**PENERAPAN SISTEM KRIPTOGRAFI HYBRID MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION SYSTEM DAN RIVEST SHAMIR ADLEMAN PADA APLIKASI PYTHON  
BERBASIS DESKTOP**

**SKRIPSI**

Diajukan untuk memenuhi salah satu persyaratan

kelulusan program Sarjana Strata Satu (S1)

Disusun oleh

**Afif Farakhan**

**NRP. 161014039**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA**

**D A N I L M U K O M P U T E R L P K I A**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**BANDUNG**

**2020**

# BAB I PENDAHULUAN

## Latar Belakang Masalah

Di zaman modern ini keamanan merupakan aspek yang sangat penting untuk kita perhatikan ketika kita menggunakan teknologi - teknologi modern. Salah satu bidang keilmuan yang dapat menjaga keamanan kita di zaman modern ini khususnya dalam berkomunikasi adalah kriptografi. Menurut Niels, Bruce dan Tadayoshi (2010), dalam buku nya yang berjudul Cryptography Engineering menyebutkan bahwa "Cryptography is the art and science of encryption." Yang mana artinya kriptografi merupakan seni sekaligus keilmuan mengamankan pesan.

Kriptografi merupakan ilmu yang memanfaatkan suatu rumus atau algoritma matematika yang diterapkan pada suatu teks (*plaintext*) untuk diacak menjadi tulisan yang tidak dapat dimengerti (*ciphertext*) agar teks atau pesan tersebut tidak dapat dimengerti oleh orang - orang yang tidak berwenang untuk membaca nya dan hanya orang - orang tertentu saja yang memiliki kunci yang dapat memahami isi teks asli tersebut.

Kriptografi sudah digunakan sejak zaman dahulu yang juga sering disebut kriptografi klasik. Salah satu kriptografi klasik yang terkenal di zaman dahulu adalah kriptografi yang digunakan oleh Julius Caesar bernama Caesar Cipher yang mana digunakan untuk mengamankan dokumen - dokumen penting agar informasi tersebut terjaga kerahasiaannya. Namun di zaman modern sekarang ini kriptografi digunakan dalam berbagai aspek teknologi seperti dalam sistem keamanan jaringan, keamanan data, juga termasuk dalam perpesanan modern.

Kriptografi di zaman modern ini sudah memiliki berbagai jenis dan berbagai macam algoritma yang mana masing – masing jenis atau macam algoritma tersebut memiliki karakteristik masing – masing. Diantaranya ada 2 jenis kriptografi modern yaitu jenis simetris dan asimetris. Untuk kriptografi simetris diantaranya ada algoritma kriptografi DES (Data Encryption Standard), Blowfish dan AES (Advanced Encryption System). Sedangkan untuk kriptografi asimetris diantaranya ada algoritma kriptografi ECC (Elliptic Curve Cryptography) dan RSA (Rivest Shamir Adleman).

Kriptografi Simetris merupakan kegiatan mengacak suatu pesan yang hanya menggunakan 1 buah kunci baik itu untuk menenkrip pesan maupun mendekrip pesan. Sedangkan Kriptografi Asimetris merupakan kegiatan mengacak suatu pesan yang mana menggunakan 2 buah kunci dimana 1 kunci untuk menenkrip pesan dan 1 kunci lagi untuk mendekrip pesan. Dari kedua jenis kategori kriptografi modern ini masing - masing memiliki karakteristik, rumus, algoritma, kelebihan maupun kelemahan nya tersendiri.

Pada Kriptografi Simetris jumlah kunci yang digunakan biasanya sebanyak 128-bit - 256-bit yang mana terbilang sedikit dan proses atau kegiatan menenkrip maupun mendekrip akan sangat cepat. Kegiatan menenkrip maupun mendekrip suatu pesan hanya menggunakan 1 kunci yang sama yang mana artinya siapapun yang memiliki atau mengetahui kunci tersebut dapat mendekrip pesan sehingga isi pesan dapat diketahui. Disini lah yang dapat menjadi titik kelemahan kriptografi simetris dimana dibutuhkan nya saluran yang aman untuk pertukaran kunci antara pengirim pesan dan penerima pesan. Bila saluran tadi disadap oleh pihak yang tidak berwenang maka kunci pun bisa dicuri dan pesan yang sudah dienkrip tadi dapat didekrip pula oleh pihak yang tidak berwenang tersebut.

Pada kriptografi asimetris kunci yang digunakan untuk menenkrip dan mendekrip pesan merupakan kunci yang berbeda sehingga terdapat 2 buah kunci yaitu kunci publik dan kunci privat. Kriptografi asimetris tidak membutuhkan saluran yang aman untuk pertukaran kunci namun letak kelemahan nya ada pada jumlah kunci yang digunakan yaitu sebanyak 1024-bit - 2048-bit yang mana dalam proses enkripsi maupun dekripsi akan memakan waktu yang lama apalagi bila data yang akan dienkrip maupun didekrip memiliki ukuran yang sangat besar maka waktu yang dibutuhkan akan lebih lama lagi.

Namun sistem kriptografi modern yang baru yaitu sistem kkriptografi hybrid dapat mengatasi masalah kelemahan 2 sistem kriptografi tersebut dengan menggabungkan kelebihan masing - masing untuk mengatasi kelemahan - kelemahan yang ada.

Maka dari itu permasalahan-permasalahan yang ada di atas menjadi gagasan untuk menuangkannya ke dalam tugas akhir dengan mengambil sebuah judul "PENERAPAN SISTEM KRIPTOGRAFI HYBRID MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION SYSTEM DAN RIVEST SHAMIR ADLEMAN PADA APLIKASI PYTHON BERBASIS DESKTOP”.

