagai pribadi yang cerdas.

Logika identik dengan masuk akal dan penalaran.Penalaran adalah salah satu bentuk pemikiran.Pemikiran adalah pengetahuan tak langsung yang didasarkan pada pernyataan langsung pemikiran mungkin benar dan mungkin juga tak benar.Definisi logika sangat sederhana yaitu ilmu yang memberikan prinsip-prinsip yang harus diikuti agar dapat berfikir valid menurut aturan yang berlaku.Pelajaran logika menimbulkan kesadaran untuk menggunakan prinsip-prinsip untuk berfikir secara sistematis.Logika berasal dari dari bahasa Yunani yaitu LOGOS yang berarti ilmu. Logika dapat diartikan ilmu yang mengajarkan cara berpikir untuk melakukan kegiatan dengan tujuan tertentu. Algoritma berasal dari nama seorang Ilmuwan Arab yang bernama Abu JafarMuhammad Ibnu Musa Al Khuwarizmi penulis buku berjudul Al Jabar Wal Muqabala. Kata Al Khuwarizmi dibaca orang barat menjadi Algorism yang kemudian lambat laun menjadi Algorithm diserap dalam bahasa Indonesia menjadi Algoritma.

Logika identik dengan masuk akal dan penalaran.Penalaran adalah salah satu bentuk pemikiran. Pemikiran adalah pengetahuan tak langsung yang didasarkan pada pernyataan langsung pemikiran mungkin benar dan mungkin juga tak benar.Definisi logika sangat sederhana yaitu ilmu yang memberikan prinsip-prinsip yang harus diikuti.

2.4 Kriptografi

Menurut M. Miftakhul Amin, kriptografi (*Cryptography*) berasal dari bahasa yunani terdiri dari dua suku kata yaitu kripto dan graphia. Kripto artinya menyembunyikan sedangkan graphia artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Tetapi tidak semua aspek keamanan informasi dapat diselesaikan dengan kriptografi (Amin, 2016). Adapun istilah-istilah yang sering digunakan dalam ilmu kriptografi diantara sebagai berikut:

1. *Plaintext*

   Plaintext merupakan pesan asli yang belum disandikan atau informasi yang ingin dikirimkan atau dijaga keamanannya.

2. *Ciphertext*

   Ciphertext merupakan pesan yang telah disandikan (dikodekan) sehingga siap untuk dikirimkan.

3. Enkripsi

   Enkripsi merupakan proses yang dilakukan untuk menyandikan plaintext menjadi ciphertext dengan tujuan pesan tersebut tidak dapat dibaca oleh pihak yang tidak berwenang.

4. Deskripsi

   Deskripsi merupakan proses yang dilakukan untuk memperoleh kembali plaintext dari ciphertext.

5. Kunci

Kunci yang dimaksud disini adalah kunci yang dipakai untuk melakukan dekripsi dan enkripsi. Kunci terbagi menjadi dua bagian, diantaranya yaitu kunci pribadi (*private key*) dan kunci umum (*public key*).

6. Kriptosistem

   Kriptosistem merupakan sistem yang dirancang untuk mengamankan suatu sistem informasi dengan memanfaatkan kriptografi.

7. Kriptanalisis

   Kriptanalisis merupakan suatu ilmu untuk mendapatkan plaintext tanpa harus mengetahui kunci secara wajar.

Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan. Ketika suatu pesan dikirim dari suatu tempat ke tempat lain, isi pesan tersebut mungkin dapat disadap oleh pihak lain yang tidak berhak untuk mengetahui isi pesan tersebut. Untuk menjaga pesan maka pesan tersebut dapat diubah menjadi sebuah kode yang tidak dapat dimengerti pihak lain.

2.4.1 Sejarah Kriptografi

Sejarah kriptografi sebagian besar merupakan sejarah kriptografi klasik, yaitu metode enkripsi yang menggunakan kertas dan pensil atau mungkin dengan bantuan alat mekanik sederhana. Secara umum algoritma kriptografi klasik dikelompokkan menjadi dua kategori, yaitu algoritma transposisi (*transposition cipher*) dan algoritma substitusi (*substitution cipher*). *Cipher* transposisi mengubah susunan huruf-huruf di dalam pesan, sedangkan *cipher* substitusi mengganti setiap

huruf atau kelompok huruf dengan sebuah huruf atau kelompok huruf lain (Pabokory, Astuti, & Kridalaksana, 2015).

2.4.2 Tujuan Kriptografi

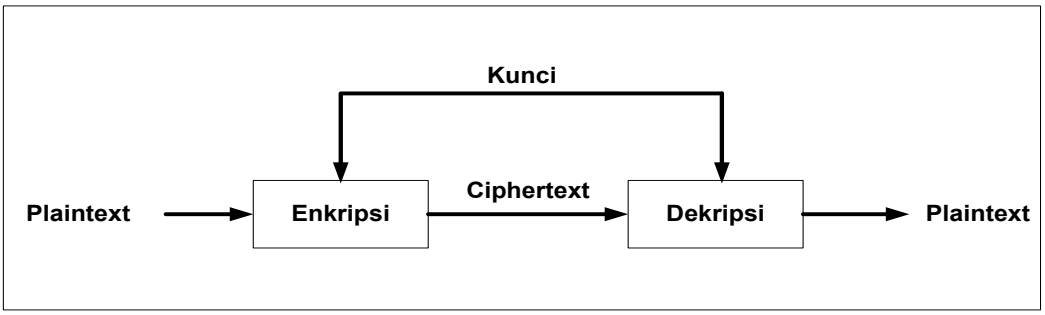
Dari paparan awal dapat dirangkumkan bahwa kriptografi bertujuan untuk member layanan keamanan. Yang dinamakan aspek-aspek keamanan:

1. Kerahasiaan (*confidentiality*) adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
2. Integritas data (*data integrity*) adalah layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman.
3. Otentikasi (*authentication*) adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi(*user autehentication*).
4. *Non-repudiation*adalah layanan untuk menjaga entitas yang berkomunikasi melakukan penyangkalan (Pabokory, 2015: 22).

2.4.3 Kriptografi Simetris

Kriptografi simetris adalah teknik kriptografi dimana kunci yang digunakan pada proses enkripsi dan dekripsi adalah kunci yang sama. Dalam kriptografi kunci simetris dapat diasumsikan bahwa si penerima dan pengirim pesan telah terlebih dahulu berbagi kunci sebelum pesan dikirimkan.Keamanan dari sistem ini terletak pada kerahasiaan kuncinya.

Pada umumnya yang termasuk ke dalam kriptografi simetris ini beroperasi dalam mode blok (*block cipher*), yaitu setiap kali proses enkripsi atau dekripsi dilakukan terhadap satu blok data (yang berukuran tertentu), atau beroperasi dalam mode aliran (*stream cipher*), yaitu setiap kali enkripsi atau dekripsi dilakukan terhadap satu bit atau satu byte data. Contoh algoritma simetris, yaitu : Trithemius, Double Transposition Cipher,DES (Data Encryption Standard), AES (Advanced Encryption Standard). Gambar 2.1 adalah skema algoritma simetris.



Gambar 2.1 Skema kriptografi simetris
Sumber: (Putri, Setyorini, & Rahayani, 2018)

Kelebihan kriptografi simetris adalah:

1. Proses enkripsi atau dekripsi kriptografi simetris membutuhkan waktu yang singkat.
2. Ukuran kunci simetris *relative* lebih pendek.
3. Otentikasi pengiriman pesan langsung dari *ciphertext* yang diterima, karena kunci hanya diketahui oleh penerima dan pengirim saja.

