IMPLEMENTASI DAN PENGGUNAAN ALGORITMA BASE64 DALAM PENGAMANAN FILE VIDEO

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	:	KHAIRANI ANGELA PUTRI SEMBIRING
NPM	:	1414370213
PROGRAM STUDI	:	SISTEM KOMPUTER

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2020

ABSTRAK

KHAIRANI ANGELA PUTRI SEMBIRING
Implementasi dan Penggunaan Algoritma Base64 dalam Pengamanan File Video
2020

Data-data pribadi sering kali menjadi sasaran orang yang tidak bertanggung jawab untuk disalahgunakan. Pencurian yang dilakukan adalah untuk mendapatkan keuntungan dari orang yang memiliki data tersebut. Selain pencurian file-file kerja, pencurian juga dilakukan terhadap file video. Pencurian file ini bertujuan untuk mengetahui apa isi video tersebut. Seseorang memiliki video rekaman pribadi yang tidak boleh diketahui oleh orang lain. Penyalahgunaan file video akan berakibat fatal bagi pemilik video tersebut. Teknik kriptografi diperlukan dalam pengamanan video. Algoritma Caesar Cipher dapat membantu pengguna dalam mengamankan file video tersebut. Algoritma Base64 dapat digunakan untuk mengganti format ASCII 256 menjadi Base64 sehingga mudah untuk dikirimkan atau disimpan dalam suatu media penyimpanan. Algoritma ini akan membuat struktur file tersebut menjadi lebih sederhana agar dapat ditampilkan dan disimpan. Dengan menerapkan algoritma Base64 dan Caesar Cipher pada file video, keamanan dan kerahasiaan file tersebut akan terjamin.

Kata Kunci: algoritma, keamanan, Base64, enkripsi, dekripsi, Caesar

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

KATA PENGANTAR

Puji syukur penulis ucapkan ke hadirat Allah SWT serta shalawat atas Nabi Muhammad SAW karena berkat rahmat kesehatan dan hidayah-Nya, sehingga penulis dapat menyelesaikan penulisan skripsi tepat pada waktunya. Dalam penulisan skripsi ini, penulis memilih judul **"IMPLEMENTASI DAN PENGGUNAAN ALGORITMA *BASE64* DALAM PENGAMANAN FILE VIDEO"**.

Penulisan skripsi ini adalah Salah satu syarat untuk memperoleh gelar sarjana komputer, selama proses penulisan skripsi ini, penulis telah banyak mendapatkan bimbingan dan bantuan baik moral maupun materi dari berbagai pihak. Pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Dr. H. Muhammad Isa Indrawan, SE., M.M., selaku Rektor Universitas Pembangunan Panca Budi Medan.
2. Hamdani, ST., M.T., selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
3. Bapak Eko Hariyanto, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. Bapak Hermansyah, S.Kom., M.Kom., selaku dosen pembimbing I yang telah meluangkan waktunya untuk membimbing dan memberikan arahan kepada penulis sehingga penulisan skripsi ini dapat diselesaikan.
5. Bapak Supiyandi, S.Kom., M.Kom., selaku dosen pembimbing II yang telah meluangkan waktunya untuk membimbing dan memberikan arahan kepada penulis sehingga penulisan skripsi ini dapat diselesaikan.
6. Terima kasih kepada kedua orang tua penulis yang telah banyak memberikan dukungan kepada penulis, memberikan motivasi dan doa sehingga penulis dapat menyelesaikan skripsi ini.
7. Dan tidak lupa juga penulis mengucapkan banyak terima kasih kepada Bapak dan Ibu Dosen selaku Pengajar pada Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.

Penulis menyadari bahwa dalam penulisan skripsi ini masih banyak terdapat kesalahan dan kekurangan. Untuk itu saran dan kritik yang sehat dari semua pihak sangat penulis harapkan demi perbaikan isi skripsi ini. Akhirnya penulis berharap skripsi ini dapat berguna bagi para pembaca dan bagi penulis khususnya.

Medan, 02 Januari 2020
Penulis

Khairani Angela Putri Sembiring
1414370213

BAB I

PENDAHULUAN

1.1 Latar Belakang

Keamanan dan kerahasiaan aspek terpenting yang dibutuhkan dalam proses pertukaran informasi melalui internet, dengan berkembangnya teknologi berkembang pula kejahatan yang terjadi, sudah berbagai macam teknik keamanan telah dikembangkan untuk melindungi dan menjaga kerahasiaan data agar terhindar dari orang yang tidak berhak dan tidak bertanggung jawab. Sistem komputer bisa dikatakan sebagai suatu sistem yang aman jika telah memenuhi beberapa syarat tertentu untuk mencapai suatu tujuan keamanan.

Video adalah salah satu jenis media yang dapat menyimpan hasil rekaman bergerak seseorang. Kadang-kadang video memiliki konten terbatas atau konten yang tidak boleh dimiliki atau diketahui oleh orang lain sehingga video ini merupakan rekaman rahasia. Seperti contoh, video kenegaraan, video wawancara, atau video tindak kekerasan dan kejahatan. Video-video ini harus disimpan secara rahasia dan tidak boleh tersebar secara luas. Hal ini bertujuan agar tidak ada kesalah pahaman terhadap isi video tersebut.

Ada beberapa teknik yang dapat dilakukan untuk mengamankan video tersebut. Salah satunya adalah dengan teknik kriptografi. Algoritma Caesar cipher dapat digunakan dalam mengamankan video. Hasil enkripsi dari video tersebut akan berubah menjadi karakter-karakter simbol yang tidak dapat dibaca. Karakter tersebut akan disesuaikan dengan karakter yang berada pada tabel ASCII. Akan

tetapi, pada proses dekripsi video tersebut sering mengalami masalah karena pembacaan karakter pada ciphertext tidak sesuai dengan apa yang ada pada tabel ASCII. Hal ini dapat dilakukan dengan penyederhanaan agar karakter yang digunakan pada hasil enkripsi adalah karakter yang sudah standar digunakan.

Untuk menyederhanakan *file* video tersebut diperlukannya sebuah cara untuk mengubah format data atau informasi tersebut menjadi format yang sederhana. Teknik ini menggunakan format Base64. Saat ini sudah banyak berkembang teknik konversi format yang mendukung untuk menyederhanakan suatu data atau informasi yang. Penyederhanaan ini dilakukan dengan mengubah format ASCII 256 karakter menjadi format Base64 dengan jumlah total sebanyak 64 karakter. Penyederhanaan ini dilakukan agar file video tersebut dapat disimpan dan dikirimkan dengan baik.

Transformasi Base64 merupakan salah satu algoritma untuk *encoding* dan *decoding* suatu data ke dalam format karakter yang didasarkan pada jumlah bilangan sebanyak 64 karakter atau dapat dikatakan sebagai salah satu metode yang digunakan untuk melakukan *encoding* (penyandian) terhadap data *binary*.

Kombinasi algoritma Caesar Cipher dan Base64 merupakan salah satu penyelesaian masalah di atas. Berdasarkan latar belakang yang sudah dikemukakan, maka penulis mengambil penelitian dengan judul **"IMPLEMENTASI DAN PENGGUNAAN ALGORITMA BASE64 DALAM PENGAMANAN FILE VIDEO".**

1.2 Rumusan Masalah

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

1. Bagaimana melakukan proses enkripsi dan dekripsi dengan algoritma Caesar Cipher pada *file* video?
2. Bagaimana menentukan pergeseran kunci pada Caesar Cipher?
3. Bagaimana melakukan transformasi karakter menggunakan algoritma Base64 pada *file* video hasil enkripsi?
4. Bagaimana menentukan padding bit pada algoritma Base64?
5. Bagaimana mengembalikan *file* hasil enkripsi dan transformasi menjadi bentuk *file* video seperti semula?

1.3 Batasan Masalah

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

1. Kunci enkripsi yang digunakan adalah hanya menggunakan angka dengan rentang 1 – 100.
2. *File* video yang digunakan bertipe .MP4.
3. *File* hasil enkripsi bertipe .B64.
4. Besar *file* video yang akan dienkripsi tidak melebihi dari 20 MB.
5. 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 melakukan proses enkripsi dan dekripsi dengan algoritma Caesar Cipher pada *file* video.
2. Untuk menentukan pergeseran kunci pada Caesar Cipher.
3. Untuk melakukan transformasi karakter menggunakan algoritma Base64 pada *file* video hasil enkripsi.
4. Untuk menentukan padding bit pada algoritma Base64.
5. Untuk mengembalikan *file* hasil enkripsi dan transformasi menjadi bentuk *file* video seperti semula.

1.5 Manfaat Penelitian

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

1. Hasil penelitian ini bermanfaat untuk memberikan informasi kepada masyarakat pada umunya agar lebih menjaga *file* video yang sudah diciptakan.
2. Meningkatkan keamanan *file* video sehingga tidak mudah untuk disalahgunakan
3. Memberikan kemudahan dalam mengirimkan *file* video yang sudah menjadi format Base64.

BAB II

LANDASAN TEORI

2.1 Pengertian Implementasi

Implementasi berasal dari bahasa Inggris yaitu *to implement* yang berarti mengimplementasikan. Implementasi merupakan penyediaan sarana untuk melaksanakan sesuatu yang menimbulkan dampak atau akibat terhadap sesuatu (Sopyan, Supriyadi, & Kurniadi, 2016).

2.2 Data

Data, dalam konteks komputasi, mengacu pada bagian informasi digital yang berbeda. Data biasanya diformat dengan cara tertentu dan dapat ada dalam berbagai bentuk, seperti angka, teks, dll. Ketika digunakan dalam konteks media transmisi, data merujuk ke informasi dalam format digital biner. Data adalah istilah luas dalam teknologi komputer, tetapi sering digunakan untuk mengidentifikasi dan memisahkan informasi dari bit belaka. Dalam telekomunikasi, data sering merujuk pada informasi digital, bukan analog. Tidak seperti transmisi analog, yang memerlukan koneksi garis keras selama durasi transmisi, data digital dikirim dalam paket (Sun, Zhang, Xiong, & Zhu, 2014).

Dalam komputasi, data adalah informasi yang telah diterjemahkan ke dalam bentuk yang efisien untuk pergerakan atau pemrosesan. Relatif terhadap komputer dan media transmisi saat ini, data adalah informasi yang diubah menjadi bentuk digital biner. Data dapat diterima untuk digunakan sebagai subjek tunggal atau

subjek jamak. Data mentah adalah istilah yang digunakan untuk menggambarkan data dalam format digital paling dasar.

Konsep data dalam konteks komputasi berakar pada karya Claude Shannon, seorang ahli matematika Amerika yang dikenal sebagai bapak teori informasi. Dia mengantarkan konsep digital biner berdasarkan penerapan logika Boolean dua nilai ke sirkuit elektronik. Format digit biner mendasari CPU, memori semikonduktor dan disk drive, serta banyak perangkat periferal yang umum dalam komputasi saat ini. Input komputer awal untuk kontrol dan data berupa kartu punch, diikuti oleh pita magnetik dan hard disk.

Pada awalnya, pentingnya data dalam komputasi bisnis menjadi jelas dengan popularitas istilah "pemrosesan data" dan "pemrosesan data elektronik," yang, untuk beberapa waktu, datang untuk mencakup keseluruhan dari apa yang sekarang dikenal sebagai teknologi informasi. Selama sejarah komputasi perusahaan, spesialisasi terjadi, dan profesi data yang berbeda muncul seiring dengan pertumbuhan pemrosesan data perusahaan.

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 Keamanan Data

Keamanan data adalah seperangkat standar dan teknologi yang melindungi data dari kehancuran, modifikasi, atau pengungkapan yang disengaja atau tidak disengaja. Keamanan data dapat diterapkan dengan menggunakan berbagai teknik dan teknologi, termasuk kontrol administratif, keamanan fisik, kontrol logis, standar organisasi, dan teknik perlindungan lainnya yang membatasi akses ke pengguna atau proses yang tidak sah atau berbahaya (Rao & Selvamani, 2015).

Keamanan data mengacu pada langkah-langkah privasi digital pelindung yang diterapkan untuk mencegah akses tidak sah ke komputer, database, dan situs web. Keamanan data juga melindungi data dari korupsi. Keamanan data adalah aspek penting dari TI untuk organisasi dari berbagai ukuran dan tipe. Keamanan data juga dikenal sebagai keamanan informasi atau keamanan komputer.