## Identifikasi Permasalahan

Berdasarkan latar belakang masalah yang telah tertulis diatas, maka berikut adalah identifikasi masalah yang akan dijadikan bahan penelitian yaitu:

1. Bagaimana mengatasi kelemahan sistem kriptografi simetris dan sistem kriptografi asimetris dengan menggunakan sistem kriptografi hybrid.
2. Bagaimana proses dan penerapan sistem kriptografi hybrid dengan menggunakan algoritma kriptografi Advanced Encryption System (AES) dan algoritma kriptografi Rivest Shamir Adleman (RSA) pada aplikasi Python berbasis desktop.

## Ruang Lingkup Permasalahan

Adapun ruang lingkup permasalahan dari penelitian ini yaitu:

1. Yang akan dienkripsi dan didekripsi hanyalah pesan teks.
2. Karakter yang digunakan untuk proses enkripsi dan dekripsi hanya karakter ASCII dan UTF-8.

## Tujuan Perancangan

Tujuan dari penelitian ini yaitu menerapkan sistem kriptografi hybrid pada aplikasi Python berbasis desktop dengan menggunakan algoritma AES dan RSA untuk mengatasi kelemahan sistem kriptografi simetris dan asimetris.

## Metodologi Penelitian

1. Metodologi Penelitian

Dalam menyelesaikan masalah, penulis melakukan metode penelitian sebagai berikut :

1. Studi Literatur

Studi literatur adalah pencarian terhadap sumber tertulis yang sudah tersedia dan terverifikasi baik dari buku, arsip, artikel maupun jurnal yang relevan dengan permasalahan yang dibahas. Sehingga informasi yang didapat valid dan hasil dari skripsi ini dapat memperkual argumen – argument yang sudah ada.

1. Konsultasi

Konsultasi dilakukan untuk mengumpulkan saran-saran yang didapat dari beberapa pihak, seperti Dosen, Pebimbing dan Rekanan yang memahami tentang permasalahan yang diangkat.

1. Metode Pengumpulan Data
2. Observasi

Observasi merupakan salah satu metode pengumpulan data yang dilakukan dengan cara pengamatan sekaligus pencatatan secara urut yang terdiri dari unsur-unsur yang bermunculan dalam suatu fenomena-fenomena dalam objek penelitian. Hasil dari pengamatan akan dilaporkan dengan susunan yang sistematis dan sesuai kaidah yang berlaku.

1. Metode Pengolahan Data

Apriori merupakan sebuah algoritma yang mengolah suatu database transaksi dengan setiap transaksi adalah suatu himpunan item-item. Kemudian mencari seluruh kaidah apriori yang memenuhi kendala minimum support dan minimum confidence yang diberikan user.

1. Metode Pengembangan Sistem

Metode untuk penelitian ini adalah Prototype. Prototype model ini merupakan metode siklus hidup pengembangan perangkat lunak yang bertujuan mendapatkan kebutuhan yang jelas. Metode ini memungkinkan untuk pengembangan piranti lunak yang jauh lebih cepat dibanding metode waterfall. Metode prototype digunakan karena cocok untuk sistem yang bersifat customize (Susanto, 2016).

1. Metode Pengembangan Perangkat Lunak

Perancangan dan pembangunan aplikasi dilakukan dengan menggunakan bahasa pemrograman Python dan database mySQL untuk sistem berbasis web dengan menerima masukan berupa file excel berformat .xls untuk data training dan data testing. Metode yang digunakan merupakan Object Oriented Programming (OOP).

1. Metode Pengujian Perangkat Lunak

Pengujian akan dilakukan dengan metode Black-box testing, karena black-box testing yang dibutuhkan untuk pengujian adalah fungsionalitas program tanpa alur kineja program.

## Sistematika Penulisan

Dalam penulisan skripsi ini dibagi dalam 5 bab, yaitu:

**BAB I : PENDAHULUAN**

Bab ini digunakan untuk mendefinisikan persoalan, ruang lingkup dan perencanaan kegiatan dilakukan. Bab ini berisi latar belakang, identifikasi permasalahan, ruang lingkup dan batasan permasalahan, tujuan perancangan, metodologi penelitian dan sistematika penulisan.

**BAB II : DASAR TEORI**

Bab ini berisi teori-teori pendukung tentang teori permasalahan, analisa penggalian data, pengembangan sistem, pengembangan perangkat lunak, yang meliputi: konsep dasar sistem, konsep dasar data mining, konsep dasar algoritma naive bayes, pengertian Prototype, pengertian OOP, pengertian web, pengertian UML, PHP, dan Framework CI, serta teori-teori lainnya yang digunakan untuk mendukung penganalisaan dan pengembangan sistem baru yang diusulkan.

**BAB III : ANALISIS DAN PERANCANGAN PERANGKAT LUNAK**

Bab ini berisi analisa penggalian data dalam memprediksi kelulusan mahasiswa berdasarkan IPK serta rancang bangun perangkat lunak yang akan di implementasikan.

**BAB IV : ANALISIS DAN PERANCANGAN PERANGKAT LUNAK**

Bab ini berisi uraian hasil rancangan sistem secara nyata, menggambarkan penggunaan perangkat lunak serta implementasi.