Kelemahan kriptografi simetris antara lain:

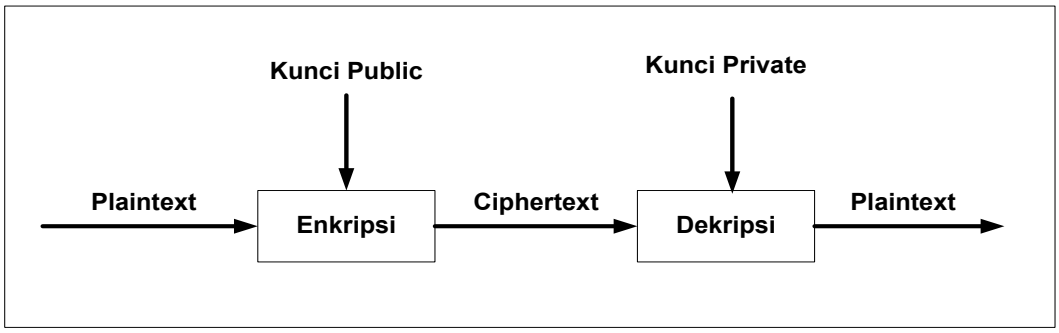
1. Kunci simetris harus dikirim melalui saluran komunikasi yang aman, dan kedua entitas yang berkomunikasi harus menjaga kerahasiaan kunci.
2. Kunci harus sering diubah, setiap kali melaksanakan komunikasi. Apabila kunci tersebut hilang atau lupa, maka pesan tersebut tidak dapat dibuka.

2.4.4 Kriptografi Asimetris

Berbeda dengan kriptografi kunci simetris, kriptografi kunci public memiliki dua buah kunci yang berbeda pada proses enkripsi dan dekripsinya. Dimana kunci yang digunakan untuk proses enkripsi atau sering disebut *public key* dan dekripsi atau sering disebut *private key* menggunakan kunci yang berbeda. Entitas pengirim akan mengenkripsi dengan menggunakan kunci *public*, sedangkan entitas penerima mendekripsi menggunakan kunci *private* (Kamil, 2016).

Contoh algoritma asimetris, yaitu RSA (*Riverst Shamir Adleman)*, Knapsack, Rabin, ElGamal (Ayushi, 2010) (S., L. Ribeiro, & David, 2012). Pada algoritma tak simetri kunci terbagi menjadi dua bagian:

1. Kunci umum (*public key*) adalah kunci yang dapat dan boleh diketahui oleh semua orang.
2. Kunci pribadi (*private key*) adalah kunci yang hanya dapat diketahui penerima dan bersifat rahasia.



Gambar 2.2 Skema kriptografi asimetris
Sumber: (Putri et al., 2018)

Kelebihan kriptografi asimetris adalah:

1. Hanya kunci *private* yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi. Tidak ada kebutuhan mengirim kunci *private* sebagaimana kunci simetri.
2. Pasangan kunci *private* dan kunci *public* tidak perlu diubah dalam jangka waktu yang sangat lama.
3. Dapat digunakan dalam pengaman pengiriman kunci simetris.

Kelemahan kriptografi asimetris adalah:

1. Proses enkripsi dan dekripsi umumnya lebih lambat dari algortima simetri, karena menggunakan bilangan yang besar dan operasi bilangan yang besar.
2. Ukuran *ciphertext* lebih besar dari *plaintext*.
3. Ukuran kunci relatif lebih besar daripada ukuran kunci simetris.

2.5 Enkripsi

Enkripsi adalah proses penyandian *plaintext* menjadi *ciphertext*, atau pengubahan data menjadi bentuk rahasia. Proses *enkripsi algoritma AES* terdiri dari 4 jenis *transformasi bytes*, yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pada awal proses *enkripsi*, input yang telah dicopykan ke dalam *state* akan mengalami *transformasi byte AddRoundKey*. Setelah itu, *state* akan mengalami *transformasi SubBytes, ShiftRows, MixColumns,* dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam *algoritma AES* disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada round terakhir, state tidak mengalami transformasi *MixColumns* (Amin, 2016).

Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang lain. Dengan ekkripsi, data kita disandikan (Encrypted) dengan menggunakan sebuah kunci (key). Untuk membuka (mendecrypt) data tersebut, digunakan kunci yang sama ketika mengenkrip. Enkripsi adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus. Dikarenakan enkripsi telah digunakan untuk mengamankan komunikasi di berbagai negara, hanya organisasi-organisasi tertentu dan individu yang memiliki kepentingan yang sangat mendesak akan kerahasiaan yang menggunakan enkripsi. Di pertengahan tahun 1970-an, enkripsi kuat dimanfaatkan untuk pengamanan oleh sekretariat agen pemerintah Amerika Serikat pada domain publik, dan saat ini enkripsi telah

digunakan pada sistem secara luas, seperti Internet e-commerce, jaringan Telepon bergerak dan ATM pada bank.

Keamanan dari enkripsi tergantung beberapa faktor salah satunya yaitu menjaga kerahasiaan kuncinya bukan algoritmanya. Proses enkripsi dapat diterangkan sebagai berikut:

1. Masukkan file dan key
2. Baca isi file
3. Lakukan perhitungan untuk melakukan enkripsi
4. Outputnya adalah ciphertext
5. Pilih Folder Penyimpanan
6. Selesai

Langkah-langkah pada proses enkripsi adalah sebagai berikut:

1. *Plaintext* diubah ke dalam bentuk bilangan. Untuk mengubah plaintext yang berupa huruf menjadi bilangan dapat digunakan kode *ASCII* dalam sistem bilangan desimal.
2. *Plaintext* m dinyatakan menjadi blok-blok m1,m2,m3,...., sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang [0, n-1] , sehingga transformasinya menjadi satu ke satu.
3. Setiap blok mi dienkripsi menjadi blok ci dengan rumus mi = ci e mod n

2.6 Dekripsi

Dekripsi digunakan untuk mengembalikan data-data atau informasi yang sudah dienkripsi kebentuk awal sehingga dapat dibaca kembali dengan baik.satu kaidah upaya pengolahan data menjadi sesuatu yang dapat diutarakan secara jelas dan tepat dengan tujuan agar dapat dimengerti oleh orang yang tidak langsung mengalaminya sendir dalam keilmuan, deskripsi diperlukan agar peneliti tidak melupakan pengalamannya dan agar pengalaman tersebut dapat dibandingkan dengan pengalaman peneliti lain, sehingga mudah untuk dilakukan pemeriksaan dan kontrol terhadap deskripsi tersebut. Pada umumnya deskripsi menegaskan sesuatu, seperti apa sesuatu itu kelihatannya, bagaimana bunyinya, bagaimana rasanya, dan sebagainya (Amin, 2016).

Deskripsi yang detail diciptakan dan dipakai dalam disiplin ilmu sebagai istilah teknik. Saat data yang dikumpulkan, deskripsi, analisis dan kesimpulannya lebih disajikan dalam angka-angka maka hal ini dinamakan penelitian kuantitatif. Sebaliknya, apabila data, deskripsi, dan analisis kesimpulannya disajikan dalam uraian kata-kata maka dinamakan penelitian kualitatif. Proses deskripsi dapat diterangkan sebagai berikut:

1. Pilih folder penyimpanan
2. Masukkan file cipher & key
3. Baca isi file
4. Lakukan perhitungan untuk dekripsi
5. Outputnya adalah plaintext

Dekripsi adalah proses memperoleh kembali *plaintext* menjadi *ciphertext*, atau proses pengubahan kembali data yang berbentuk rahasia menjadi semula. *Transformasi byte* yang digunakan pada invers cipher adalah *InvShiftRows, InvSubBytes, InvMixColumns,* dan *AddRoundKey*. Langkah-langkah pada proses *dekripsi* adalah sebagai berikut:

1. Setiap blok *ciphertext* ci *didekripsi* kembali menjadi blok mi dengan rumus mi = ci d mod n
2. Kemudian blok-blok m1,m2,m3,...., diubah kembali ke bentuk huruf dengan melihat kode *ASCII* hasil *dekripsi*. (Yuza, dkk, 2018)