Contoh teknologi keamanan data termasuk backup, masking data dan penghapusan data. Ukuran teknologi keamanan data utama adalah enkripsi, di mana data digital, perangkat lunak / perangkat keras, dan hard drive dienkripsi dan karenanya tidak dapat dibaca oleh pengguna dan peretas yang tidak sah. Salah satu metode yang paling umum dijumpai dalam mempraktikkan keamanan data adalah penggunaan otentikasi. Dengan otentikasi, pengguna harus memberikan kata sandi,

kode, data biometrik, atau bentuk data lainnya untuk memverifikasi identitas sebelum akses ke sistem atau data diberikan. Keamanan data juga sangat penting untuk catatan perawatan kesehatan, sehingga pendukung kesehatan dan praktisi medis di AS dan negara-negara lain berupaya menerapkan privasi rekam medis elektronik dengan menciptakan kesadaran tentang hak-hak pasien terkait dengan pelepasan data ke laboratorium, dokter, rumah sakit dan fasilitas medis lainnya.

2.3.1 Pentingnya Keamanan Data

Semua bisnis saat ini menangani data hingga taraf tertentu. Dari raksasa perbankan yang menangani data pribadi dan keuangan dalam volume besar hingga bisnis satu orang yang menyimpan detail kontak pelanggannya di ponsel, data berperan di perusahaan baik besar maupun kecil.

Tujuan utama keamanan data adalah untuk melindungi data yang dikumpulkan, disimpan, diterima, atau ditransmisikan oleh suatu organisasi. Kepatuhan juga merupakan pertimbangan utama. Tidak masalah perangkat, teknologi, atau proses mana yang digunakan untuk mengelola, menyimpan, atau mengumpulkan data, itu harus dilindungi. Pelanggaran data dapat menyebabkan kasus litigasi dan denda yang sangat besar, belum lagi kerusakan reputasi organisasi. Pentingnya melindungi data dari ancaman keamanan lebih penting saat ini daripada sebelumnya.

Keamanan data mengacu pada proses melindungi data dari akses yang tidak sah dan korupsi data sepanjang siklus hidupnya. Keamanan data termasuk enkripsi data, tokenization, dan praktik manajemen kunci yang melindungi data di semua

aplikasi dan platform. Organisasi di seluruh dunia banyak berinvestasi dalam kemampuan pertahanan cyber teknologi informasi untuk melindungi aset penting mereka. Apakah suatu perusahaan perlu melindungi merek, modal intelektual, dan informasi pelanggan atau menyediakan kontrol untuk infrastruktur penting, sarana untuk mendeteksi insiden dan merespons melindungi kepentingan organisasi memiliki tiga elemen umum: orang, proses, dan teknologi.

2.3.2 Solusi Keamanan Data

Data membutuhkan enkripsi dalam mengamankan informasi yang ada dalam data tersebut. Dengan enkripsi data canggih, tokenization, dan manajemen utama untuk melindungi data di seluruh aplikasi, transaksi, penyimpanan, dan platform big data, Teknik ini menyederhanakan perlindungan data sensitif bahkan dalam kasus penggunaan yang paling kompleks sekalipun. Beberapa model keamanan data antara lain:

1. Keamanan akses cloud - Platform perlindungan yang memungkinkan Anda untuk pindah ke cloud dengan aman sambil melindungi data dalam cloud.
2. Enkripsi data - Solusi keamanan data-sentris dan tokenisasi yang melindungi data di lingkungan perusahaan, cloud, seluler, dan data besar.
3. Modul keamanan perangkat keras - Modul keamanan perangkat keras yang menjaga data keuangan dan memenuhi persyaratan keamanan dan kepatuhan industri.
4. Manajemen kunci - Solusi yang melindungi data dan memungkinkan kepatuhan regulasi industri.

5. Enterprise Data Protection - Solusi yang menyediakan pendekatan data-centric end-to-end untuk perlindungan data perusahaan.
6. Keamanan Pembayaran - Solusi menyediakan enkripsi dan tokenisasi point-to-point lengkap untuk transaksi pembayaran ritel, memungkinkan pengurangan lingkup PCI.
7. Big Data, Hadoop, dan perlindungan data IofT - Solusi yang melindungi data sensitif di Danau Data - termasuk Hadoop, Teradata, Micro Focus Vertica, dan platform Big Data lainnya.
8. Keamanan Aplikasi Seluler - Melindungi data sensitif di aplikasi seluler asli sembari menjaga data dari ujung ke ujung.
9. Keamanan Peramban Web - Melindungi data sensitif yang diambil di peramban, dari titik pelanggan memasukkan pemegang kartu atau data pribadi dan menjaganya agar tetap terlindungi melalui ekosistem ke tujuan tuan rumah tepercaya.
10. eMail Security - Solusi yang menyediakan enkripsi ujung ke ujung untuk email dan olahpesan seluler, menjaga informasi pribadi dan informasi kesehatan pribadi tetap aman dan pribadi.

2.3.3 Kerahasiaan

Kerahasiaan mengacu pada melindungi informasi agar tidak diakses oleh pihak yang tidak berwenang. Dengan kata lain, hanya orang yang diberi wewenang untuk melakukannya yang dapat memperoleh akses ke data sensitif. Bayangkan catatan bank harus dapat diakses, tentu saja, dan karyawan di bank yang membantu

dalam menjalankan transaksi harus dapat mengaksesnya, tetapi tidak ada orang lain yang seharusnya. Kegagalan untuk menjaga kerahasiaan berarti bahwa seseorang yang seharusnya tidak memiliki akses telah berhasil mendapatkannya, melalui perilaku yang disengaja atau karena kecelakaan. Kegagalan kerahasiaan seperti itu, umumnya dikenal sebagai pelanggaran, biasanya tidak dapat diperbaiki. Setelah rahasia itu terungkap, tidak ada cara untuk mengetahuinya. Jika catatan bank diposting di situs web publik, semua orang dapat mengetahui nomor rekening bank, saldo, dll., Informasi itu tidak dapat dihapus dari pikiran, kertas, komputer, dan tempat lain mereka. Hampir semua insiden keamanan utama yang dilaporkan di media saat ini melibatkan kerugian besar kerahasiaan. Jadi, secara ringkas, pelanggaran kerahasiaan berarti bahwa seseorang memperoleh akses ke informasi yang seharusnya tidak memiliki akses ke sana.

2.3.4 Integritas

Integritas mengacu pada memastikan keaslian informasi — bahwa informasi tidak diubah, dan bahwa sumber informasi itu asli. Bayangkan jika seseorang memiliki situs web dan Anda menjual produk di situs itu. Sekarang bayangkan penyerang dapat berbelanja di situs web dan dengan jahat mengubah harga produk Anda sehingga mereka dapat membeli apa pun dengan harga berapa pun yang mereka pilih. Itu akan menjadi kegagalan integritas karena informasi dalam hal ini, harga suatu produk telah diubah dan perubahan ini tidak dapat digagalkan. Contoh lain dari kegagalan integritas adalah ketika seseorang mencoba

terhubung ke situs web dan penyerang jahat antara Anda dan situs web mengalihkan lalu lintas ke situs web yang berbeda. Dalam hal ini, situs yang dituju tidak asli.

2.3.5 Ketersediaan

Ketersediaan berarti informasi dapat diakses oleh pengguna yang berwenang. Jika penyerang tidak dapat mengkompromikan dua elemen pertama dari keamanan informasi (lihat di atas) mereka dapat mencoba melakukan serangan seperti penolakan layanan yang akan menurunkan server, membuat situs web tidak tersedia untuk pengguna yang sah karena kurangnya ketersediaan.

2.3.6 Kontrol Akses

Kesalahan terbesar yang bisa dilakukan oleh perancang aplikasi adalah mengabaikan kontrol akses sebagai bagian dari fungsionalitas yang diperlukan. Jarang bahwa setiap pengguna atau sistem yang berinteraksi dengan suatu aplikasi harus memiliki hak yang sama di seluruh aplikasi itu. Beberapa pengguna mungkin memerlukan akses ke data tertentu dan bukan yang lain; beberapa sistem harus atau tidak dapat mengakses aplikasi. Akses ke komponen, fungsi, atau modul tertentu dalam aplikasi juga harus dikontrol. Kontrol akses juga penting untuk kepatuhan audit dan peraturan. Beberapa cara umum mengelola kontrol akses adalah:

1. Baca, tulis, dan jalankan hak istimewa: File
2. Kontrol akses berbasis peran: administrator, pengguna
3. Alamat IP akses berbasis host, nama mesin
4. Objek kode kontrol akses tingkat objek, banyak pembaca / penulis tunggal

2.4 Algoritma

Untuk membuat komputer melakukan apa pun, seseorang harus menulis program komputer. Untuk menulis program komputer, seseorang harus memberi tahu komputer, langkah demi langkah, persis apa yang seseorang inginkan. Komputer kemudian "mengeksekusi" program, mengikuti setiap langkah secara mekanis, untuk mencapai tujuan akhir. Ketika seseorang memberi tahu komputer apa yang harus dilakukan, seseorang juga harus memilih bagaimana melakukannya. Di situlah algoritma komputer masuk. Algoritma adalah teknik dasar yang digunakan untuk menyelesaikan pekerjaan (Gurevich, 2012). Mari kita ikuti contoh untuk membantu mendapatkan pemahaman tentang konsep algoritma. Katakanlah seseorang memiliki seorang teman yang tiba di bandara, dan teman seseorang perlu pergi dari bandara ke rumah. Berikut adalah empat algoritma berbeda yang mungkin akan diberikan kepada orang lain untuk sampai ke rumah:

1. Algoritma taksi:
    a. Pergi ke tempat taksi.
    b. Naik taksi.
    c. Berikan alamat saya pada pengemudi.

2. Algoritma panggilan-saya:
    a. Ketika pesawat Anda tiba, hubungi ponsel saya.
    b. Temui saya di luar klaim bagasi.

3. Algoritma rent-a-car:
    a. Naik shuttle ke tempat rental mobil.
    b. Menyewa mobil.
    c. Ikuti petunjuk untuk sampai ke rumah saya.

4. Algoritma bus:
    a. Di luar klaim bagasi, naik bus nomor 70.
    b. Transfer ke bus 14 di Main Street.
    c. Turun di Elm street.
    d. Berjalanlah dua blok ke utara ke rumah saya.

Keempat algoritma ini mencapai tujuan yang persis sama, tetapi masing-masing algoritma melakukannya dengan cara yang sama sekali berbeda. Setiap algoritma juga memiliki biaya dan waktu perjalanan yang berbeda. Naik taksi, misalnya, mungkin adalah cara tercepat, tetapi juga yang paling mahal. Naik bus jelas lebih murah, tetapi jauh lebih lambat. Anda memilih algoritma berdasarkan keadaan.

Dalam pemrograman komputer, seringkali ada banyak cara berbeda - algoritma - untuk menyelesaikan tugas yang diberikan. Setiap algoritma memiliki kelebihan dan kekurangan dalam situasi yang berbeda. Penyortiran adalah satu tempat di mana banyak penelitian telah dilakukan karena komputer menghabiskan banyak daftar penyortiran waktu. Berikut adalah lima algoritma berbeda yang digunakan dalam penyortiran:

1. Bin sort
2. Gabungkan semacam
3. Semacam gelembung
4. Semacam shell
5. Quicksort

Jika ada sejuta nilai integer antara 1 dan 10 dan perlu diurutkan, jenis bin sort adalah algoritma yang tepat untuk digunakan. Jika Anda memiliki sejuta judul buku, quicksort mungkin merupakan algoritma terbaik. Dengan mengetahui kekuatan dan kelemahan dari berbagai algoritma, Anda memilih yang terbaik untuk tugas yang ada.

2.4.1 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.

2.4.2 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."

2.4.3 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.

2.5 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.

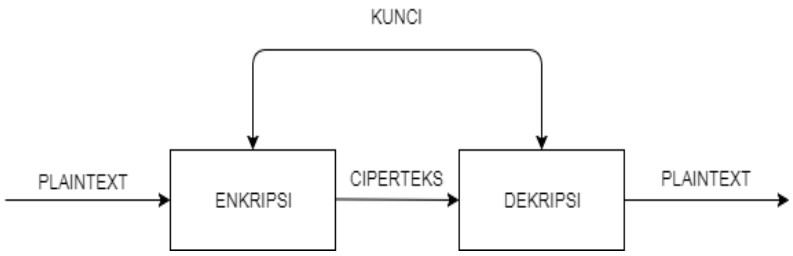
7. Kriptanalisis

   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.