**BAB IV : KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dan saran untuk kajian yang dapat dituliskan.

# BAB II DASAR TEORI

## Teori Tentang Permasalahan

Pada bab ini akan dibahas menganai kriptografi dimulai dari definisi kriptografi, sejarah kriptografi, sistem kriptografi klasik, sistem kriptografi modern, sistem kriptografi kunci simetris juga sistem kriptografi Advanced Encryption System (AES), sistem kriptografi kunci asimetris terutama sistem kriptografi Rivest Shamir Adleman (RSA), sistem kriptografi hybrid, juga teori-teori bilangan untuk kriptografi kunci publik dan kunci privat.

### Kriptografi

Kata kriptografi berasal dari bahasa Yunani yang terdiri dari 2 kata yaitu *“kryptos”* (tersembunyi atau rahasia) dan *“graphia”* (tulisan) yang artinya “tulisan rahasia”. Menurut Schneiner “Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (message).” [Schneier,1996].

Kriptografi merupakan seni atau ilmu untuk menjaga keamanan, kerahasiaan atau keautentikasian suatu pesan, dimana pesan ini nanti nya hanya akan dibaca oleh orang – orang yang berhak untuk membaca nya saja dan aman dari pihak – pihak yang tidak berwenang untuk membaca nya.

### Jenis Algoritma Kriptografi Modern

Algoritma kriptografi modern memiliki beberapa jenis diantaranya ada simetris dan asimetris. Namun, seiring perkembangan zaman terdapat jenis baru yaitu algoritma kriptografi hybrid.

#### Algoritma Kriptografi Simetris

Algoritma yang menggunakan sebuah kunci yang sama baik untuk proses enkripsi maupun dekripsi. Kunci algoritma ini menggunakan jumlah kunci yang sedikit sehingga proses enkripsi maupun dekripsi hanya memakan waktu yang sedikit. Kunci dari algoritma ini bersifat rahasia dan dalam pertukaran kunci antara pengirim dan penerima pesan dibutuhkan saluran yang aman dari penyadapan. Contoh algoritma simetris yaitu Beaufort, Spritz, Blowfish, Twofish, DES(Data Encryption Standard).

#### Algoritma Kriptografi Asimetris

Algoritma yang menggunakan 2 buah kunci yang mana satu untuk mengenkripsi dan satu lagi untuk mendekripsi. Kunci untuk mengenkripsi disebut kunci publik yang dapat diketahui oleh siapapun karena bersifat tidak rahasia. Sedangkan kunci untuk mendekripsi disebut kunci privat yang mana harus dijaga kerahasiaan nya. Kunci ini menggunakan jumlah kunci yang lebih banyak dari pada algoritma simetris sehingga kurang cocok untuk menkeripsi data yang berjumlah besar karena proses nya akan memakan waktu yang lama. Contoh algoritma asimetris yaitu RSA (Riverst Shamir Adleman), ECC (Elliptic Curve Cryptography),Elgamal.

#### Algoritma Kriptografi Hybrid

Algoritma ini disebut hybrid dikarenakan algoritma ini memanfaatkan kedua jenis algoritma sebelumnya yaitu simetris dan asimetris. Dengan memanfaatkan kedua buah algoritma modern sebelumnya tujuan dari kritografi ini adalah mengatasi kelemahan dari masing – masing algoritma dengan memanfaatkan kelebihan dari masing – masing algoritma itu sendiri.

### Algoritma Kriptografi RSA

RSA merupakan algoritma kriptografi asimetris karena menggunakan dua kunci, yaitu kunci publik dan kunci pribadi. Ada tiga algoritma dalam kriptografi RSA, yaitu pembangkitan kunci, proses enkripsi, dan proses dekripsi. Algoritma ini memiliki tingkat keamanan yang terletak pada sulitnya memfaktorkan sebuah bilangan besar menjadi dua buah bilangan prima. Kelemahan dari algoritma kriptografi RSA adalah waktu yang dibutuhkan untuk melakukan proses pembangkitan kunci, enkripsi dan dekripsi lambat. Sedangkan kelebihannya terletak pada sulitnya memecahkan kunci dan penggunaan kunci yang lebih efektif.

#### Prosedur Pada Sistem Kriptografi RSA

Sistem kriptografi RSA terdiri dari 3 prosedur: pembangkit kunci publik dan kunci privat, enkripsi dan dekripsi. Berikut ini adalah algoritma untuk prosedur pembangkit Kunci RSA:

Algoritma pembangkit kunci pada sistem kriptografi RSA membuat 2 buah kunci yaitu kunci publik dan kunci privat. Algoritma pembangkit kunci RSA:

1. Pilih dua bilangan prima *p* dan *q*. kedua bilangan *p* dan *q* ini tidak boleh sama. Hitung *N* = *p* x *q*. *N* hasil perkalian dari *p* dikalikan dengan *q.*
2. Hitung φ = (*p*-1)(*q*-1).
3. Pilih bilangan bulat (*integer*) antara satu dan φ (1 < *e* < φ) yang juga merupakan koprima dari φ.
4. Hitung *d* hingga *d e* ≡ 1 (mod φ).