2.7 Gronsfeld Cipher

Gronsfeld Cipher adalah metode menyandikan teks alfabet dengan menggunakan deretan sandi *Caesar* berdasarkan huruf-huruf pada kata kunci. Sandi *Vigenère* merupakan bentuk sederhana dari sandi substitusi *polialfabetik*. Kelebihan sandi ini dibanding sandi *Caesar* dan sandi *monoalfabetik* lainnya adalah sandi ini tidak begitu rentan terhadap metode pemecahan sandi yang disebut analisis frekuensi. Giovan Batista Belaso menjelaskan metode ini dalam buku La cifra del. Sig. Giovan Batista Belaso (1553); dan disempurnakan oleh diplomat Perancis Blaise de Vigenère, pada 1586. Pada abad ke-19, banyak orang yang mengira *Vigenère* adalah penemu sandi ini, sehingga, sandi ini dikenal luas sebagai "sandi *Vigenère*". Sandi ini dikenal luas karena cara kerjanya mudah dimengerti dan dijalankan, dan bagi para pemula sulit dipecahkan. Pada saat kejayaannya, sandi ini dijuluki le chiffre indéchiffrable (bahasa Prancis: 'sandi yang tak terpecahkan').

Metode pemecahan sandi ini baru ditemukan pada abad ke-19. Pada tahun 1854, Charles Babbage menemukan cara untuk memecahkan sandi Vigenère. Metode ini dinamakan tes Kasiski karena Friedrich Kasiski-lah yang pertama mempublikasikannya. Tabel Vigenère, atau tabula recta, dapat digunakan untuk enkripsi maupun dekripsi sandi Vigenère. Sandi Vigenère sebenarnya merupakan pengembangan dari sandi Caesar. Pada sandi Caesar, setiap huruf teks terang digantikan dengan huruf lain yang memiliki perbedaan tertentu pada urutan alfabet. Misalnya pada sandi Caesar dengan geseran 3, A menjadi D, B menjadi Eand dan seterusnya. Sandi Vigenère terdiri dari beberapa sandi Caesar dengan nilai geseran yang berbeda. Untuk menyandikan suatu pesan, digunakan sebuah tabel alfabet yang disebut tabel Vigenère. Tabel Vigenère berisi alfabet yang dituliskan dalam 26 baris, masing-masing baris digeser satu urutan ke kiri dari baris sebelumnya, membentuk ke-26 kemungkinan sandi Caesar. Setiap huruf disandikan dengan menggunakan baris yang berbeda-beda, sesuai kata kunci yang diulang Misalnya, teks terang yang hendak disandikan adalah perintah "Serbu Berlin" Sedangkan kata kunci antara pengirim dan tujuan adalah "PIZZA" diulang sehingga jumlah hurufnya sama banyak dengan plaintext nya yaitu PIZZAPIZZAP.

Huruf pertama pada teks terang, S, disandikan dengan menggunakan baris berjudul P, huruf pertama pada kata kunci. Pada baris P dan kolom S di tabel Vigenère, terdapat huruf H. Demikian pula untuk huruf kedua, digunakan huruf yang terletak pada baris I (huruf kedua kata kunci) dan kolom E (huruf kedua teks terang), yaitu huruf M. Proses ini dijalankan terus sehingga:

Plaintext : serbuberlin

Kata kunci : PIZZAPIZZAP

Ciphertext : HMQAUQMQKIC

Proses sebaliknya (disebut dekripsi), dilakukan dengan mencari huruf teks bersandi pada baris berjudul huruf dari kata kunci. Misalnya, pada contoh di atas, untuk huruf pertama, kita mencari huruf H (huruf pertama teks tersandi) pada baris P(huruf pertama pada kata kunci), yang terdapat pada kolom S, sehingga huruf pertama adalah S. Lalu M terdapat pada baris I di kolom E, sehingga diketahui huruf kedua teks terang adalah E, dan seterusnya hingga didapat perintah "*serbuberlin*".

Salah satu cipher substitusi sederhana *polyalphabetic* adalah *Gronsfeld*. *Gaspar Schot* adalah seorang *kriptografer* abad ke 17 di Jerman, yang belajar *cipher* ini selama perjalanan antara *Mainz* dan *Frankfurt* dengan menghitung *Gronsfeld*, maka terciptalah nama dari cipher tersebut yaitu *Gronsfeld* (*Optimal Cryptography Technique*, Abhishek P.S. Rathore dan K. Avinash Muthuswamy).

Algoritma Gronsfeld menggunakan suatu kunci numerik yang biasanya cukup pendek misalnya 7341, kunci ini diulang secara priodik, sesuai dengan jumlah kata plainteks. Idenya adalah dengan mengganti huruf dengan bilangan desimal maka akan melainkan hanya berupa susunan angka. Kemudian enkripsi menggunakan prinsip yang sama dengan *Algoritma Vigenère* yaitu menggunakan tabel yang hanya berukuran 10x10. (Azanuddin, 2015).

Tabel 2.1 Gronsfeld Tabel

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2
4	4	5	6	7	8	9	0	1	2	3
5	5	6	7	8	9	0	1	2	3	4
6	6	7	8	9	0	1	2	3	4	5
7	7	8	9	0	1	2	3	4	5	6
8	8	9	0	1	2	3	4	5	6	7
9	9	0	1	2	3	4	5	6	7	8

2.7.1 Proses Enkripsi

Untuk mengenkripsi, hanya menambahkan jumlah huruf yang akan dienkripsi sesuai dengan jumlah kunci tetapi terlebih dahulu pesan tersebut diubah ke kode nilai desimal, plainteks yang dihasilkan akan menjadi *chiperteks*. Langkah-langkah proses enkripsi adalah sebagai berikut :

1. Tentukan *plainteks* yang akan dienkripsi beserta kunci.
2. Jika panjang kunci tidak sama dengan panjang plainteks maka kunci yang ada diulang secara priodik sehingga jumlah karakter kuncinya sama dengan jumlah plainteks nya.
3. Selanjutnya ubah plainteks ke bentuk nilai desimal kemudian ditambahkan dengan kunci. Jika penambahan lebih besar dari jumlah *mod*, maka diambil nilai sisa hasil bagi nya.
4. Setelah dijumlahkan dengan kunci maka langkah berikutnya adalah mengubah kembali ke bentuk karakter.

Algoritma enkripsi Gronsfeld cipher : Ci = (Pi + Ki) mod 256

Contoh Proses Enkripsi :

Plaintext : GRO

Kunci : 734

G = 71

R = 82

O = 79

Key : 7,3,4

C1 = (G + k1) mod 256

= (71 + 7) mod 256

= 78 mod 256

= 78 = N

C2 = (R + k2) mod 256

= (82 + 3) mod 256

= 85 mod 256

= 85 = U

C3 = (O + k3) mod 256

= (79 + 4) mod 256

= 83 mod 256

= 83 = S

Chipertext : NUS

2.7.2 Proses Dekripsi

Dekripsi adalah proses sebaliknya, dimana *chiperteks* nya diubah menjadi nilai *decimal* dan dikurangi dengan jumlah kunci kemudian dikembalikan ke karakter. Langkah-langkah proses dekripsi adalah sebagai berikut :

1. Terlebih dahulu mengubah *chiperteks* ke nilai desimal.
2. Kemudian nilai desimal *chiperteks* nya dikurangi sesuai dengan kunci
3. Setelah dikurangi dengan kunci maka langkah berikutnya adalah mengubah kembali kebentuk karakter

Contoh Proses Dekripsi :

C1 = (N - k1) mod 256

= (78 - 7) mod 256

= 71 mod 256

= 71 = G

C2 = (U - k2) mod 256

= (85 - 3) mod 256

= 83 mod 256

= 83 = R

C3 = (S - k3) mod 256

= (83 - 4) mod 256

= 79 mod 256

= 79 = O

Plaintext : GRO

2.8 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak (Mallu, 2015). UML menawarkan sebuah standar untuk merancang model sebuah system. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*) (Isa & Hartawan, 2017).