2.5.1 Kriptografi Simetris

Algoritma kriptografi disebut juga *cipher* yaitu aturan untuk *enchipering* dan *dechipering* atau fungsi matematika yang digunakan untuk enkripsi dan deskripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk *enciphering* dan *dechiphering*. Algoritma Simetris sering disebut dengan algoritma klasik, karena memakai kunci yang sama untuk kegiatan enkripsi dan deskripsinya. Gambar 2.1 adalah skema algoritma simetris.

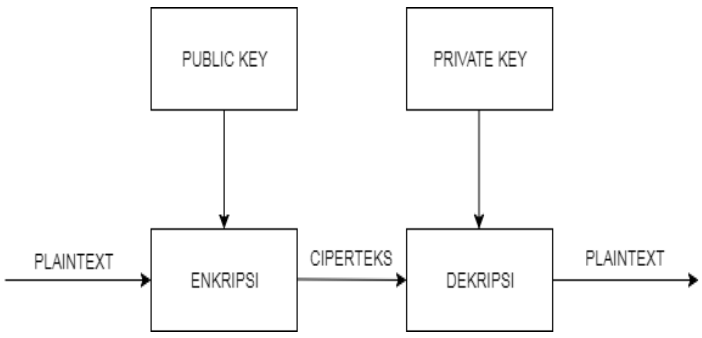


Gambar 2.1 Skema kriptografi simetris
Sumber: (Putri, Setyorini, & Rahayani, 2018)

2.5.2 Kriptografi Asimetris

Algoritma tak simetris sering juga disebut dengan algoritma kunci publik, dengan arti kata kunci yang digunakan untuk melakukan enkripsi dan dekripsinya berbeda (Ayushi, 2010) (S., L. Ribeiro, & David, 2012). Pada algoritma tak simetri kunci terbagi menjadi 2 (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)

2.6 Enkripsi dan Dekrpsi

Menurut M. Miftakhul Amin, enkripsi adalah sebuah proses penyandian yang melakukan perubahan sebuah kode (pesan) dari yang bisa dimengerti (plaintext) menjadi sebuah kode yang tidak bisa dimengerti (chipertext). Sedangkan proses kebalikannya untuk mengubah chipertext menjadi plaintext disebut dekripsi. Proses enkripsi dan deskripsi memerlukan suatu mekanisme dan kunci tertentu. Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Dekripsi menggunakan kunci dekripsi mendapatkan kembali data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa parameter. Biasanya algoritma tidak dirahasiakan, bahkan enkripsi yang mengandalkan kerahasiaan algoritma dianggap sesuatu yang tidak baik. Rahasia terletak di beberapa parameter yang digunakan jadi kunci ditentukan oleh parameter (Amin, 2016).

2.7 Caesar Cipher

Caesar Cipher adalah salah satu dari sandi yang paling awal diketahui dan paling sederhana. Ini adalah jenis sandi pengganti di mana setiap huruf dalam plaintext 'digeser' sejumlah tempat ke bawah alfabet. Misalnya, dengan pergeseran 1, A akan digantikan oleh B, B akan menjadi C, dan seterusnya. Metode ini dinamai setelah Julius Caesar, yang tampaknya menggunakannya untuk berkomunikasi dengan jenderalnya.

Skema enkripsi yang lebih kompleks seperti cipher Vigenère menggunakan cipher Caesar sebagai salah satu elemen dari proses enkripsi. 'Enkripsi' ROT13 yang dikenal luas hanyalah cipher Caesar dengan offset 13. Cipher Caesar pada dasarnya tidak menawarkan keamanan komunikasi, dan akan ditunjukkan bahwa ia dapat dengan mudah dipatahkan bahkan dengan tangan (Pratama & Tamatjita, 2015).

Untuk mengirimkan pesan terenkripsi dari satu orang ke orang lain, pertama-tama perlu bahwa kedua belah pihak memiliki 'kunci' untuk sandi, sehingga pengirim dapat mengenkripsi dan penerima dapat mendekripsi. Untuk sandi caesar, kuncinya adalah jumlah karakter untuk menggeser alfabet sandi.

Berikut adalah contoh cepat langkah enkripsi dan dekripsi yang terlibat dengan caesar cipher. Teks yang akan kami enkripsi adalah 'membela dinding timur kastil', dengan pergeseran (kunci) 1.

```
plaintext:  defend the east wall of the castle

ciphertext: efgfoe uif fbtu xbmm pg uif dbtumf
```

Sangat mudah untuk melihat bagaimana setiap karakter dalam plaintext digeser ke atas alfabet. Dekripsi sama mudahnya, dengan menggunakan offset -1.

```
plain:  abcdefghijklmnopqrstuvwxyz

cipher: bcdefghijklmnopqrstuvwxyza
```

Jelas, jika kunci yang berbeda digunakan, alfabet sandi akan digeser dengan jumlah yang berbeda. Deskripsi Matematika menjelaskan bahwa sandi Caesar menerjemahkan semua karakter ke angka, 'a' = 0, 'b' = 1, 'c' = 2, ..., 'z' = 25. Diwakili bahwa fungsi enkripsi sandi Caesar, e (x), di mana x adalah karakter yang telah dienkripsi, sebagai:

$$e(x) = (x + k)\ mod\ 256$$

Di mana k adalah kunci (shift) yang diterapkan pada setiap huruf. Setelah menerapkan fungsi ini hasilnya adalah angka yang kemudian harus diterjemahkan kembali menjadi surat. Fungsi dekripsi adalah:

$$d(x) = (x - k)\ mod\ 256$$

2.8 Algoritma Base64

Menurut Ahir Yugo Nugroho, transformasi *Base64* merupakan salah satu algoritma untuk *encoding* dan *decoding* suatu data ke dalam format *ASCII*, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metoda yang digunakan untuk melakukan encoding (penyandian) terhadap data *binary*. Karakter yang dihasilkan pada transformasi *Base64* ini terdiri dari A....Z, a....z dan 0....9, serta ditambah dengan dua karakter terakhir yang bersimbol yaitu + dan / serta satu buah karakter sama dengan (=) yang digunakan untuk penyesuaian dan menggenapkan data binary atau istilahnya disebut sebagai pengisi pad. Karakter

simbol yang akan dihasilkan akan tergantung dari proses algoritma yang berjalan. Kriptografi transformasi *Base64* banyak digunakan di dunia internet sebagai media data format untuk mengirimkan data, ini dikarenakan hasil dari *Base64* berupa *plaintext*, maka data ini akan jauh lebih mudah dikirim, dibandingkan dengan format data yang berupa *binary* (Nugroho, 2015). Dalam implementasinya beberapa contoh dalam transformasi *Base64*, yang antara lain adalah sebagai berikut:

1. *PEM (Privacy-Enhaced Mail)* adalah protocol pertama dengan teknik *Base64* yang didasarkan pada *RFC 989*, yang terdiri dari 7 karakter (7-bit) yang digunakan pada *SMTP* dalam transfer data tapi untuk sekarang *PEM* sudah tidak menggunakan *RFC 989* tapi sudah diganti dengan *RFC 1421* yang menggunakan karakter A....Z, a....z, 0....9.
2. *MIME (Multi Purpose Mail Extension)* didasarkan pada *RFC 2045*. Teknik *encoding Base64 MIME*, mempunyai konsep yang berdasarkan *RFC 1421* versi *PEM*. Sedangkan *MIME* diakhiri dengan padding "=" pada hasil akhir *encoding*.
3. *UTF-7* didasarkan pada *RFC 2152*, yang umumnya disebut *"MODIFICATION BASE" UTF-7* menggunakan karakter *MIME*, tidak memakai *padding* "=", karakter "=" digunakan sebagia *escape* untuk *encoding*.

Dalam menentukan karakter yang akan digunakan pada format Base64, karakter hasil pemecahan bit akan ditentukan berdasarkan tabel berikut ini.

Tabel 2.1 Daftar karakter pada Base64

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	G	48	w
1	B	17	R	33	H	49	x
2	C	18	S	34	I	50	y
3	D	19	T	35	J	51	z
4	E	20	U	36	K	52	0
5	F	21	V	37	L	53	1
6	G	22	W	38	M	54	2
7	H	23	X	39	N	55	3
8	I	24	Y	40	O	56	4
9	J	25	Z	41	P	57	5
10	K	26	a	42	Q	58	6
11	L	27	b	43	R	59	7
12	M	28	c	44	S	60	8
13	N	29	d	45	T	61	9
14	O	30	e	46	U	62	+
15	P	31	f	47	V	63	/

Sumber: (Nugroho, 2015)

Teknik *encoding Base64* sebenarnya sederhana, jika ada satu (*string*) *bytes* yang akan disandikan ke *Base64* maka caranya adalah:

1. Pecah *string bytes* tersebut ke per-3 *bytes*.

2. Gabungkan 3 bytes menjadi 24 bit. Dengan catatan 1 *bytes* = 8 bit, sehingga 3 x 8 =24 bit.
3. Lalu 24 bit yang disimpan di-*buffer* (disatukan) dipecah-pecah menjadi 6 bit, maka akan menghasilkan 4 pecahan.
4. Masing masing pecahan diubah ke dalam nilai *decimal*, dimana maksimal nilai 6 bit dalah 63.
5. Terakhir, jadikan nilai-nilai desimal tersebut menjadi indeks untuk memilih karakter penyusun dari *base64* dan maksimal adalah 63 atau indeks ke 64.

2.9 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).

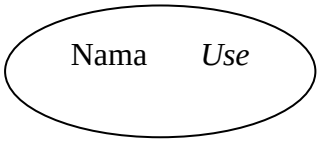
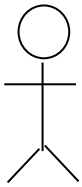
2.9.1 Use Case Diagram

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)

2.9.2 Activity Diagram

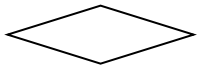
Menurut Indra Griha Tofik Isa dan George Pri Hartawan, Activity Digram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk

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)

2.10 File

File adalah entitas dari data yang disimpan di dalam sistem *file* yang dapat diakses dan diatur oleh pengguna. Sebuah *file* memiliki nama yang unik dalam direktori dimana ia berada. Alamat direktori dimana suatu berkas ditempatkan diistilahkan dengan *path* (Pabokory, Astuti, & Kridalaksana, 2015).

Sebuah *file* berisi aliran data atau data *stream* yang berisi sekumpulan data yang saling berkaitan serta atribut berkas yang disebut dengan properties yang berisi informasi mengenai *file* yang bersangkutan seperti informasi mengenai kapan sebuah berkas dibuat.

Mengacu pada pengertian *file* diatas, kata *file* ini diperuntukkan pada bidang komputerisasi. Setelah memahami apa itu *file*, tentunya kita juga perlu mengetahui apa saja jenis *file* yang ada di komputer. Jenis-jenis *file* yang ada di dalam komputer ada banyak sekali formatnya. Berikut ini adalah jenis-jenis *file* dalam komputer berdasarkan format filenya:

1. *File* Sistem

   Beberapa ekstensi dalam *file* sistem diantaranya adalah sys, com, bak, bat, tmp, dan exe. *File* sistem ini berfungsi untuk menjalankan program di dalam komputer sesuai dengan peruntukkannya dan juga menjalankan berbagai aplikasi yang diinstal ke dalam komputer.

2. *File* Video

   Beberapa ekstensi pada *file* video adalah *mpg*, *wmv, mp4, 3gp*, *avi*, *flv, KV*. Masing-masing ekstensi ini menunjukkan bahwa masing-masing video memiliki jenis pemutar yang berbeda. Tidak semua jenis video dapat diputar dengan *software* yang biasanya terinstal didalam komputer. Ada beberapa jenis video yang hanya bisa diputar dengan *software* tertentu.

3. *File* Dokumen

   Beberapa ekstensi pada *file* dokumen diantaranya adalah *doc, odt, doc, xls, ods, pdf, ppt, txt*. Masing-masing ekstensi tersebut menunjukkan jenis *file* dokumennya dan hanya bisa dibuka jika di dalam komputer terinstal *software* atau aplikasi yang sesuai.

4. *File* Gambar

   Beberapa ekstensi *file* gambar diantaranya adalah jpg, jpeg, png, gif, tif dan lain-lain. Pada umumnya gambar yang dihasilkan oleh kamera digital ataupun kamera manual akan berekstensi jpg atau jpeg. Gambar berekstensi *tif, png* dan lainnya biasanya hasil penyimpanan dari *software* tertentu, misalnya *Photoshop, CorelDraw, AutoCad* dan lain-lain.