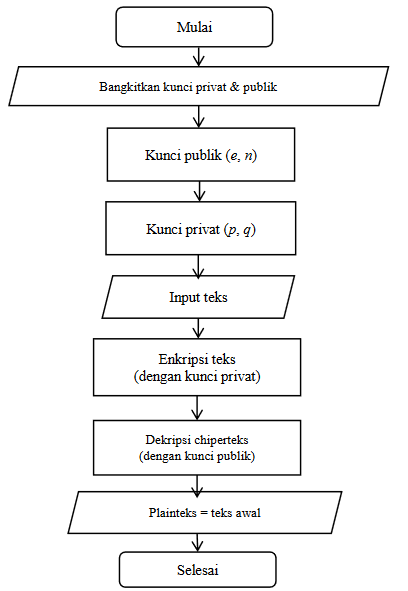
Pada kunci publik terdiri dari:

* *N*, modulus yang digunakan
* *e*, eksponen publik (untuk menenkripsi) tidak rahasia dan boleh diketahui

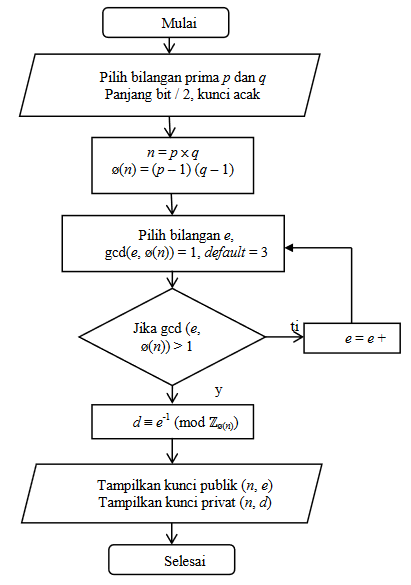
Pada kunci privat terdiri dari:

* *N*, modulus yang digunakan dan digunakan juga pada kunci publik
* *d*, eksponen privat (untuk mendekripsi) dan harus dijaga kerahasiaannya

Berikut adalah Flowchart untuk Algoritma Sistem Kriptografi RSA:



Berikut adalah Flowchart pembangkitan kunci Sistem Kritografi RSA:



#### Proses Enkripsi Pada Sistem Kriptografi RSA

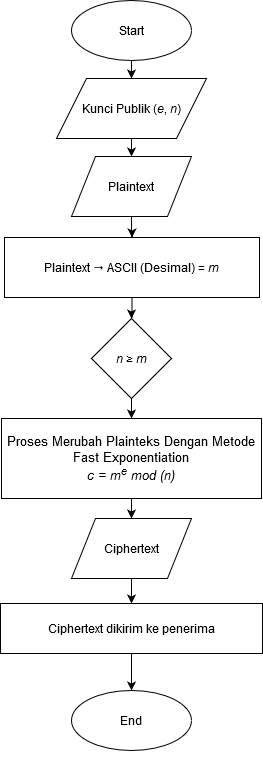
Misalkan Bob ingin mengirim pesan *m* ke Alice. Bob mengubah *m* menjadi angka *n* < *N*, menggunakan protokol yang sebelumnya telah disepakati dan dikenal sebagai *padding scheme.*

Maka Bob memiliki *n* dan mengetahui *N* dan *e*, yang telah diumumkan oleh Alice. Bob kemudian menghitung *ciphertext* *c* yang terkait pada *n:*

*c* = *ne* mod *N*

Perhitungan tersebut dapat diselesaikan dengan cepat menggunakan metode *exponentiation by squaring*. Bob kemudian mengirimkan *c* kepada Alice.

Berikut adalah Flowchart proses enkripsi pada algoritma Sistem Kriptografi RSA:



#### Proses Dekripsi Pada Sistem Kriptografi RSA

Alice menerima *c* dari Bob, dan mengetahui *private key* yang digunakan oleh Alice sendiri. Alice kemudian memulihkan *n* dari *c* dengan langkah-langkah berikut:

*n* = *cd* mod *N*

Perhitungan di atas akan menghasilkan *n*, dengan begitu Alice dapat mengembalikan pesan semula *m*. Prosedur dekripsi bekerja karena

*cd* ≡ (*ne*)*d* ≡ *ned* (mod *N*)

Kemudian, dikarenakan *ed* ≡ 1 (mod p-1) dan *ed* ≡ 1 (mod q-1), hasil dari *Fermat’s little theorem.*

*ned* ≡ *n* (mod *p*)

Dan

*ned* ≡ *n* (mod *q*)

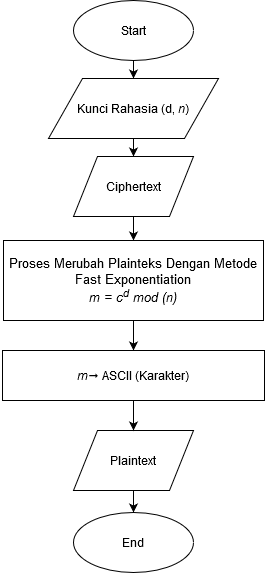
Dikarenakan *p* dan *q* merupakan bilangan prima yang berbeda, mengaplikasikan Chinese Remainder Theorem akan menghasilkan dua macam kongruen

*ned* ≡ *n* (mod *pq*)

Serta

*cd* ≡ *n* (mod *N*)

Berikut adalah Flowchart proses dekripsi pada algoritma Sistem Kriptografi RSA:



### Algoritma Kriptografi AES

AES adalah algoritma kriptografi berbasis chipertext simetrik yang dapat mengenkripsi dan dekripsi video surveillance. AES memiliki sifat cipher yang diharapkan yaitu: tahan menghadapi analisis sandi yang diketahui serta fleksibel digunakan dalam berbagai perangkat keras dan perangkat lunak. Algoritma ini menggunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi.

#### Prosedur Pada Sistem Kriptografi AES

Pada Proses enkripsi awalnya teks asli dibentuk sebagai sebuah state. Kemudian sebelum ronde 1 dimulai blok teks asli dicampur dengan kunci ronde ke-0 (transformasi ini disebut AddRoundKey). Setelah itu, ronde ke-1 sampai dengan ronde ke-(Nr-1) dengan Nr adalah jumlah ronde. AES menggunakan 4 jenis transformasi yaitu:

1. *SubBytes*, sebagai transformasi subtitusi
2. *ShiftRows,* sebagai transformasi permutasi
3. *MixColumns,* sebagai transformasi pengacakan
4. *AddRoundKey,* sebagai transformasi penambahan kunci

Pada ronde terakhir, yaitu ronde ke-Nr dilakukan transformsi serupa dengan ronde lain namun tanpa transfomasi serupa dengan ronde lain namun tanpa transformasi MixColumns.