Unifed Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendekskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (Wasserkrug et al., 2009).

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).

2.8.1 Use Case Diagram

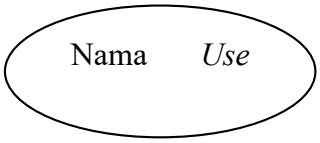
Use Case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). sehingga pembuatan use case diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan

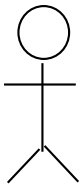
alur atau urutan kejadian. Sebuah use case diagram mempresentasikan sebuah interaksi antara aktor dengan sistem (Isa & Hartawan, 2017).

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenarion. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu.

Sedangkan menurut Ade Hendini, *Use Use case diagram* merupakan pemodelan untuk kelakuakn (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Hendini., 2016). Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel 2.2 Simbol Use Case Diagram

No	Simbol	Deskripsi
1	*Use case* 	Gambaran unit yang saling berkaitan antara aktor dengan sistem yang berjalan

2	Aktor  Nama aktor	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan dibuat.
3	Asosiasi / *Association*	Komunikasi antara aktor dan *use case*.
4	Ekstensi / *Extend* <<extend>>	Kelakuan yang hanya berjalan di bawah kondisi tertentu. Seperti jika akun sesuai, atau jika *session* sesuai.
5	Generalisasi	Elemen yang menjadi spesialisasi elemen lain.
6	*Include* <<include>>	Kelakuan yang harus terpenuhi agar suatu *event* dapat terjadi.

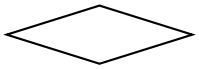
Sumber: (Hendini., 2016)

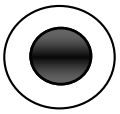
2.8.2 Activity Diagram

Menurut Indra Griha Tofik Isa dan George Pri Hartawan, Activity Digram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan flowchart karena memodelkan *workflow* dari suatu aktifitas ke aktifitas yang lainnya, atau dari aktifitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case* (Isa & Hartawan, 2017).

Activity Diagram adalah bagian penting dari *UML*, yang menggambarkan aspek dinamis dari sistem. logika prosedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram*. *Activity diagram* mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung prilaku paralel sedangkan *flowchart* tidak bisa (Kurniawan, 2018). *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu:

Tabel 2.3 Simbol Activity Diagram

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / *decision* 	Asosiasi percabangan dimana jika ada aktivitas pilihan lebih dari satu.
4	Penggabungan / Join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

5	Status Akhir 	Tahap akhir dari proses sistem.

Sumber: (Hendini., 2016)

2.9 Visual Basic.Net 2010

Bahasa Pemrograman *Microsoft Visual Basic .NET* adalah sebuah bahasa pemrograman tingkat tinggi untuk *Microsoft .NET Framework*. Walaupun *VB.NET* ini memang dibuat supaya mudah dipahami dan dipelajari, namun bahasa pemrograman ini juga cukup *powerful* untuk memenuhi kebutuhan dari *programmer* yang berpengalaman. Bahasa pemrograman *Visual Basic .NET* mirip dengan bahasa pemrograman *Visual Basic*, namun keduanya tidak sama".

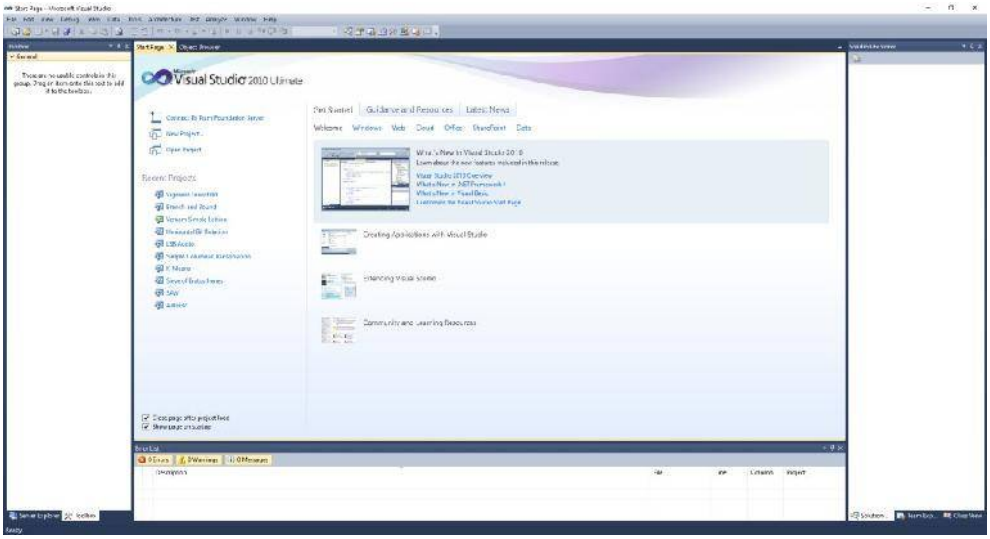
Bahasa pemrograman *Visual Basic .NET* memiliki struktur penulisan yang mirip dengan bahasa Inggris, di mana hal ini juga menyebabkan kemudahan dalam membaca dan mengerti dari sebuah kode. Di mana dimungkinkan, kata ataupun frasa yang memiliki arti digunakan dan bukannya menggunakan singkatan, akronim ataupun *special characters*".

Pada intinya *Visual Basic.NET* ini adalah sebuah bahasa pemrograman yang berorientasi pada *object*, yang bisa dianggap sebagai evolusi selanjutnya dari bahasa pemrograman *Visual Basic* standar (Wibowo, 2014).

2.9.1 Lingkungan kerja Visual Basic.Net 2010

Pada saat pertama kali dijalankan Visual Basic 2010 Ultimate, akan menampilkan sebuah jendela Splash Visual Studio 2010 Ultimate, setelah jendela

Splash Visual Studio 2010 Ultimate muncul kemudian akan keluar sebuah start page Microsoft Visual Studio seperti gambar 2.3.



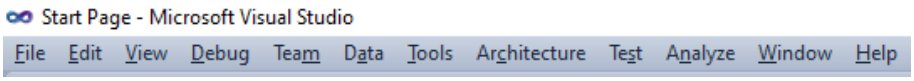
Gambar 2.3 Tampilan Microsoft Visual Studio 2010

2.9.2 Komponen Visual Basic.Net 2010

Pada saat membuka program Visual Basic.Net, ada beberapa komponen yang terlihat. Berikut ini adalah beberapa komponen dari Visual Basic.Net:

1. Menu Bar

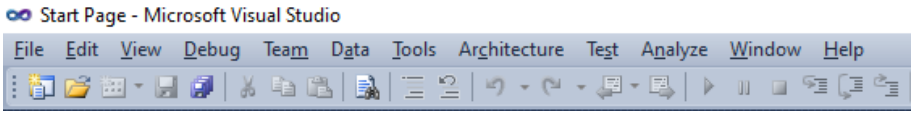
   Menu Bar adalah bagian dari *IDE* yang terdiri atas perintah-perintah untuk mengatur *IDE*, mengedit kode, dan mengeksekusi program. Menu yang terdapat pada menu bar adalah *menu file, edit, view, project, build, debug, data, tools*, *window* dan *help*. *Menu bar* pada *Visual Sudio* 2010 seperti terlihat pada gambar 2.5.