5. *File* Suara

   Beberapa ekstensi *file* suara diantaranya adalah *wav*, *mp3*, *midi*, dan *rm*. Sama halnya dengan *file* komputer lainnya, tidak semua *file* suara dapat dibukan dengan satu aplikasi.

2.11 Video

Video adalah teknologi pemrosesan sinyal elektronik mewakilkan gambar bergerak (Shi, Renwick, Turner, & Kirsh, 2019). Jadi audio video adalah teknologi yang mewakili pemrosesan pesan (pita suara atau piringan suara) dalam bentuk auditif dan gerak gambar. Video memiliki berbagai macam jenis, adapun jenis-jenis dari video diantaranya yaitu:

1. *WVM*

   Format video ini diciptakan oleh *Microsoft*. *Windows Media Player* merupakan *software* buatan *Microsoft* untuk memutar *file* video tersebut. Ditinjau dari aspek kualitas video file *WMV* biasanya tak terlalu besar. Karenanya, bila dibandingkan dengan video *MOV*, kualitas video *WMV* sebenarnya kurang mendetail. Karena berukuran kecil, *file WMV* begitu praktis untuk diupload ataupun diunduh.

2. *MOV*

   Format video ini diciptakan oleh perusahaan *Apple*. *QuickTime* merupakan *software* buatan *Apple* untuk memutar *file* video tersebut. Ada juga *software QuickTime* yang dapat dijalankan dengan menggunakan sistem operasi *Windows*. Lalu, apa keunikan format *file* video MOV? Ciri khas file video

MOV yaitu tampilan video yang berkualitas sangat baik. Format *file* video *MOV* pun biasanya berukuran besar.

3. *FLV*

   Adobe menciptakan format video *FLV*. Video *FLV* pun sering diunduh di beberapa media sosial seperti *YouTube*. Keunikan *file FLV* yaitu tak sedikit pengunggah video yang men-*convert* video format *FLV* menjadi video format lainnya yang berukuran lebih kecil. Misalnya, agar ukuran *file* video menjadi lebih kecil, meng-convertnya menjadi format *WMV*. Meskipun di-convert menjadi video format yang berukuran lebih kecil, tak mengurangi kualitas tampilan video.

4. *MP4*

   Apa keunikan format *file* video *MP4*? Format video yang diperkenalkan pada tahun 1998 menggunakan kompresi video dan audio yang berbeda. Kompresi yaitu memadatkan file agar berukuran menjadi lebih kecil. Nah, kompresi gambar video *MP4* menggunakan kompresi *H.264*. Sedangkan kompresi audio menggunakan kompresi *AAC*. Kompresi H.264 yaitu sejenis kompresi yang menghasilkan video berkualitas baik meski *file* berukuran kecil. Sedangkan kompresi *AAC* yaitu sejenis kompresi yang menghasilkan suara berkualitas baik meskipun *file* berukuran kecil.

5. *AVI*

   File video *AVI* dapat disebut generasi tua *file* video. Format video ini diciptakan oleh *Microsoft* pada tahun 1992. Karakteristik video *AVI* yaitu tak dikompresi. Karena tak dikompresi, video *AVI* terlihat lebih baik dan

berkualitas. Namun, ukuran *file AVI* menjadi relatif besar. Dalam hal penggunaannya, video *AVI* kurang cocok untuk kita upload dimedia sosial. Sebabnya, karena berukuran besar, membutuhkan waktu lebih lama untuk menguploadnya. Bila kita men-d*ownload*nya, tentunya membutuhkan waktu yang lebih lama.

2.12 Visual Basic.Net 2010

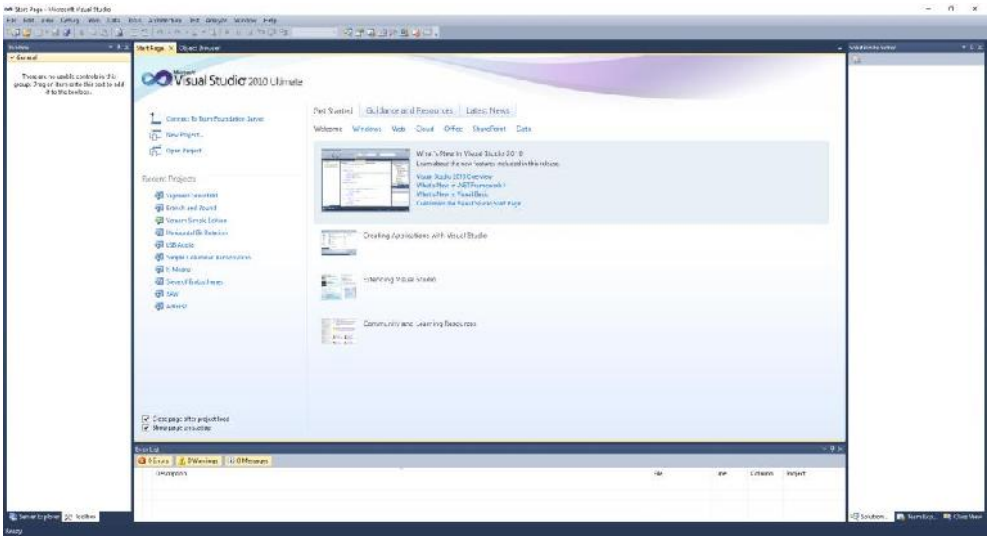
Bahasa Pemrograman *Microsoft Visual Basic .NET* adalah sebuah bahasa pemrograman tingkat tinggi untuk *Microsoft .NET Framework*. Walaupun *VB.NET* ini memang dibuat supaya mudah dipahami dan dipelajari, namun bahasa pemrograman ini juga cukup *powerful* untuk memenuhi kebutuhan dari *programmer* yang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

2.12.1 Lingkungan kerja Visual Basic.Net

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

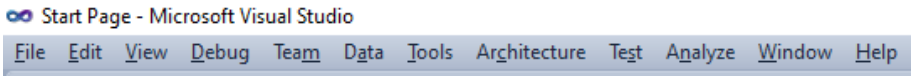
2.12.2 Komponen Visual Basic.Net

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,*

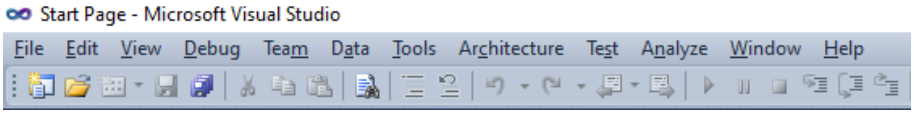
data, tools, window dan *help*. *Menu bar* pada *Visual Sudio* 2010 seperti terlihat pada gambar 2.5.



Gambar 2.4 Tampilan Menu Bar

2. Toolbar

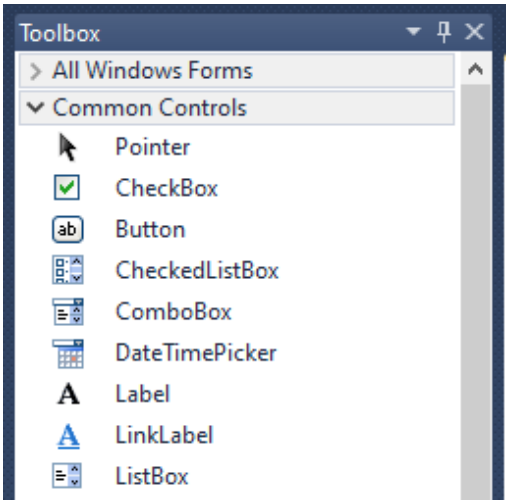
   Fasilitas ini dapat mempercepat pengaksesan perintah-perintah yang ada dalam pemrograman seperti terlihat pada gambar 2.6.



Gambar 2.5 Tampilan Toolbar

3. Toolbox

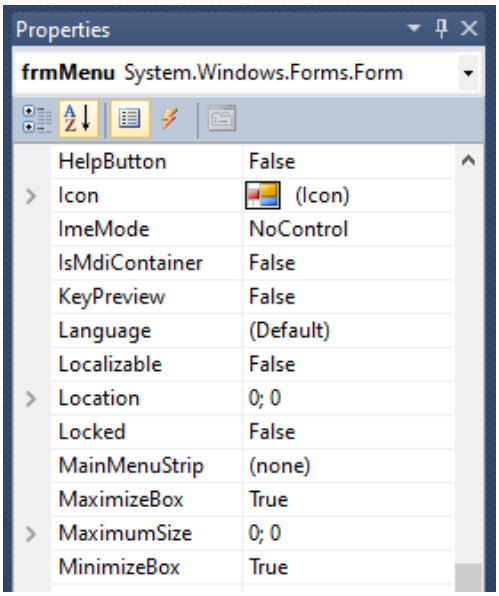
   Sebuah *window* yang berisi tombol-tombol kontrol yang akan Anda gunakan untuk mendesain atau membangun sebuah *form* atau *report* seperti terlihat pada gambar 2.7.



Gambar 2.6 Tampilan Toolbox

4. Properties Window

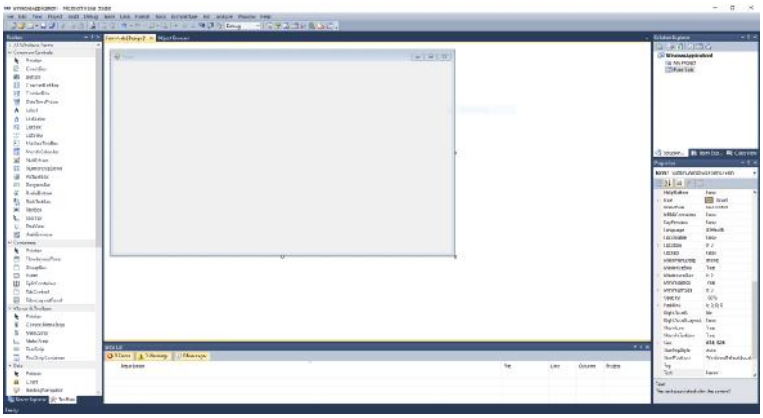
   Properties window adalah tempat menyimpan *property* dari setiap objek control dan komponen.



Gambar 2.7 Tampilan Properties

5. Form

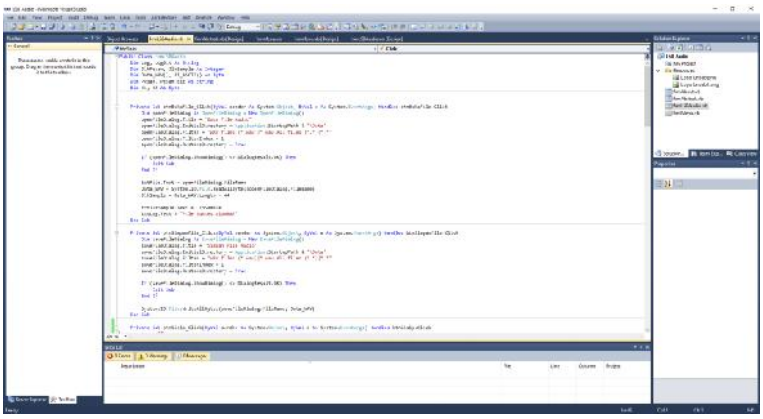
   Form merupakan tempat di mana kontrol-kontrol diletakkan. Form juga berfungsi sebagai tempat pembuatan tampilan atau antarmuka (*user interface)* dari sebuah aplikasi *windows*.



Gambar 2.8 Tampilan Form

6. Code Editor

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



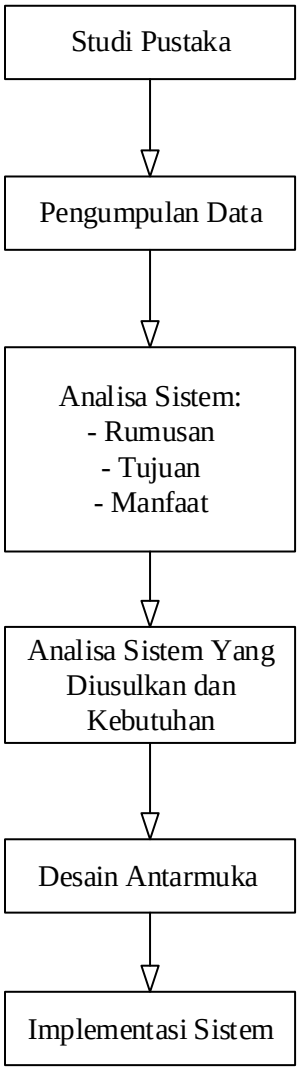
Gambar 2.9 Tampilan Code Editor

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian adalah bagaimana alur penelitian itu dilakukan. Tahapan dilakukan dengan cara mengelompokkan tugas menjadi beberapa fase.