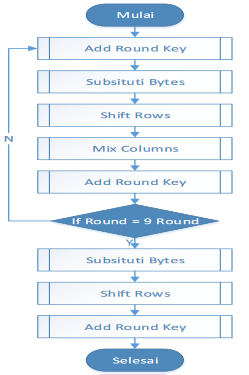
Penyandian AES membutuhkan kunci ronde untuk setiap ronde transformasi kunci ronde ini di bangkitkan (di ekspansi) dari kunci AES. Pada bagian ini di bahas bagaimana kunci ronde di bangkitkan oleh kunci AES. Kunci AES 128 bit atau 4 word menghasilkan sebuah larik sebanyak 44 word yang menjadi kunci. Berikut adalah langkah langkah mengekspansi kunci:

1. Pertama kunci AES 128 bit di organisir menjadi 4 word dan disalin ke word keluaran (W) pada 4 elemen pertama (W[0], W[1], W[2], W[3]).
2. Untuk elemen keluaran selanjutnya W[i]dengan i ={4,...,43} dihitung sebagai berikut:
3. Salin W [i-1] pada word t.
4. Jika i mod 4 = 0 ( I habis dibagi 4 ) maka lakukan W[i]= f (t,i) ⊕W[i-4] ,dengan fungsi f(t,i) adalah sebagai berikut:

f(t,i) = Subword (rotword(t)) ⊕RC[i/4]

1. Jika I mod 4 tidak sama dengan 0, lakukan W[i]= t⊕W[i-4].

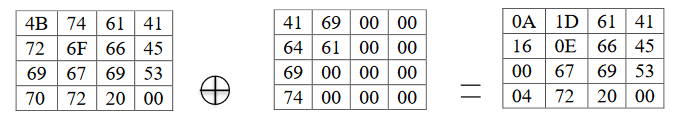
Berikut adalah Flowchart pada algoritma Sistem Kriptografi AES:



###

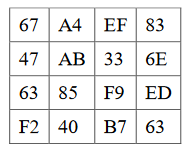
1. **Proses Enkripsi pada Sistem Kriptografi AES**

Hal pertama yang harus dilakukan adalah Melakukan xor antara plainteks dan kunci, seperti berikut:

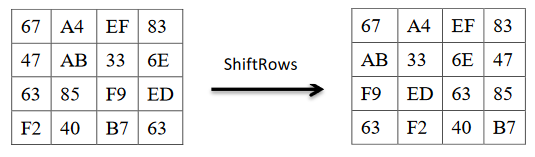


**Ronde 1**

1. Hasil Proses Subbyte ( menggunakan table s-box)



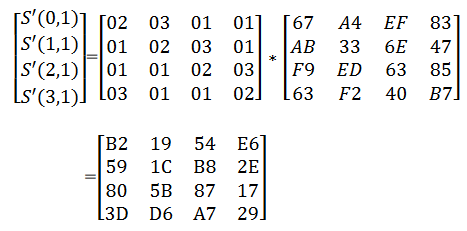
1. Transformasi ShiftRows



1. Proses MixColumns

Proses ini merupakan proses terbanyak dari pada proses proses lain setiap rounde. Kali ini penulis membagi proses mixcolumnsmenjadi 4 bagian untuk sebuah matriks atau state karena dikerjakan untuk setiap kolom sebagai berikut:

1. Proses mixcolumn untuk kolom pertama.



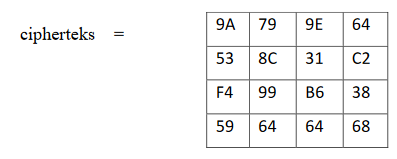
Pada ronde pertama didapat Cipherteks yang akan menjadi masukan atau input untuk ronde 2, begitu juga cipherteks yang didapat pada ronde 2 kan digunakan menjadi input pada ronde 3. Proses seperti ini berlangsung hingga ronde 10. Pada ronde 10 didapat hasil enkripsi sebagai berikut:

**Ronde ke-10**:





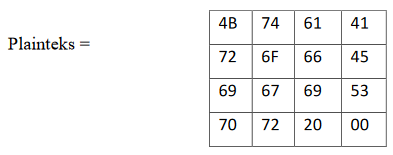
Pada ronde 10 transformasi yang dilakukan hanya 3 transformasi yaituSubbyte, ShiftRows,Addroundkey. Dan didapat cipherteks yang sesungguhnya yaitu:



jika didalam bentuk ASCII maka di dapat cipherteks: š S ô Y y Œ ™ d ž 1¶ d d Â 8 h.

1. **Proses Dekripsi pada Sistem Kriptografi AES**

Untuk mengubah kembali cipherteks menjadi plainteks maka dilakukan proses dekripsi dengan menggunakan transformasi invers semua transformasi dasar yang digunakan pada algoritma enkripsi AES. Setiap transformasi dasar AES memiliki transformasi invers, yaitu L: invsubbytes, invshiftrows, dan invmixcolumns.Dari proses dekripsi yang dilaksanakan 10 ronde didapat:

****

Plainteks yang di konversi menjadi bentuk ASCII menjadi: “Kriptografi AES”. Maka pada Algoritma kriptografi AES untuk Plainteks= “Kriptografi AES”dan kunci= “Aditia” didapat Cipherteks= “š S ô Y yŒ ™ d ž 1¶ d d Â 8 h”.