Gambar 2.4 Tampilan Menu Bar

2. Toolbar

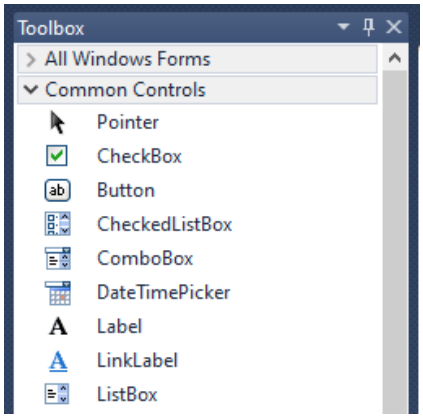
   Fasilitas ini dapat mempercepat pengaksesan perintah-perintah yang ada dalam pemrograman seperti terlihat pada gambar 2.6.



Gambar 2.5 Tampilan Toolbar

3. Toolbox

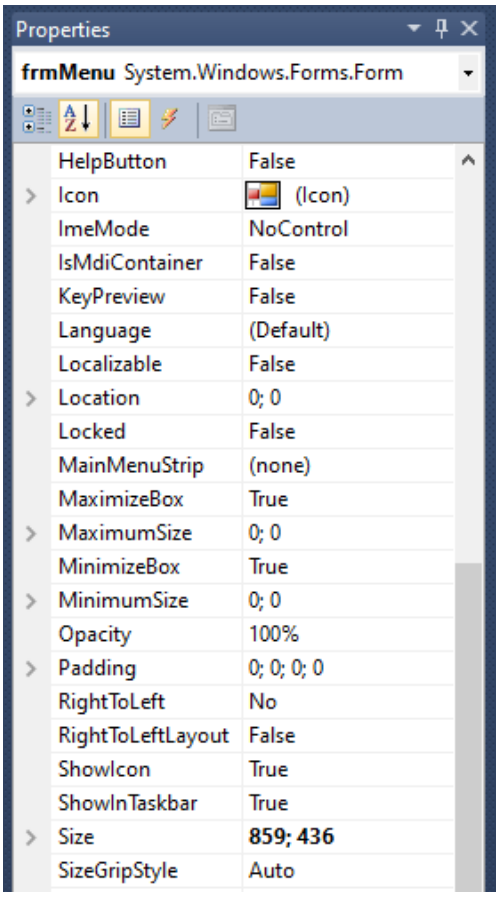
   Sebuah *window* yang berisi tombol-tombol kontrol yang akan Anda gunakan untuk mendesain atau membangun sebuah *form* atau *report* seperti terlihat pada gambar 2.7.



Gambar 2.6 Tampilan Toolbox

4. Properties Window

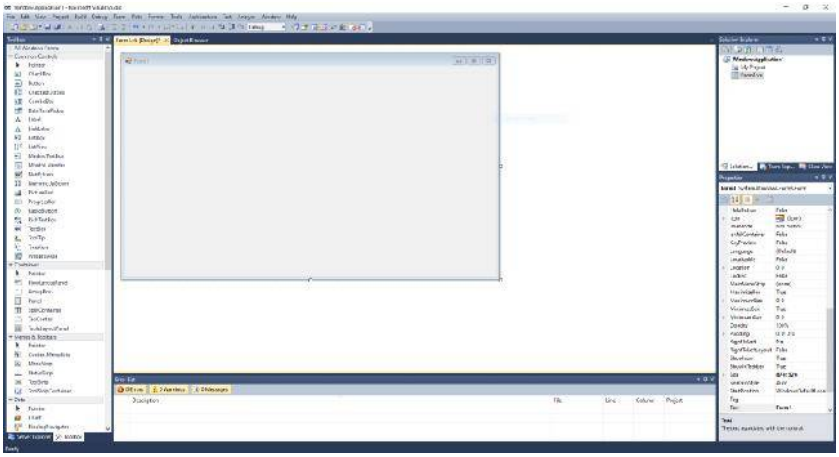
   Properties window adalah tempat menyimpan *property* dari setiap objek control dan komponen.



Gambar 2.7 Tampilan Properties

5. Form

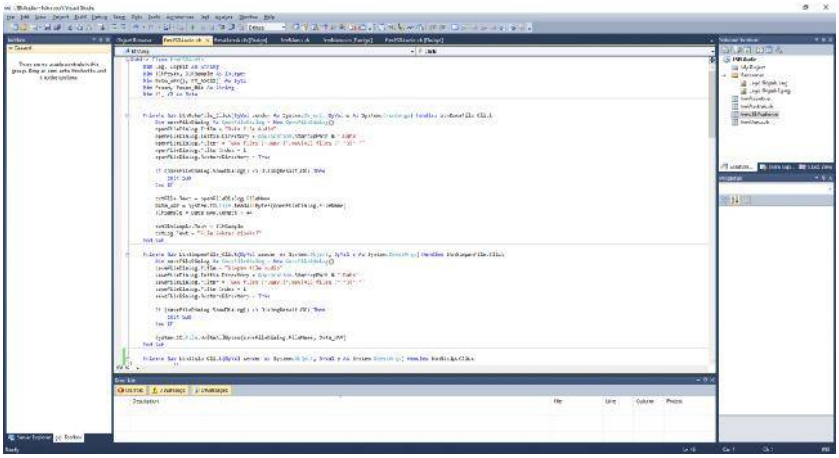
   Form merupakan tempat di mana kontrol-kontrol diletakkan. Form juga berfungsi sebagai tempat pembuatan tampilan atau antarmuka (*user interface)* dari sebuah aplikasi *windows*.



Gambar 2.8 Tampilan Form

6. Code Editor

   Code Editor adalah tempat di mana kita meletakkan atau menuliskan kode program dari program aplikasi kita.



Gambar 2.9 Tampilan Code Editor

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Perangkat Keras dan Lunak

Sistem yang telah dirancang membutuhkan perangkat kerass dan perangkat lunak dalam mendukung kinerja program aplikasi tersebut. Berikut ini adalah kebutuhan perangkat tersebut:

1. *Hardware* (Perangkat Keras)

   Untuk menjalankan sistem ini, penulis menggunakan laptop dengan spesifikasi RAM 2GB, Processor Intel Core i3, Hard drive 500GB dan Display 14".

2. Software (Perangkat Lunak)

   Sedangkan pada sisi software, penulis menggunakan beberapa perangkat lunak yaitu:

   a. Windows 7
   b. Microsoft Visual Studio 2010
   c. Microsoft Word 2019
   d. Microsoft Excel 2019
   e. Microsoft Visio 2019
   f. Snipping Tool

3.1 Implementasi Sistem

Implementasi sistem akan menjelaskan bagaimana program aplikasi dapat digunakan oleh pengguna. Pada penggunaan sistem, pengguna dapat mulai memilih menu utam yang kemudian memilih sub menu yang ada didalamnya. Berikut merupakan hasil tampilan dari program aplikasi yang telah dibuat oleh penulis tentang aplikasi enkripsi dan dekripsi menggunakan algoritma Gronsfeld Cipher.