Gambar 3.1 Tahapan Penelitian

Berikut merupakan penjelasan dari gambar tahapan penelitian yang ada di atas:

1. Studi pustaka, dalam skripsi ini penulis ambil dari beberapa sumber seperti jurnal dan buku.
2. Pengumpulan data, dalam skripsi ini penulis mengumpulkan data dengan cara mencari dan mengunduh video dengan ukuran maksimal 30Mb.
3. Analisa sistem, masalah yang diangkat dalam skripsi ini ialah bagaimana cara mengamankan suatu *file* video dengan menggunakan algoritma *Base64*.
4. Analisa sistem usulan, penulis akan membuat suatu sistem yang dapat digunakan dalam mengenkripsi dan mendekripsi *file* video agar dapat dikirimkan secara lebih aman.
5. Analisa kebutuhan, untuk membuat sistem ini penulis membutuhkan beberapa perangkat keras dan perangkat lunak seperti *software visual studio code* dan laptop.
6. Metode, metode algoritma yang penulis gunakan dalam penulisan skripsi ini ialah metode *Base64*.
7. Desain sistem, penulis memulai proses mendesain sistem dengan menggunakan *UML* agar terlihat alur proses data *file* video yang akan dienkripsi ataupun didekripsi.
8. Pembuatan sistem, penulis membuat sistem dengan menggunakan bahasa pemrograman Microsoft Visual Basic.Net 2010.

9. Implementasi, setelah pembuatan sistem selesai, penulis mengimplementasikan sistem dengan cara mencoba dan melakukan evaluasi apakah terdapat kesalahan atau sudah berjalan dengan benar.

3.2 Metode Pengumpulan Data

Metode pengumpulan data dilakukan untuk mendapatkan informasi tentang kebutuhan sistem. Metode ini dilakukan dengan beberapa cara antara lain:

1. Studi Pustaka

   Pengumpulan data-data berupa teori mencari dan mengumpulkan bahan yang berhubungan dengan masalah yang sedang diteliti.

2. Studi Lapangan

   Studi lapangan yaitu pengumpulan data secara langsung ke lapangan dengan menggunakan teknik pengumpulan data.

3. Observasi

   Observasi merupakan teknik yang digunakan untuk mengumpulkan data dengan cara melakukan pengamatan secara langsung terhadap cara kerja dari enkripsi dan dekripsi pada *file* video.

3.3 Analisa Sistem

Analisa sistem adalah penguraian sistem informasi yang terbagi ke dalam bagian-bagian komponen dengan tujuan untuk mengidentifikasi masalah-masalah dan mengevaluasi permasalahan yang terjadi sehingga diharapkan atau dapat diusulkan.

Kegiatan analisa adalah sebuah sistem informasi dengan tujuan untuk mengidentifikasi serta mengevaluasi masalah yang akan muncul, yang mungkin akan terjadi sehingga menjadi kebutuhan yang diharapkan serta perkembangan teknologi.

Saat ini banyak dari proses pengiriman video tidak dienkripsi sehingga *file* video tersebut mudah dilihat oleh siapapun. Dengan tidak amannya pengiriman *file* video ini seringkali informasi *file* tersebut dapat dilihat secara umum oleh siapapun sehingga tingkat kerahasiaan informasi tersebut tidak terjaga. Atas dasar ini penulis akan membuat suatu sistem yang dapat mengenkripsi dan mendekripsi suatu *file* video sehingga informasi yang terkandung di dalamnya menjadi lebih aman.

3.3.1 Analisa Sistem Yang Berjalan

Saat ini baik dalam proses pengiriman ataupun penerimaan video masih tidak menggunakan enkripsi yang artinya informasi dari video tersebut dapat dilihat siapapun secara bebas tanpa harus mendekripsi video tersebut dahulu. Hal ini membuat informasi yang terkandung pada video tersebut dapat dilihat oleh siapapun dan mengurangi kerahasiaan informasi dari video tersebut. Penggunaan enkripsi pada video juga dibutuhkan untuk memperkecil ukuran *byte* pada video sehingga dapat dikirim dan diterima secara cepat.

3.3.2 Analisa Sistem Yang Diusulkan

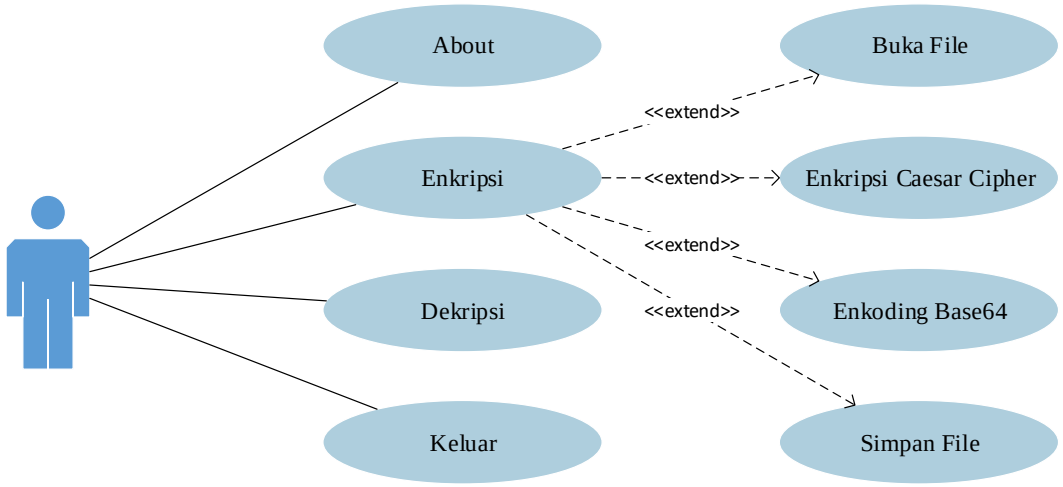
Pada sistem yang penulis buat, sistem dapat melakukan enkripsi dan dekripsi pada video dengan format MP4 dengan cara menambahkan algoritma

Caesar Cipher pada proses kriptografi. Pada sistem ini, pengguna dapat memilih video apapun dengan format MP4 dan maksimal kapasitas 20Mb. Pada penggunaan enkripsi base64 di sistem, sistem akan mengenkripsi byte dari video tersebut sehingga video tidak dapat dibuka meskipun nama file dari video tersebut diganti. Selain mengenkripsi byte dari video tersebut, sistem juga akan mengenkripsi nama file video sehingga video mendapatkan pengamanan tambahan pada sisi nama file. Algoritma base64 akan diimplementasikan pada saat pengguna mulai memproses enkripsi atau dekripsi pada video yang mereka pilih. Pengguna juga dapat mengunduh hasil enkripsi atau dekripsi dari video yang mereka pilih.

3.4 Rancangan Sistem Secara Global

3.4.1 Use Case Diagram Enkripsi

Berikut ini adalah use case diagram yang digunakan dalam melakukan proses enkripsi file video:

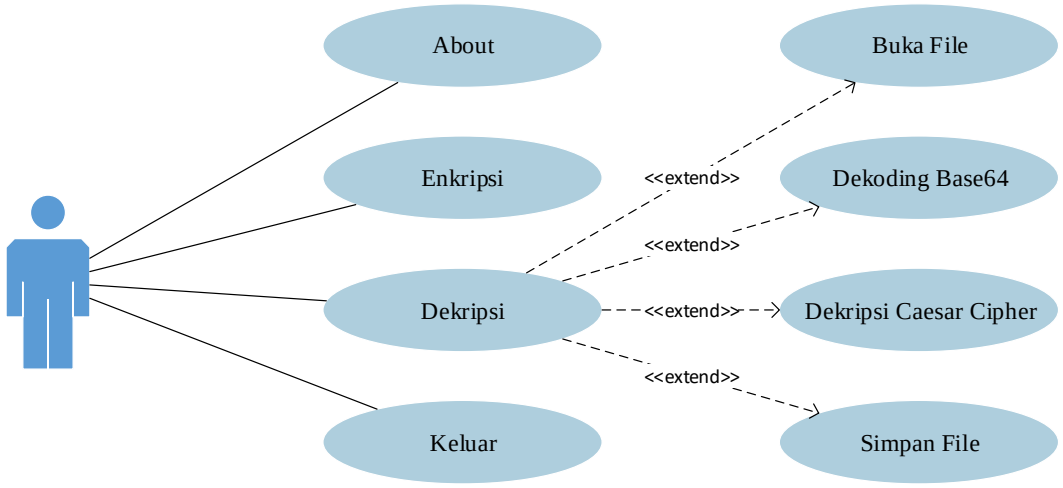


Gambar 3.2 Use Case Diagram Enkripsi

Gambar 3.2 merupakan rancangan *use case* diagram enkripsi. Pada *use case* diagram di atas, tahap pertama yang akan dilakukan oleh pengguna ialah memilih video dengan format MP4. Setelah pengguna berhasil memilih video, sistem akan secara otomatis memproses enkripsi video tersebut dengan menggunakan metode algoritma Caesar Cipher dan *base64*. Setelah proses enkripsi video dengan menggunakan algoritma *base64* selesai, sistem akan mengirimkan hasil enkripsi video sehingga pengguna dapat menyimpan *file* video hasil enkripsi tersebut.

3.4.2 Use Case Diagram Dekripsi

Berikut ini adalah use case diagram yang digunakan dalam melakukan proses dekripsi file video:



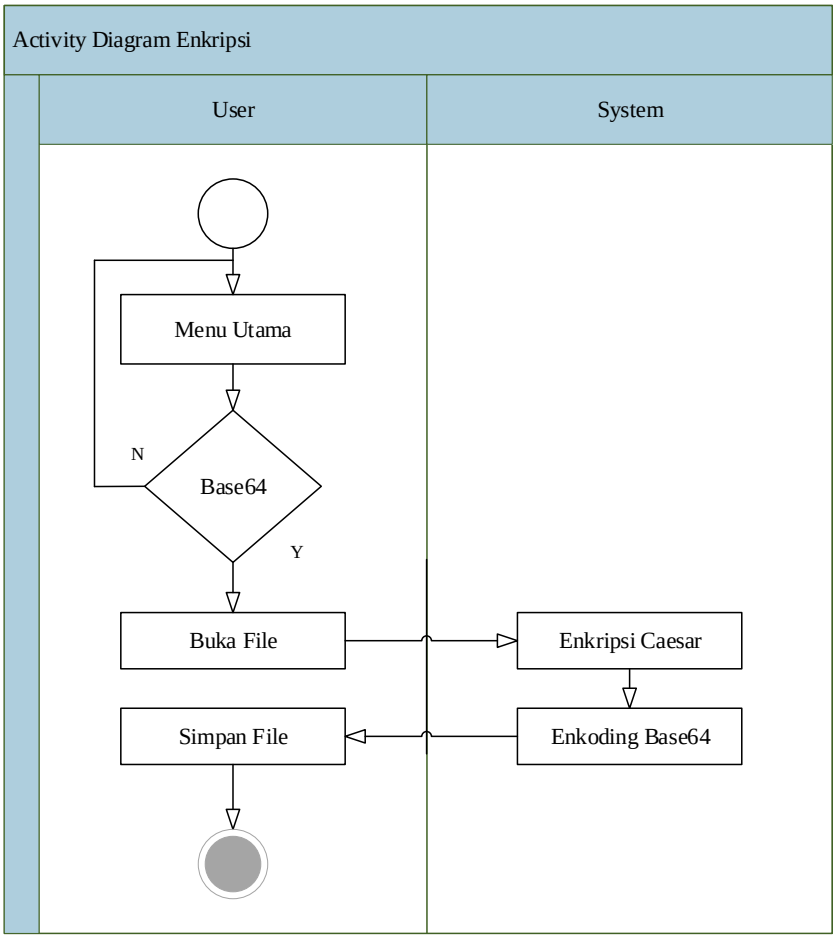
Gambar 3.3 Use Case Diagram Dekripsi

Gambar 3.3 ini merupakan rancangan *use case* diagram dekripsi. Pada *use case* diagram dekripsi tersebut, tahap awal yang harus dilakukan oleh pengguna

ialah memilih *file* video yang akan didekripsi. Pengguna dapat memilih *file* video yang berformat Base64. Sistem akan melakukan dekripsi video tersebut dengan menggunakan algoritma Caesar Cipher dan algoritma base64. Setelah proses dekripsi berhasil, sistem akan menampilkan hasil video tersebut sehingga pengguna dapat menyimpan hasil video tersebut.