### Algoritma Kriptografi Hybrid

Kriptografi hybrid melakukan enkripsi secara berlapis. Hal ini secara teori akan membuat pemecahannya menjadi semakin kompleks. Algoritma kriptografi hybrid (hybrid cryptosystem) merupakan gabungan antara kriptografi simetris dan kriptografi asimetris. Permasalahan yang menarik pada bidang keamanan informasi adalah adanya trade off antara kecepatan dengan kenyamanan. Salah satu contohnya adalah bidang kriptografi. Tetapi hal ini dapat diatasi dengan penggunaan kriptografi hybrid. Kriptografi hybrid sering dipakai karena memanfaatkan keunggulan kecepatan pemrosesan data oleh algoritma kriptrografi simetrik dan kemudahan transfer kunci menggunakan algoritma kriptografi asimetrik. Hal ini mengakibatkan peningkatan kecepatan tanpa mengurangi kenyamanan serta keamanan.

1. **Prosedur pada Sistem Kriptografi Hybrid**

Kriptografi hybrid menggabungkan kedua buah sistem kriptografi simetris dan sistem kriptografi asimetris. Katakan ada dua pihak yang akan melakukan komunikasi yaitu Alice (Pengirim Pesan) dan Bob (Penerima Pesan). Pertama-tama, Bob sang penerima pesan akan membuat (*Generate*) kunci RSA terlebih dahulu yang terdiri dari kunci publik dan kunci privat. Lalu Alice sebagai pengirim pesan akan membuat sebuah Session Key menggunakan sistem kriptografi simetris AES. Tahap berikut nya Alice akan mengenkripsi Session Key yang sudah dibuat tadi dengan Public Key milik Bob yang mana siapapun boleh mengetahuinya. Lalu Alice mengirim Session Key yang sudah dienkripsi tadi oleh Public Key milik Bob dan mengirim nya kepada Bob. Setelah Bob menerima Session Key yang sudah dienkripsi oleh Alice tadi Bob mendekripsi Session Key tersebut dengan Private Key miliknya sehingga Bob dapat mengetahui Session Key yang akan digunakan untuk mengirim pesan nanti. Setelah kedua belah pihak Alice dan Bob sama-sama sudah memiliki Session Key yang akan digunakan maka Alice dan Bob sudah dapat melakukan komunikasi, bertukar pesan atau informasi rahasia secara aman dan cepat dengan memanfaatkan kedua sistem kriptografi simetris dan sistem kriptografi asimetris tadi atau sebutan nya yaitu Sistem Kriptografi Hybrid.

### Bahasa Pemrograman Python

1. Python merupakan interpreter bahasa pemrograman tingkat tinggi berbasis objek dengan semantik yang dinamis, dimana bersifat freeware atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan source codenya, debugger dan profiler, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, antarmuka pengguna grafis (GUI), dan basis datanya. Python dapat digunakan dalam beberapa sistem operasi, seperti kebanyakan sistem UNIX, PCs (DOS, Windows, OS/2), Macintosh, dan lainnya. Pada kebanyakan sistem operasi linux, bahasa pemrograman ini menjadi standarisasi untuk disertakan dalam paket distribusinya.

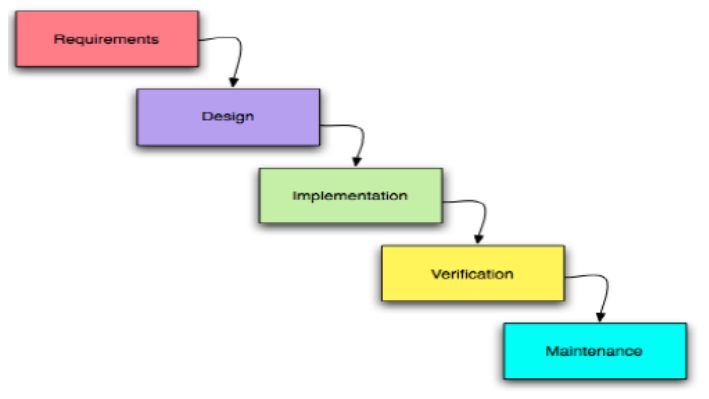
### Aplikasi

## Metodogi Yang Digunakan

Sub bab ini menjelaskan tentang landasan teoritis dari metodologi yang digunakan yakni model *waterfall* yang digunakan dalam perancangan sistem. Pada sub bab ini juga dijabarkan tools atau alat yang digunakan untuk tahapan perancangan yaitu menggunakan SDLC.

### Pengembangan Perangkat Lunak (*Waterfall*)

Menurut Imam Fahrurrozi dan Azhari, Model *waterfall* adalah proses pengembangan perangkat lunak tradisional yang umum digunakan dalam proyek-proyek perangkat lunak yang paling pembangunan. Ini adalah model sekuensial, sehingga penyelesaian satu set kegiatan menyebabkan dimulainyaaktivitas berikutnya. Hal ini disebut *waterfall* karena proses mengalir "secara sistematis dari satu tahap ke tahap lainnya dalam mode ke bawah ". Membentuk kerangka kerja untuk pengembangan perangkat lunak. Beberapa varian dari model ada, setiap label yang berbeda menggunakan untuk setiap tahap. Secara umum, bagaimanapun, model ini dianggap memiliki enam tahap yang berbeda seperti yang ditunjukkan pada Gambar yaitu: analisis Kebutuhan, desain, implementasi, verifikasi, instalasi dan pemeliharaan.