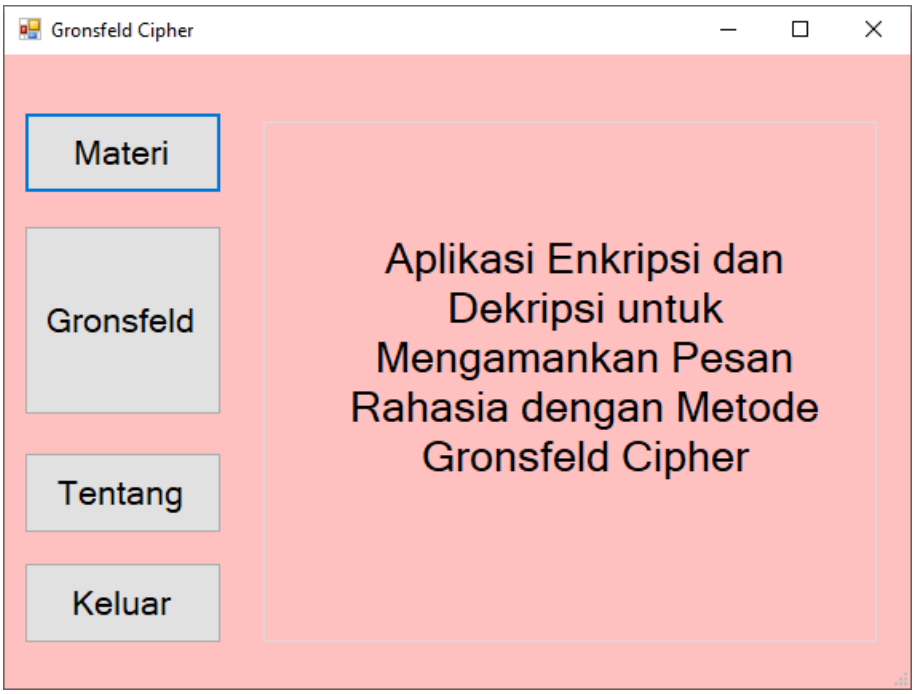
3.1.1 Tampilan Halaman Judul

Halaman judul adalah halaman yang pertama sekali muncul ketika program aplikasi dijalankan. Gambar 4.1 adalah hasil tampilan halaman judul.

Gambar 4.1 Halaman Judul

3.1.2 Tampilan Halaman Menu Utama

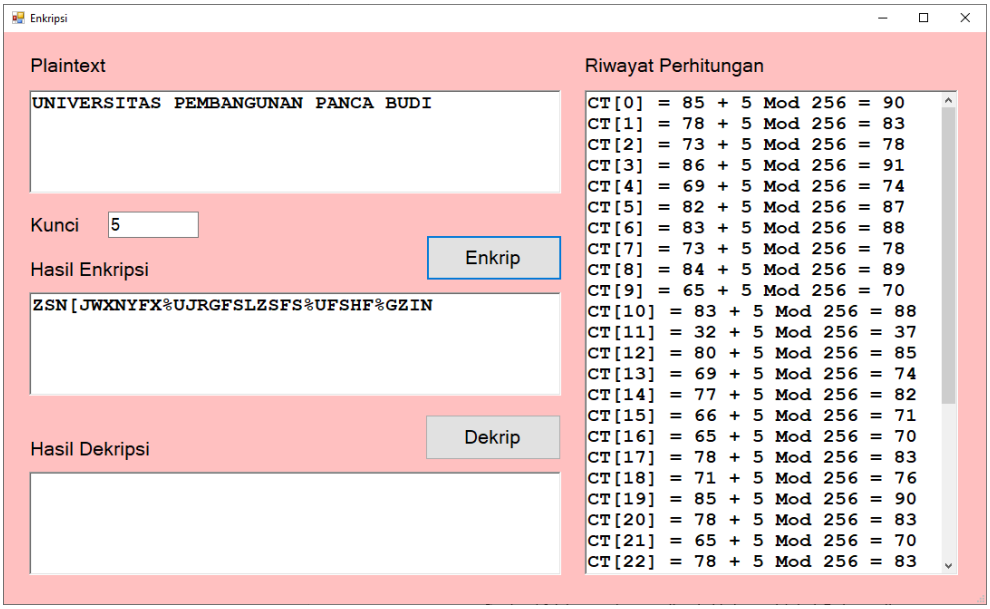
Halaman Menu Utama merupakan halaman utama sebuah program aplikasi dimana pengguna dapat melakukan perpindahan ke menu-menu yang ada di dalamnya. Gambar 4.2 adalah hasil tampilan menu utama.



Gambar 4.2 Halaman Menu Utama

3.1.3 Tampilan Halaman Enkripsi

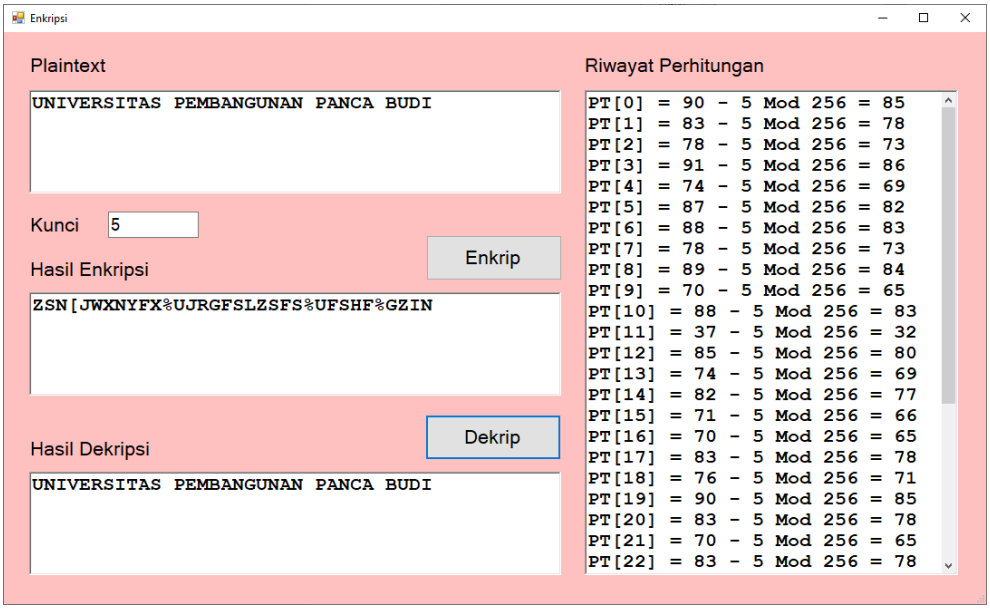
Gambar 4.3 merupakan tampilan dari halaman enkripsi pesan dengan algoritma Gronsfeld Cipher. Pada tampilan ini pengguna dapat mengetikkan pesan pada textbox yang sudah tersedia. Pengguna menentukan pergeseran dengan mengetikkan kunci pada textbox kunci. Proses enkripsi dilakukan dengan mengekan tombol enkrip. Hasil ciphertext dapat dilihat pada textbox ciphertext.



Gambar 4.3 Halaman Enkripsi

3.1.4 Tampilan Halaman Dekripsi

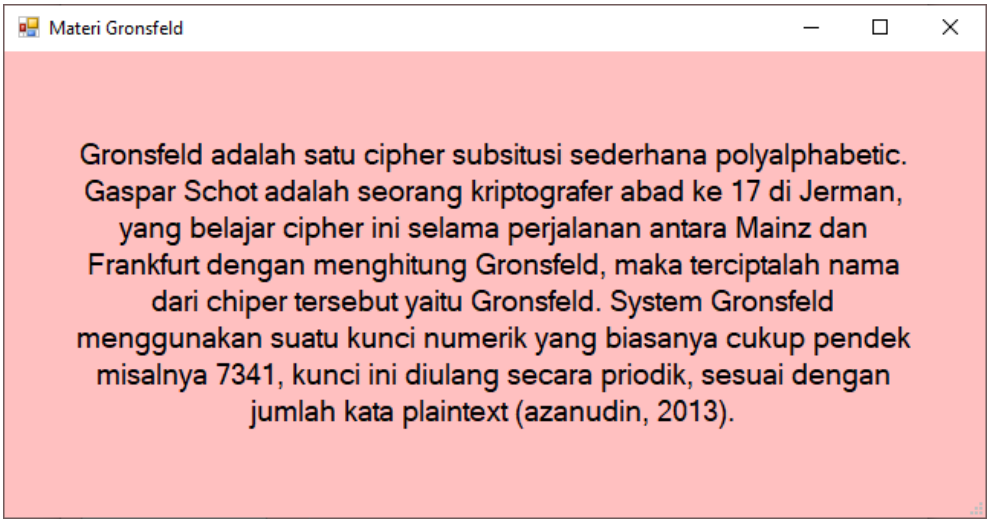
Gambar 4.4 ini merupakan tampilan dari halaman dekripsi. Pada tampilan ini pengguna menginputkan kembali ciphertext pada hasil enkripsi atau dapat menggunakan dengan karakter yang telah dihasilkan pada proses enkripsi sebelumnya. Kunci yang digunakan harus sesuai dengan proses pada saat enkripsi. Hasil plaintext dapat dilihat ketika tombol dekrip ditekan. Hasil dekripsi akan tampil pada textbox dekripsi. Riwayat perhitungan juga dapat dilihat pada textbox log untuk mengetahui proses dekripsi tersebut.