3.4.3 Activity Diagram Enkripsi

Berikut ini adalah *activity* diagram proses enkripsi.

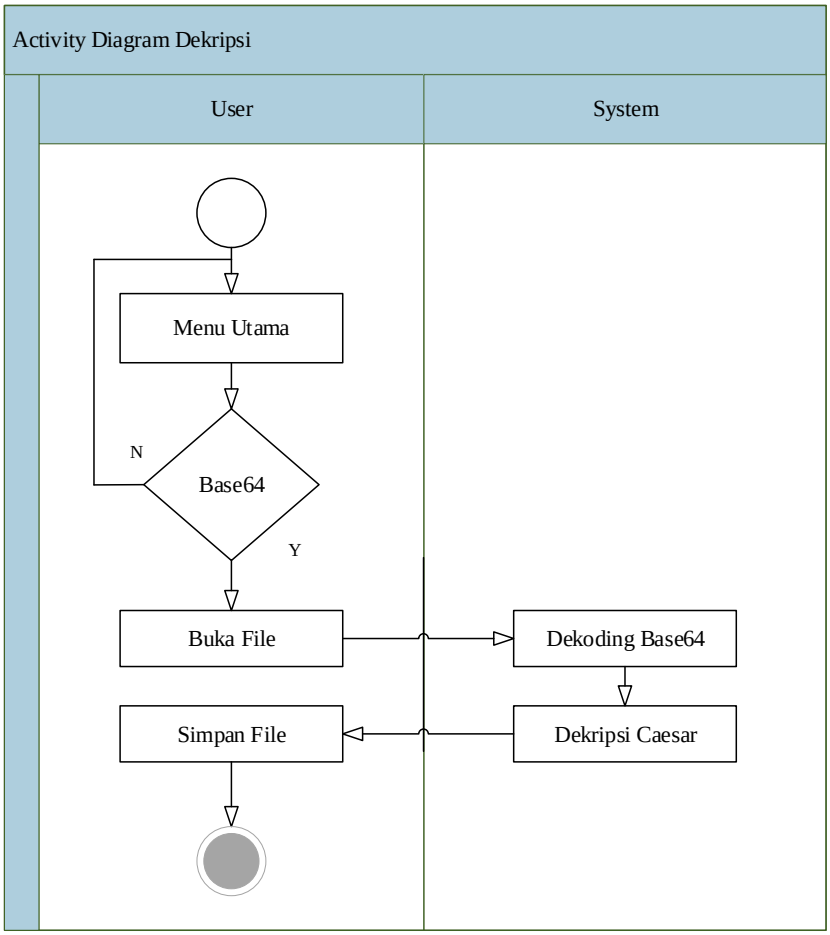


Gambar 3.4 Activity Diagram Enkripsi

Gambar 3.4 merupakan rancangan *activity* diagram enkripsi dari video dengan menggunakan algoritma *base64*. Pada *activity* diagram enkripsi, pengguna akan memilih video yang belum terenkripsi dahulu yang nantinya video tersebut akan diproses dan dienkripsi dengan menggunakan algoritma Caesar Cipher dan *base64*.

3.4.4 Activity Diagram Dekripsi

Berikut ini adalah activity diagram proses dekripsi.

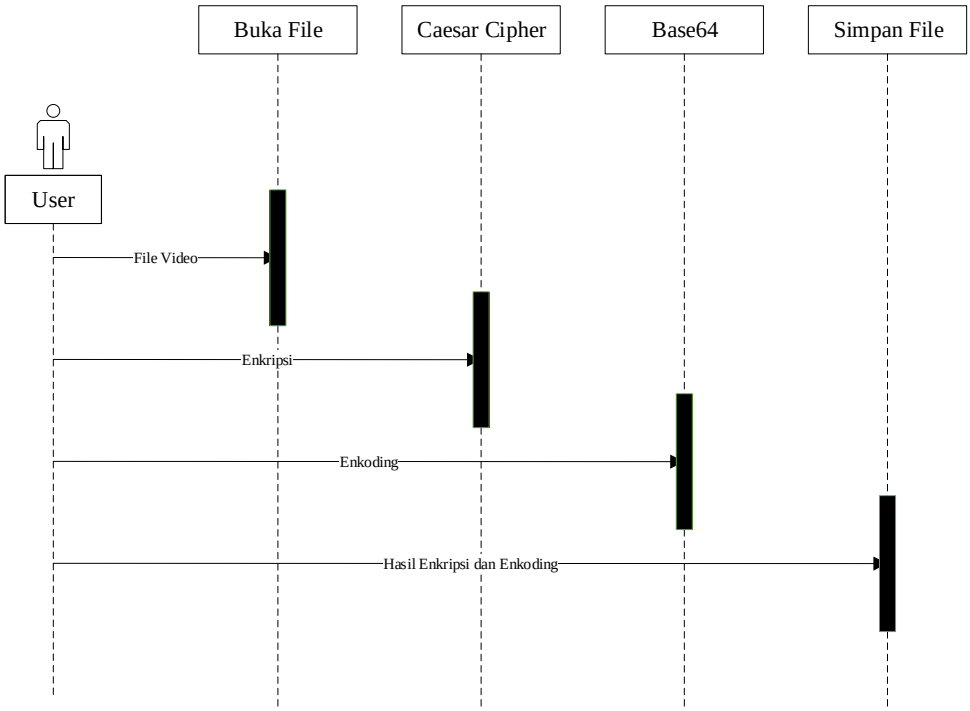


Gambar 3.5 Activity Diagram Dekripsi

Gambar 3.5 ini merupakan rancangan *activity* diagram dekripsi video dengan menggunakan algoritma *base64*. Pada *activity* diagram dekripsi, pengguna dapat memilih video yang berformat Base64 (yang telah dienkripsi sebelumnya) yang nantinya video ini akan diproses dan didekripsi oleh sistem dengan menggunakan algoritma Caesar Cipher dan *base64*. Setelah sistem berhasil mendekripsi video tersebut, sistem akan menampilkan hasil dekripsi sehingga pengguna dapat menyimpan video hasil dekripsi tersebut.

3.4.5 Sequence Diagram Enkripsi

Berikut ini adalah sequence diagram proses enkripsi yang menjelaskan alur dari proses enkripsi menggunakan algoritma Caesar Cipher dan Base64.

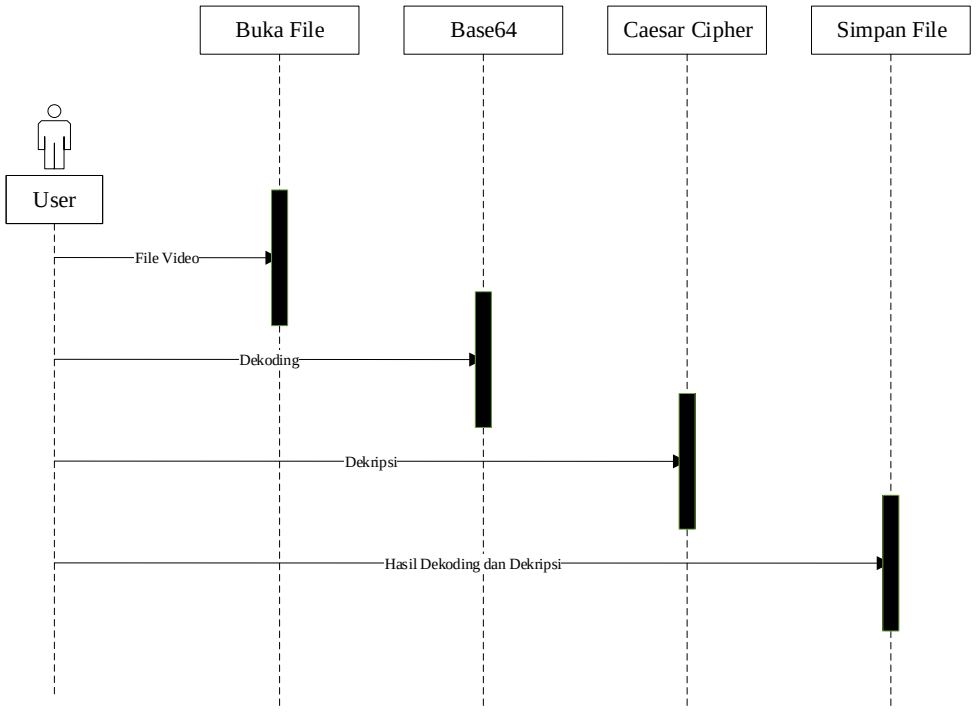


Gambar 3.6 Sequence Diagram Enkripsi

Gambar 3.6 merupakan rancangan *sequence diagram* enkripsi. Pada *sequence diagram* enkripsi, terdapat empat tahapan utama yang akan dilalui oleh pengguna yaitu memilih video (yang belum terenkripsi), proses enkripsi, implementasi algoritma Caesar Cipher dan *base64* dan hasil enkripsi. Pada tahap-tahap ini, pengguna akan menunggu beberapa saat sampai video yang dipilih dapat dilihat hasil enkripsinya dan dapat disimpan.

3.4.6 Sequence Diagram Dekripsi

Berikut ini adalah sequence diagram proses dekripsi yang menjelaskan alur dari proses dekripsi menggunakan algoritma Caesar Cipher dan Base64.



Gambar 3.7 Sequence Diagram Dekripsi

Gambar 3.7 merupakan rancangan *sequence diagram* dari proses dekripsi video dengan menggunakan algoritma *base64*. Pada *sequence diagram* di atas, pengguna akan melalui empat tahapan proses dekripsi diantaranya yaitu pilih video (yang telah dienkripsi), proses dekripsi Caesar Cipher dan dekoding dengan *base64* dan hasil dekripsi. Setelah empat proses utama tersebut dilalui, pengguna dapat melihat dan menyimpan hasil dekripsi dari video yang mereka masukkan.

3.5 Analisis Algoritma

3.5.1 Analisis Algoritma Caesar Cipher

Algoritma Caesar Cipher adalah algoritma yang bekerja dengan cara menggeser posisi karakter dengan suatu angka berdasarkan tabel ASCII. Jumlah pergeseran dapat dilakukan dengan angka berapa saja, tetapi jika hasil perhitungan karakter telah melewati dari 255 maka, karakter tersebut harus berputar dari awal lagi. Untuk memutar karakter tersebut dilakukan proses modulo 256.

Contoh:

PT = A

ASCII = 65

Kunci = 5

CT = 65 + 5

= 70

= F

3.5.2 Analisa Algoritma Base64

Transformasi *Base64* merupakan salah satu algoritma untuk *Encoding* dan *Decoding* suatu data ke dalam format ASCII, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metoda yang digunakan untuk melakukan *encoding* (penyandian) terhadap data *extension*. Karakter yang dihasilkan pada transformasi *Base64* ini terdiri dari A..Z, a..z dan 0..9, serta ditambah dengan dua karakter terakhir yang bersimbol yaitu+ dan / serta satu buah karakter sama dengan(=) yang digunakan untuk penyesuaian dan menggenapkan data *extension* atau istilahnya disebut sebagai pengisi pad. Karakter simbol yang akan dihasilkan akan tergantung dari proses algoritma yang berjalan. Dalam *Encoding Base64* dapat dikelompokkan dan dibedakan menjadi beberapa kriteria yang tertera.