Gambar *Waterfall* Model

1. Kebutuhan berbasis Pengujian

Ini adalah langkah pertama dan paling penting dari model *waterfall*. Ini melibatkan pengumpulan informasi mengenai solusi akhir dari kebutuhan pelanggan pelanggan dan pemahaman. Ini melibatkan definisi yang jelas tentang tujuan pelanggan, harapan terhadap proyek dan masalah produk akhir diharapkan untuk memecahkan. "Analisis meliputi pemahaman konteks bisnis pelanggan dan kendala, fungsi produk harus melakukan, tingkat kinerja itu harus mematuhi dan sistem eksternal itu harus sesuai dengan".

Elisitasi persyaratan adalah proses mengumpulkan informasi dari para pemangku kepentingan dari sistem. Beberapa teknik digunakan untuk elisitasi adalah pelanggan wawancara, prototyping cepat, kasus penggunaan dan brainstorming. Kasus penggunaan umumnya persyaratan fungsional dari sistem.

1. Desain

Langkah ini dimulai dengan menggunakan informasi yang ditangkap di SRS. Ini dapat dianggap sebagai memberikan solusi untuk masalah di lingkup menggunakan sumber daya yang tersedia. Tahap ini terdiri dari bagaimana perangkat lunak akan dibangun, dengan kata lain perencanaan solusi perangkat lunak. Para pemangku kepentingan yang terlibat dalam modul ini adalah para perancang sistem. Desain perangkat lunak mungkin mencakup desain sistem dan desain komponen. "Tahap desain melibatkan mendefinisikan perangkat keras dan perangkat lunak arsitektur, menentukan kinerja dan parameter keamanan, merancang kontainer penyimpanan data dan kendala, memilih IDE dan bahasa pemrograman, dan menunjukkan strategi untuk menghadapi masalah-masalah seperti penanganan eksepsi, pengelolaan sumber daya dan konektivitas antarmuka". Di sinilah desain pengguna sistem juga diakui tergantung pada kemudahan akses dan navigasi. Output dari tahap ini adalah satu atau lebih software desain deskripsi (SDD) yang berfungsi sebagai masukan untuk tahap berikutnya. Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

1. Implementasi

Masukan dari fase ini adalah SDD sistem. "Di sinilah perkembangan aktual sistem terjadi sesuai dengan spesifikasi desain. Langkah ini dilakukan oleh pengembang, desainer interface dan stakeholder lainnya dengan menggunakan alat seperti compiler, debugger, penerjemah dan editor media. Output dari langkah ini adalah komponen produk satu atau lebih yang dibangun berdasarkan standar yang telah ditetapkan coding dan perbaikan, pengujian dan terintegrasi untuk memenuhi kebutuhan arsitektur sistem ".

1. Pengujian: Verifikasi dan Validasi

Pada fase ini kedua komponen individu dan solusi terintegrasi yang diverifikasi untuk melihat itu adalah bug gratis dan memenuhi spesifikasi kebutuhan perangkat lunak. Tester adalah stakeholder yang terlibat dalam fase model. Uji kasus ditulis untuk mengevaluasi apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem.

Pengujian dapat dikategorikan ke dalam unit testing (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas). Cacat yang ditemukan pada tahap ini diberikan sebagai umpan balik kepada para pengembang yang pada gilirannya memperbaiki masalah. Ini adalah tahap di mana produk dikembangkan didokumentasikan. Buku pedoman ini diterbitkan pada fase ini.

1. Maintenance

Instalasi Fase ini terjadi setelah akhir-produk telah diuji dan disetujui oleh pelanggan. "Langkah ini melibatkan penyusunan sistem atau produk untuk instalasi dan penggunaan di lokasi pelanggan". Pengiriman produk dilakukan melalui internet atau melalui metode fisik. Sejumlah revisi biasanya ditandai samping diserahkan untuk memfasilitasi update atau perubahan pada tahap berikutnya .

Pemeliharaan Ini adalah tahap akhir dari model waterfall dan terjadi setelah instalasi sistem produk / di lokasi pelanggan. "Tahap ini melibatkan membuat modifikasi pada sistem atau komponen individu untuk mengubah atribut atau meningkatkan kinerja sistem". Modifikasi yang muncul karena perubahan permintaan dipicu oleh pelanggan atau cacat yang ditemukan saat menggunakan sistem secara real time. Nomor revisi diperbarui dalam setiap rilis pemeliharaan .

### Unified Modeling Language (UML)

Menurut Prastuti Sulistyorini, *Unified Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

1. *Class Diagram*

*Class Diagram* bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi serta relasi.

1. *Object Diagram*

*Object Diagram* bersifat statis. Diagram ini memperlihatkan objek-objek serta relasi antar objek. Diagram objek memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai pada diagram kelas.

1. *Use case Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

1. *Sequence Diagram* (Diagram urutan)

Diagram ini bersifat dinamis. *Sequence Diagram* merupakan diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

1. *Collaboration Diagram*

Diagram ini bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek – objek yang menerima serta mengirim pesan (*message*).

1. *Statechart Diagram*

Diagram ini bersifat dinamis. Diagram ini memperlihatkan state – state pada sistem, memuat state, transisi, event, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem – sistem yang reaktif.

1. *Activity Diagram*

Diagram ini bersifat dinamis. Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

1. *Component Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan organisasi serta kebergantungan pada komponen – komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas- kelas, antarmuka – antarmuka serta kolaborasi – kolaborasi.