Gambar 4.4 Halaman Dekripsi

3.1.5 Halaman Materi

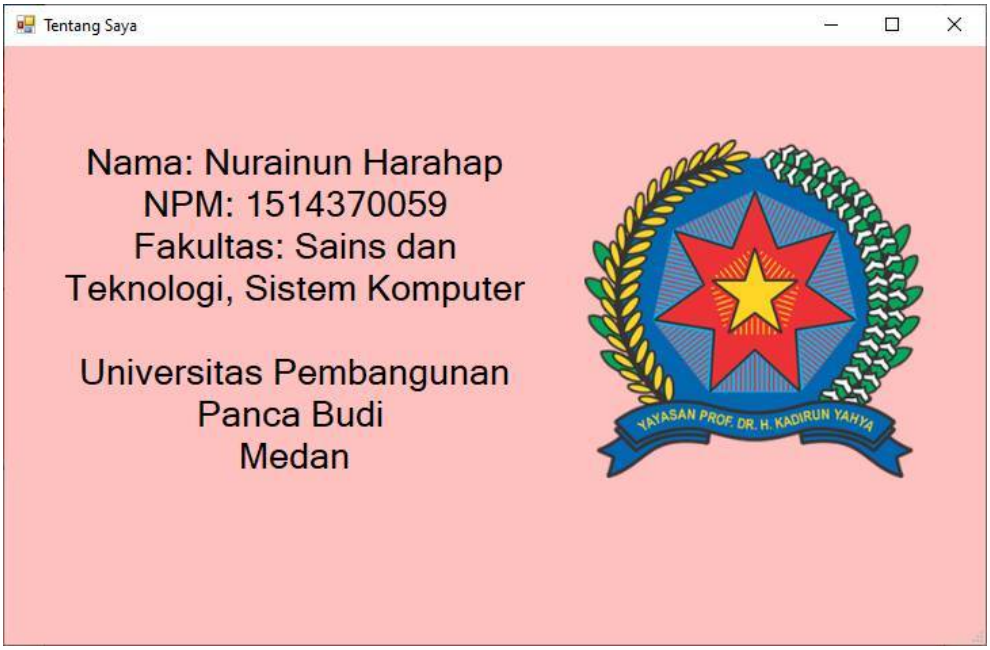
Gambar 4.5 adalah tampilan dari halaman materi yang menjelaskan secara singkat tentang algoritma Gronsfeld Cipher.



Gambar 4.5 Halaman Materi

3.1.6 Halaman Tentang

Gambar 4.6 merupakan tampilan dari halaman tentang aplikasi. Pada tampilan ini nantinya pengguna dapat melihat secara singkat biodata penulis.



Gambar 4.6 Halaman Tentang

3.2 Pembahasan

Pengujian sistem dilakukan bertujuan untuk melihat kebenaran program aplikasi yang telah dibuat sehingga menghindari kesalahan di masa akan datang. Pengujian ini bertujuan untuk menghitung secara matematika proses enkripsi dan dekripsi pada algoritma Gronsfeld Cipher tersebut. Hasil perhitungan enkripsi dengan metode Gronsfeld Cipher dapat dilakukan perhitungan manual sebagai berikut:

Pesan (Plaintext) : FAKULTAS SAINS DAN TEKNOLOGI

Kunci : 5, 3, 1

Hasil Enkripsi : KDLZOUFV!XDJSV!IDO%WFPQPQRHN

Perhitungan :

CT[0] = 70 + 5 Mod 256 = 75

CT[1] = 65 + 3 Mod 256 = 68

CT[2] = 75 + 1 Mod 256 = 76

CT[3] = 85 + 5 Mod 256 = 90

CT[4] = 76 + 3 Mod 256 = 79

CT[5] = 84 + 1 Mod 256 = 85

CT[6] = 65 + 5 Mod 256 = 70

CT[7] = 83 + 3 Mod 256 = 86

CT[8] = 32 + 1 Mod 256 = 33

CT[9] = 83 + 5 Mod 256 = 88

CT[10] = 65 + 3 Mod 256 = 68

CT[11] = 73 + 1 Mod 256 = 74

CT[12] = 78 + 5 Mod 256 = 83

CT[13] = 83 + 3 Mod 256 = 86

CT[14] = 32 + 1 Mod 256 = 33

CT[15] = 68 + 5 Mod 256 = 73

CT[16] = 65 + 3 Mod 256 = 68

CT[17] = 78 + 1 Mod 256 = 79

CT[18] = 32 + 5 Mod 256 = 37

CT[19] = 84 + 3 Mod 256 = 87

CT[20] = 69 + 1 Mod 256 = 70

CT[21] = 75 + 5 Mod 256 = 80

CT[22] = 78 + 3 Mod 256 = 81

CT[23] = 79 + 1 Mod 256 = 80

CT[24] = 76 + 5 Mod 256 = 81

CT[25] = 79 + 3 Mod 256 = 82

CT[26] = 71 + 1 Mod 256 = 7

CT[27] = 73 + 5 Mod 256 = 78

Hasil perhitungan enkripsi dengan metode Gronsfeld Cipher dapat dilakukan perhitungan manual sebagai berikut:

Pesan (Ciphertext) : KDLZOUFV!XDJSV!IDO%WFPQPQRHN

Kunci : 5, 3, 1

Hasil Dekripsi : FAKULTAS SAINS DAN TEKNOLOGI

PT[0] = 75 - 5 Mod 256 = 70

PT[1] = 68 - 3 Mod 256 = 65

PT[2] = 76 - 1 Mod 256 = 75

PT[3] = 90 - 5 Mod 256 = 85

PT[4] = 79 - 3 Mod 256 = 76

PT[5] = 85 - 1 Mod 256 = 84

PT[6] = 70 - 5 Mod 256 = 65

PT[7] = 86 - 3 Mod 256 = 83

PT[8] = 33 - 1 Mod 256 = 32

PT[9] = 88 - 5 Mod 256 = 83

PT[10] = 68 - 3 Mod 256 = 65

PT[11] = 74 - 1 Mod 256 = 73

PT[12] = 83 - 5 Mod 256 = 78

PT[13] = 86 - 3 Mod 256 = 83

PT[14] = 33 - 1 Mod 256 = 32

PT[15] = 73 - 5 Mod 256 = 68

PT[16] = 68 - 3 Mod 256 = 65

PT[17] = 79 - 1 Mod 256 = 78

PT[18] = 37 - 5 Mod 256 = 32

PT[19] = 87 - 3 Mod 256 = 84

PT[20] = 70 - 1 Mod 256 = 69

PT[21] = 80 - 5 Mod 256 = 75

PT[22] = 81 - 3 Mod 256 = 78

PT[23] = 80 - 1 Mod 256 = 79

PT[24] = 81 - 5 Mod 256 = 76

PT[25] = 82 - 3 Mod 256 = 79

PT[26] = 72 - 1 Mod 256 = 71

PT[27] = 78 - 5 Mod 256 = 73

BAB V

PENUTUP

5.1 Kesimpulan

Berikut merupakan kesimpulan yang penulis buat berdasarkan pembahasan pada implementasi dan penggunaan algoritma Gronsfeld Cipher:

1. Algoritma Gronsfeld Cipher bekerja dengan cara melakukan pergeseran karakter atau berjenis substitusi.
2. Algoritma Gronsfeld Cipher adalah jenis lain dari algoritma Vigenere Cipher.
3. Sistem enkripsi ini menggunakan metode enkripsi dan dekripsi dengan algoritma Gronsfeld dapat bekerja dengan baik.
4. Penggunaan algoritma Gronsfeld dalam proses enkripsi dan dekripsi ini dinilai efektif karena cara kerja yang mudah dan cepat karena hanya melakukan pergeseran karakter saja.
5. Pembuatan sistem enkripsi dan dekripsi pada pesan dapat dilakukan oleh pengguna sehingga kerahasiaan data yang ada di dalamnya menjadi lebih aman dan terpercaya.