Teknik *encoding Base64* sebenarnya sederhana, jika ada satu (*string*) *bytes* yang akan disandikan ke *Base64* maka caranya adalah sebagai berikut:

Misal kita ingin menyandikan teks MAN

a. Ubah huruf-huruf yang akan dienkripsi menjadi kode-kode ASCII

   Text Content : M – A – N

   ASCII : 77 – 97 - 110

b. Kode-kode ASCII tersebut diubah lagi menjadi kode biner

   Text Content : M – A – N

   ASCII : 77 – 97 – 110

   Bit Pattern : 01001101 – 01100001 - 01101110

c. Bagi kode biner tersebut menjadi hanya 6 angka per blok dan berjumlaah kelipatan 4 blok.

d. Jika angka biner tidak berjumlah 6 angka dan 4 bok maka akan ditambah kode biner 0 sehingga mencukupi menjadi 4 blok.

e. Blok-blok tersebut ubah kembali menjadi kode desimal (data dibaca sebagai index)

   Text Content : M – A – N

   ASCII : 77 – 97 – 110

   Bit Pattern : 010011 – 010110 – 000101 – 101110

   Index : 19 – 22 – 5 – 46

f. Hasil kode index tersebut diubah menjadi huruf yang ada pada index

   Text Content : M – A – N

   ASCII : 77 – 97 – 110

   Bit Pattern : 010011 – 010110 – 000101 – 101110

   Index : 19 – 22 – 5 – 46

   Base64 Encoded : T – W – F – u

g. Jika nilai blok adalah hasil tambahan (0) maka hasil dari index tersebut bernilai ‘=’

   Text Content : M – “(Kosong)” – “(Kosong)”

   ASCII : 77 – “(Kosong)” – “(Kosong)”

   Bit Pattern : 010011 - 010000 – 000000 – 000000

   Index : 19 – 16 – (Kosong) – (Kosong)

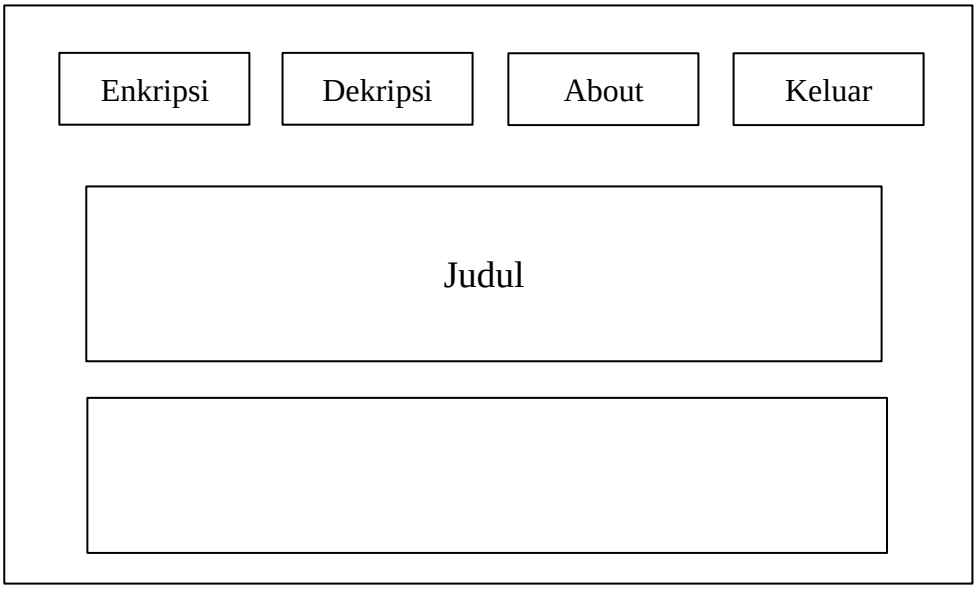
   Base64 Encode : T – Q - = - =

3.6 Perancangan Antarmuka

Perancangan antar muka merupakan gambaran (*mockup*) dari tampilan aplikasi yang akan dibuat.

3.6.1 Menu Utama

Menu utama adalah tampilan yang pertama sekali muncul pada saat program aplikasi dijalankan. Gambar 3.8 adalah hasil perancangan menu utama yang memiliki beberapa komponen lainnya.



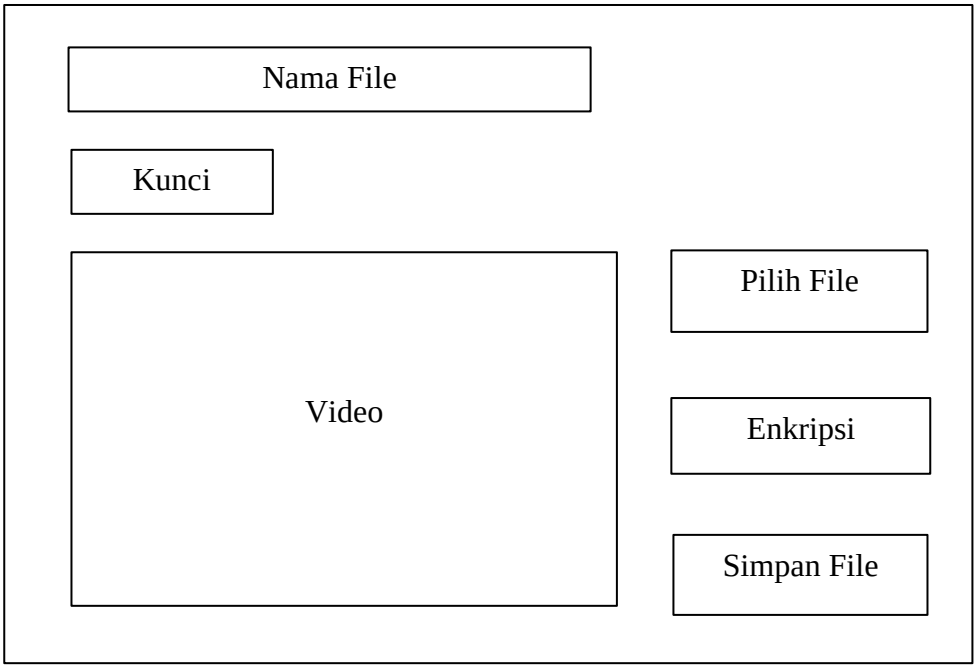
Gambar 3.8 Tampilan Menu Utama

Tampilan ini memiliki berapa sub-menu antara lain:

- Enkripsi
- Dekripsi
- About

3.6.2 Rancangan Tampilan Enkripsi

Gambar 3.9 merupakan rancangan tampilan halaman enkripsi video. Pada tampilan ini, pengguna dapat melihat menu-menu yang disediakan oleh sistem dan pengguna dapat mulai memilih video dengan menekan tombol pilih *file.* Setelah proses pemilihan selesai, pengguna dapat melihat memulai proses enkripsi, pengguna dapat menekan tombol enkripsi.

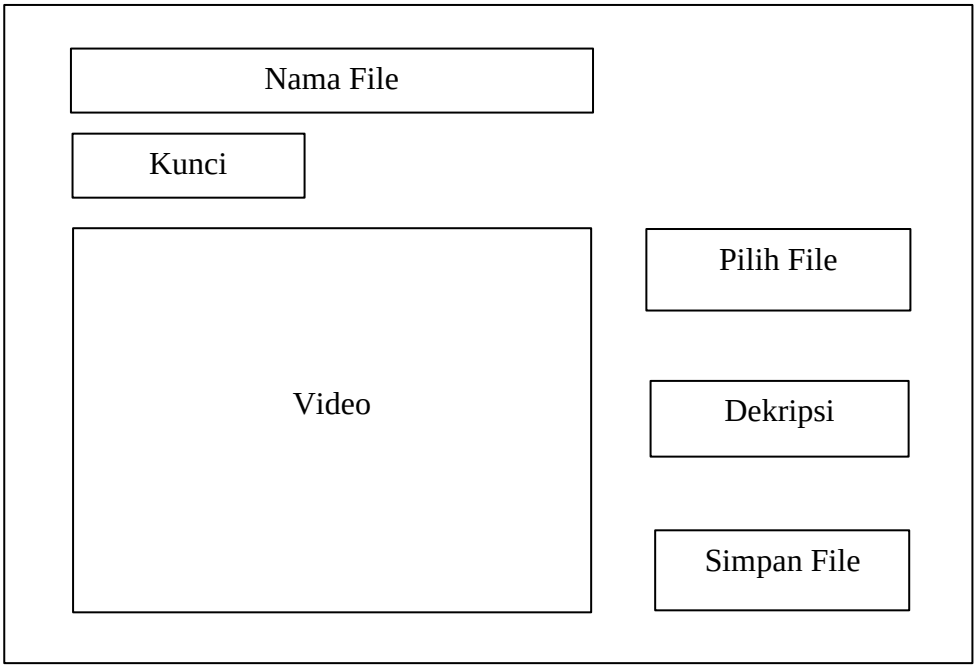


Gambar 3.9 Rancangan Tampilan Enkripsi

3.6.3 Rancangan Tampilan Dekripsi

Gambar 3.10 merupakan rancangan tampilan dekripsi video. Pada rancangan tampilan ini nantinya pengguna dapat memilih *file* video yang terenkripsi dengan menekan tombol pilih *file*. Setelah proses pemilihan video

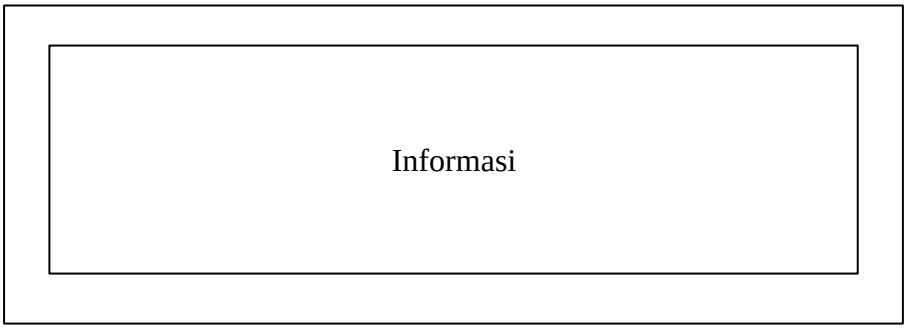
berhasil, pengguna dapat memulai proses dekripsi dengan menekan tombol dekripsi pada halaman yang telah disediakan.



Gambar 3.10 Rancangan Tampilan Dekripsi

3.6.4 Rancangan Tampilan About

Gambar 3.11 merupakan rancangan tampilan tentang aplikasi. Pada rancangan tampilan ini nantinya pengguna dapat melihat penjelasan singkat mengenai aplikasi enkripsi dan dekripsi video dengan menggunakan algoritma *base64*.



Gambar 3.11 Rancangan Tampilan About

BAB IV

HASIL DAN PEMBAHASAN

4.1 Kebutuhan Spesifikasi Minimum *Software* dan *Hardware*

Untuk menjalankan sistem yang telah penulis buat, minimum spesifikasi untuk software dan hardware yang harus digunakan adalah sebagai berikut:

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. Microsoft Windows 7

   b. Google Chrome

   c. Microsoft Visual Studio 2010

   d. Microsoft Word 2019

4.2 Implementasi Sistem

Pada tahap implementasi penulis akan menjelaskan bagaimana sistem dapat digunakan oleh pengguna. Pada penggunaan sistem, pengguna dapat mulai mengenkripsi video dengan cara memilih video yang akan dienkripsi terlebih dahulu.

Format video yang dapat digunakan pengguna ialah format *.mp4. Setelah pengguna menentukan video mana yang akan dienkripsi, tahap selanjutnya ialah pengguna dapat masuk ke dalam sistem lalu memilih *file* video tersebut. Untuk mulai mengenkripsi, pengguna dapat menekan tombol enkripsi. Setelah proses enkripsi berhasil, sistem akan menampilkan hasil enkripsi ke pengguna untuk dapat diunduh oleh pengguna. Pada proses dekripsi, pengguna dapat memilih *file* video yang telah dienkripsi sebelumnya lalu menekan tombol dekripsi untuk memulai proses dekripsi. Setelah proses dekripsi selesai, sistem akan menampilkan hasil yang nantinya *file* video yang telah berhasil didekripsi dapat diunduh oleh pengguna.