5.2 Saran

Berikut merupakan saran yang penulis paparkan berdasarkan pembahasan dalam implementasi dan penggunaan algoritma Gronsfeld CIpher:

1. Sistem ini masih berbasis desktop yang artinya sistem hanya dapat diakses pada perangkat lokal saja, hendaknya sistem dapat dilakukan secara online.
2. Dalam proses enkripsi dan dekripsi, sistem hanya dapat mengenkripsi dan mendekripsi dengan sebuah angka saja, hendaknya dapat dilakukan kombinasi angka sehingga meningkatkan keamanan kunci.
3. Kombinasi algoritma sangat diharapkan agar dapat meningkatkan keamanan pesan.

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.

Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). Jurnal Media Informatika Budidarma, 2(2).

Batubara, S., Wahyuni, S., & Hariyanto, E. (2018, September). Penerapan Metode Certainty Factor Pada Sistem Pakar Diagnosa Penyakit Dalam. In Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 81-86).

Amin, M. M. (2016). Implementasi Kriptografi Klasik Pada Komunikasi Berbasis Teks. *Jurnal Pseudocode*, *3*(2).

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

Barone, L., Williams, J., & Micklos, D. (2017). Unmet needs for analyzing biological big data: A survey of 704 NSF principal investigators. *PLOS Computational Biology*, *13*(10), e1005755. https://doi.org/10.1371/journal.pcbi.1005755

Dhany, H. W., Izhari, F., Fahmi, H., Tulus, M., & Sutarman, M. (2017, October). Encryption and decryption using password based encryption, MD5, and DES. In International Conference on Public Policy, Social Computing and Development 2017 (ICOPOSDev 2017) (pp. 278-283). Atlantis Press.

Batubara, S., Hariyanto, E., Wahyuni, S., Sulistianingsih, I., & Mayasari, N. (2019, August). Application of Mamdani and Sugeno Fuzzy Toward Ready-Mix Concrete Quality Control. In Journal of Physics: Conference Series (Vol. 1255, No. 1, p. 012061). IOP Publishing.

Damanik, W. A. (2019). Analisis Penentuan Pemberian Beasiswa Berprestasi Menggunakan Metode Decision Tree dan SVM (Support Vector Machine)(Studi Kasus: Universitas Pembangunan Pancabudi Medan). Jurnal Teknik dan Informatika, 6(1), 65-67.

Hardinata, R. S. (2019). Audit Tata Kelola Teknologi Informasi menggunakan Cobit 5 (Studi Kasus: Universitas Pembangunan Panca Budi Medan). Jurnal Teknik dan Informatika, 6(1), 42-45.

Hendrawan, J., & Perwitasari, I. D. (2019). Aplikasi Pengenalan Pahlawan Nasional dan Pahlawan Revolusi Berbasis Android. JurTI (Jurnal Teknologi Informasi), 3(1), 34-40.

Herdianto, H., & Anggraini, S. (2019, May). Perancangan sistem pendeteksi uang palsu untuk tuna netra menggunakan arduino uno. In Seminar Nasional Teknik (SEMNASTEK) UISU (Vol. 2, No. 1, pp. 136-140).

Gurevich, Y. (2012). *What Is an Algorithm?* https://doi.org/10.1007/978-3-642-27660-6_3

Hendini., A. (2016). Pemodelan UML Sistem Informasi Monitoring Penjualan Dan Stok Barang. *Jurnal Khatulistiwa Informatika*, *4*(2), 107–116. https://doi.org/10.31294/jki.v4i2.1262.g1027

Isa, I. G. T., & Hartawan, G. P. (2017). Perancangan Aplikasi Koperasi Simpan Pinjam Berbasis Web (Studi Kasus Koperasi Mitra Setia). *Jurnal Ilmiah Ilmu Ekonomi (Jurnal Akuntansi, Pajak Dan Manajemen)*, *5*(10), 139–151.

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

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

Mallu, S. (2015). Sistem Pendukung Keputusan Penentuan Karyawan Kontrak Menjadi Karyawan Teatap Menggunakan Metode TOPSIS. *Jurnal Imliah Teknologi Informasi Terapan*, *1*(2), 36–42.

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

Putri, G. G., Setyorini, W., & Rahayani, R. D. (2018). Analisis Kriptografi Simetris AES dan Kriptografi Asimetris RSA pada Enkripsi Citra Digital. *ETHOS (Jurnal Penelitian Dan Pengabdian)*, *6*(2), 197–207.

Nasution, M. Z. (2019). Penerapan principal component analysis (pca) dalam penentuan faktor dominan yang mempengaruhi pengidap kanker serviks (Studi Kasus: Cervical Cancer Dataset). Jurnal Mantik, 3(1), 204-210.

Novelan, M. S. (2019). Perancangan Alat Simulasi Sistem Kendali Lampu Rumah Menggunakan Aplikasi Android. ALGORITMA: JURNAL ILMU KOMPUTER DAN INFORMATIKA, 3(2), 1.

Sulistianingsih, I. (2019). Sistem Pendukung Keputusan Penentuan Menu Makanan Sehat untuk Pasien Rawat Inap. Jurnal Teknik dan Informatika, 6(1), 6-11.

Tasril, V., & Putri, R. E. (2019). Perancangan Media Pembelajaran Interaktif Biologi Materi Sistem Pencernaan Makanan Manusia Berbasis Macromedia Flash. Jurnal Ilmiah Core IT: Community Research Information Technology, 7(1).

Tasril, V., Khairul, K., & Wibowo, F. (2019). Aplikasi Sistem Informasi untuk Menentukan Kualitas Beras Berbasis Android pada Kelompok Tani Jaya Makmur Desa Benyumas. Informatika, 7(3), 133-142.

https://doi.org/10.29313/ethos.v6i2.2909

Rao, R. V., & Selvamani, K. (2015). Data Security Challenges and Its Solutions in Cloud Computing. *Procedia Computer Science*, *48*, 204–209. https://doi.org/10.1016/j.procs.2015.04.171

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

Sopyan, Y., Supriyadi, S., & Kurniadi, E. (2016). Implementasi Sistem Pendukung Keputusan Penerimaan Siswa baru Menggunakan Metode Simple Additive Weighting (Studi Kasus : SMK Negeri 3 Kuningan). *Jurnal Nuansa Informatika*, *11*(1).

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

Wasserkrug, S., Dalvi, N., Munson, E. V., Gogolla, M., Sirangelo, C., Fischer-Hübner, S., … Snodgrass, R. T. (2009). Unified Modeling Language. In

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

Wibowo, H. R. (2014). *Visual Basic Database*. Yogyakarta: Jubilee Enterprise.

Zhang, D., Tsotras, V. J., Levialdi, S., Grinstein, G., Berry, D. A., Gouet-Brunet, V., … Pitoura, E. (2009). Indexed Sequential Access Method. In

Encyclopedia of Database Systems (pp. 1435–1438). https://doi.org/10.1007/978-0-387-39940-9_738