4.3 Hasil Tampilan Sistem

Berikut merupakan hasil tampilan dari program aplikasi yang telah dibuat oleh penulis tentang aplikasi enkripsi dan dekripsi *file* video dengan menggunakan algoritma Caesar Cipher dan *base64.*

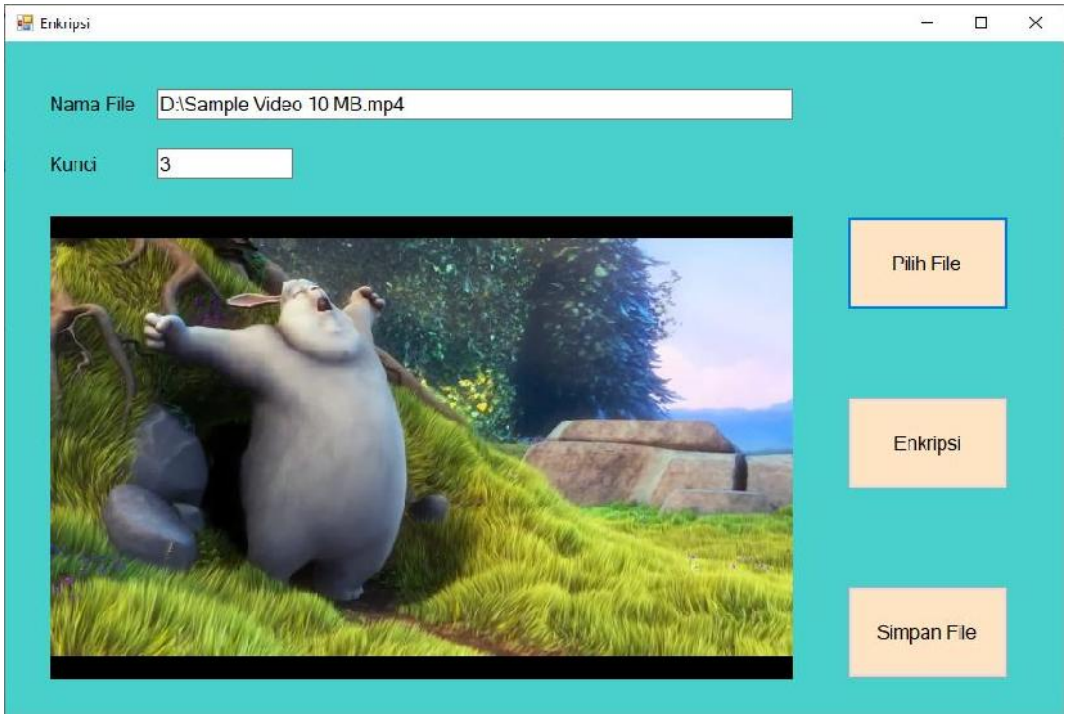
4.3.1 Tampilan Halaman Menu Utama

Halaman Menu Utama merupakan halaman utama sebuah program aplikasi di mana pengguna dapat menemukan beberapa fungsi atau dapat berpindah ke halaman-halaman yang lain pada program aplikasi tersebut. Gambar 4.1 adalah hasil tampilan menu utama.

Gambar 4.1 Halaman Menu Utama

4.3.2 Tampilan Halaman Enkripsi

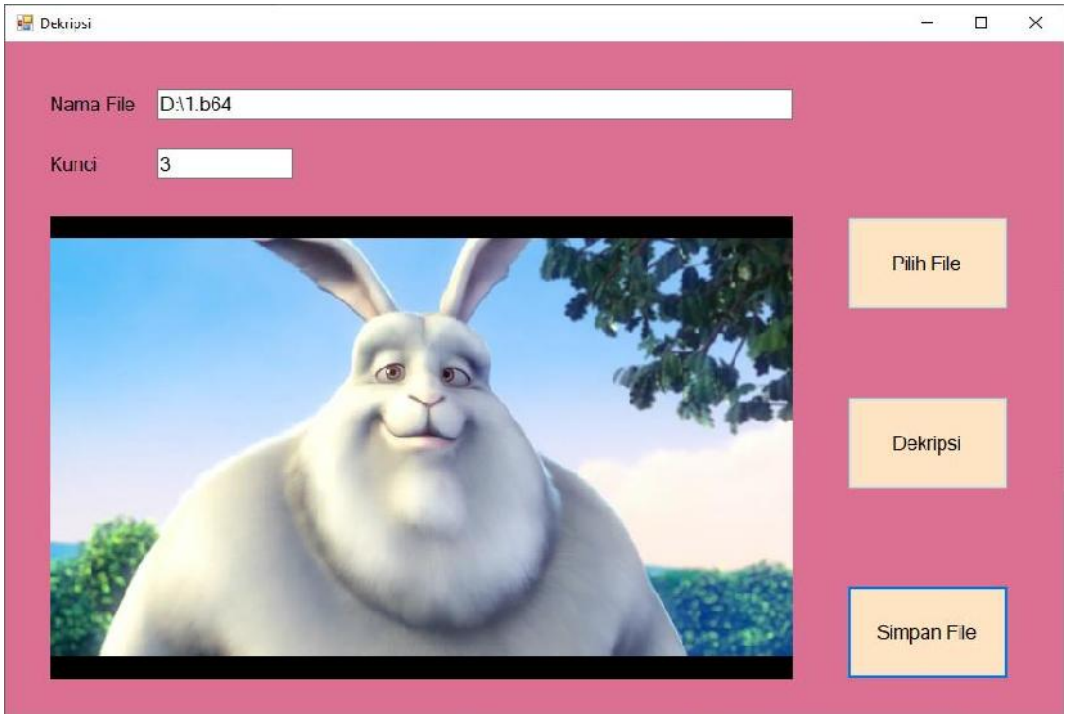
Gambar 4.1 merupakan tampilan dari halaman enkripsi video. Pada tampilan ini pengguna dapat memilih *file* video dengan format *.mp4 dengan cara menekan tombol pilih *file*. Setelah *file* video berhasil dipilih, tahap selanjutnya ialah pengguna dapat menekan tombol enkripsi untuk mulai proses enkripsi. Setelah proses enkripsi berhasil, sistem akan menampilkan hasil enkripsi video yang nantinya dapat diunduh oleh pengguna.



Gambar 4.2 Halaman Enkripsi

4.3.3 Tampilan Halaman Dekripsi

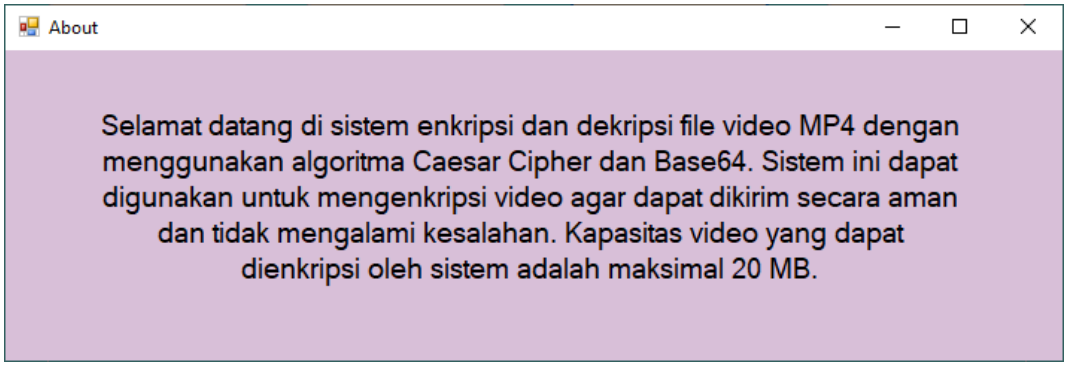
Gambar 4.3 ini merupakan tampilan dari halaman dekripsi. Pada tampilan ini pengguna dapat memilih *file* video yang telah dienkripsi sebelumnya dengan menekan tombol pilih *file*. Setelah *file* video berhasil dipilih, pengguna dapat menekan tombol dekripsi untuk memulai proses dekripsi. Setelah proses dekripsi berhasil, sistem akan menampilkan hasil dekripsi yang nantinya dapat diunduh oleh pengguna.



Gambar 4.3 Halaman Dekripsi

4.3.4 Halaman About

Gambar 4.4 merupakan tampilan dari halaman *about* atau tentang aplikasi. Pada tampilan ini nantinya pengguna dapat melihat penjelasan singkat mengenai sistem enkripsi dan dekripsi video.



Gambar 4.4 Halaman About

4.4 Pengujian Sistem

Berikut ini adalah hasil penelitian yang diperoleh. Tabel 4.1 menjelaskan bagaimana sistem yang diharapkan menghasilkan output yang sesuai.

Tabel 4.1 Pengujian Sistem

No.	Bulir Pengujian	Output yang diharapkan	Output yang keluar	Keterangan
1	Enkripsi video	Sistem dapat melakukan enkripsi dengan menggunakan metode *base64*	Sistem berhasil melakukan enkripsi dengan menggunakan metode *base64*	Sesuai
2	Dekripsi video	Sistem dapat melakukan dekripsi dengan menggunakan metode *base64*	Sistem berhasil melakukan dekripsi dengan menggunakan metode *base64*	Sesuai
3	Implementasi *Base64*	Sistem dapat mengimplementasikan metode *base64* secara baik dan tepat	Sistem berhasil mengimplementasikan metode *base64* secara baik dan tepat	Sesuai

4.5 Kelebihan dan Kekurangan Sistem

Berikut merupakan kelebihan dan kelemahan dari sistem yang telah berhasil penulis buat:

1. Kelebihan Sistem
    a. Sistem dapat melakukan enkripsi video dengan menggunakan metode *base64* secara cepat dan tepat sehingga proses enkripsi berjalan secara baik.

b. Sistem dapat melakukan dekripsi video dengan menggunakan metode *base64* secara cepat dan tepat sehingga proses dekripsi berjalan secara baik.

c. Sistem mengenkripsi *biner* (*bit*) pada *file* video sehingga *file* video tidak dapat dibuka ataupun dimanipulasi.

2. Kelemahan Sistem

a. Sistem tidak mengenkripsi sistem dengan menggunakan kunci (key) sehingga proses enkripsi hanya berjalan pada sisi *encoding* saja.

b. Sistem hanya dapat diakses secara *offline* karena belum menggunakan *server* berbasis *online*.

BAB V

PENUTUP

5.1 Kesimpulan

Berikut merupakan kesimpulan yang penulis buat berdasarkan pembahasan pada implementasi dan penggunaan algoritma *base64* dalam pengamanan *file* video:

1 Sistem enkripsi ini menggunakan metode enkripsi dan dekripsi *base64* dalam memproses *file* video yang dipilih oleh pengguna dengan batas maksimal ukuran video yaitu 20 MB.

2 Penggunaan metode *base64* dalam proses enkripsi dan dekripsi ini dinilai efektif karena *base64* memiliki proses yang mudah dan cepat namun aman sehingga metode ini memiliki tingkat kelayakan tinggi untuk dijadikan acuan sebagai mengamankan *file* video.

3 Pembuatan sistem enkripsi dan dekripsi video ini dimaksudkan untuk mengamankan *file* video yang akan dikirim dan diterima oleh pengguna sehingga kerahasiaan data yang ada di dalamnya menjadi lebih aman dan terjamin.

5.2 Saran

Berikut merupakan saran yang penulis dapatkan berdasarkan pembahasan dalam implementasi dan penggunaan algoritma *base64* dalam pengamanan *file* video:

1. Sistem ini masih menggunakan desktop yang artinya sistem hanya dapat diakses pada perangkat lokal.
2. Dalam proses enkripsi dan dekripsi, sistem hanya dapat mengenkripsi dan mendekripsi satu video saja sehingga proses pengamanan *file* menjadi lebih lambat dan harus dilakukan satu per satu.
3. Kedepannya penulis berharap sistem dapat dikembangkan sehingga dapat dikombinasikan dengan algoritma enkripsi dan dekripsi lain seperti *RC4* dan *Blowfish*.

DAFTAR PUSTAKA

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

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.

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

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

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

Hafni, Layla, and Rismawati Rismawati. "Analisis faktor-faktor internal yang mempengaruhi nilai perusahaan pada perusahaan manufaktur yang terdaftar di bei 2011-2015." Bilancia: Jurnal Ilmiah Akuntansi 1.3 (2017): 371-382.

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.

Indra permana, a. m. i. n. u. d. d. i. n. "Sistem pakar mendeteksi hama dan penyakit tanaman kelapa sawit pada pt. moeis kebun sipare-pare kabupaten batubara." (2013).

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.

Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." Int. J. Recent Trends Eng. Res 2.12 (2016): 140-151.

Nugroho, A. Y. (2015). Pembuatan Aplikasi Kriptografi Algoritma Base64 Menggunakan PHP Untuk Mengamankan Data Text. *Seminar Nasional Informatika*, *1*(1).

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

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

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

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

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

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

Shi, J., Renwick, R., Turner, N. E., & Kirsh, B. (2019). Understanding the lives of problem gamers: The meaning, purpose, and influences of video gaming. *Computers in Human Behavior*, *97*(10), 291–303. https://doi.org/10.1016/j.chb.2019.03.023

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

Syahputra, Rizki, and Hafni Hafni. "Analisis kinerja jaringan switching clos tanpa buffer." journal of science and social research 1.2 (2018): 109-115.

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.

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

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

Zen, Muhammad. "perbandingan metode dimensi fraktal dan jaringan syaraf tiruan backpropagation dalam sistem identifikasi sidik jari pada citra digital." jitekh 7.2 (2019): 42-50.