1. *Deployment Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (saat *run time*). Dengan ini memuat simpul – simpul (*node*) beserta komponen – komponen yang ada di dalamnya. *Deployment diagram* berhubungan erat dengan diagram kompoen dimana deployment diagram memuat satu atau lebih komponen – komponen. Diagram ini sangat berguna saat aplikasi berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*)

### Teknik Pengumpulan Data Yang Digunakan

1. **Wawancara**

Menurut Rachmawati, Wawancara pada penelitian kualitatif memiliki sedikit perbedaan dibandingkan dengan wawancara lainnya seperti wawancara pada penerimaan pegawai baru, penerimaan mahasiswa baru, atau bahkan pada penelitian kuantitatif. Wawancara pada penelitian kualitatif merupakan pembicaraan yang mempunyai tujuan dan didahului beberapa pertanyaan informal.

Wawancara penelitian lebih dari sekedar percakapan dan berkisar dari informal ke formal. Walaupun semua percakapan mempunyai aturan peralihan tertentu atau kendali oleh satu atau partisipan lainnya, aturan pada wawancara penelitian lebih ketat. Tidak seperti pada percakapan biasa, wawancara penelitian ditujukan untuk mendapatkan informasi dari satu sisi saja, oleh karena itu hubungan asimetris harus tampak. Peneliti cenderung mengarahkan wawancara pada penemuan perasaan, persepsi, dan pemikiran partisipan.

Uraian berikut akan menggambarkan jenis wawancara, jenis pertanyaan, lama waktu wawancara, dan prosedur melakukan wawancara pada penelitian kualitatif. Penjelasan tentang pengumpulan data merupakan hal yang penting karena akan menuntun pembaca memahami proses penelitian secara tepat.

Jenis wawancara :

1. Wawancara tidak berstruktur tidak berstandard, informal, atau berfokus

Dimulai dari pertanyaan umum dalam area yang luas pada penelitian. Wawancara ini biasanya diikuti oleh suatu kata kunci, agenda atau daftar topik yang akan dicakup dalam wawancara. Namun tidak ada pertanyaan yang ditetapkan sebelumnya kecuali dalam wawancara yang awal sekali.

Jenis wawancara ini bersifat fleksibel dan peneliti dapat mengikuti minat dan pemikiran partisipan. Pewawancara dengan bebas menanyakan berbagai pertanyaan kepada partisipan dalam urutan manapun bergantung pada jawaban. Hal ini dapat ditindaklanjuti, tetapi peneliti juga mempunyai agenda sendiri yaitu tujuan penelitian yang dimiliki dalam pikirannya dan isyu tertentu yang akan digali.

Wawancara jenis ini terutama cocok bila peneliti mewawancarai partisipan lebih dari satu kali. Wawancara ini menghasilkan data yang terkaya, tetapi juga memiliki *dross rate* tertinggi, terutama apabila pewawancaranya tidak berpengalaman. *Dross rate* adalah jumlah materi atau informasi yang tidak berguna dalam penelitian.

1. Wawancara Semi Berstruktur

Wawancara ini dimulai dari isu yang dicakup dalam pedoman wawancara. Pedoman wawancara bukanlah jadwal seperti dalam penelitian kuantitatif. Sekuensi pertanyaan tidaklah sama pada tiap partisipan bergantung pada proses wawancara dan jawaban tiap individu. Namun pedoman wawancara menjamin peneliti dapat mengumpulkan jenis data yang sama dari partisipan. Peneliti dapat menghemat waktu melalui cara ini. *Dross rate* lebih rendah daripada wawancara tidak berstruktur. Peneliti dapat mengembangkan pertanyaan dan memutuskan sendiri mana isu yang dimunculkan.

1. Wawancara berstruktur atau berstandard.

Peneliti kualitatif jarang menggunakan jenis wawancara ini. Beberapa keterbatasan pada wawancara jenis ini membuat data yang diperoleh tidak kaya. Jadwal wawancara berisi sejumlah pertanyaan yang telah direncanakan sebelumnya. Tiap partisipan ditanyakan pertanyaan yang sama dengan urutan yang sama pula. Jenis wawancara ini menyerupai kuesioner survei tertulis. Wawancara ini menghemat waktu dan membatasi efek pewawancara bila sejumlah pewawancara yang berbeda terlibat dalam penelitian. Analisis data tampak lebih mudah sebagaimana jawaban yang dapat ditemukan dengan cepat. Umumnya, pengetahuan statistik penting dan berguna untuk menganalisis jenis wawancara ini. Namun jenis wawancara ini mengarahkan respon partisipan dan oleh karena itu tidak tepat digunakan pada pendekatan kualitatif. Wawancara berstruktur bisa berisi pertanyaan terbuka, namun peneliti harus diingatkan terhadap hal ini sebagai isu metodologis yang akan mengacaukan dan akan jadi menyulitkan analisisnya. Peneliti kualitatif menggunakan pertanyaan yang berstruktur ini hanya untuk mendapatkan data *sosio- demografik*, seperti usia, lamanya kondisi yang dialami, lamanya pengalaman, pekerjaan, kualifikasi, dsb. Kadang komite etik menanyakan jadwal wawancara yang ditentukan sebelumya sehingga mereka dapat menemukan alur penelitian yang sebenarnya. Pada kasus ini, pedoman wawancara semi berstruktur lebih dianjurkan.

# BAB III ANALISIS DAN PERANCANGAN

Pada bab ini akan membahas analisis dan perancangan mulai dari gambaran perangkat lunak, analisis fungsional, perancangan antarmuka, perancangan arsitektural dan perancangan prosedural.

## Gambaran Perangkat Lunak

Sistem baru yang akan dibuat merupakan implementasi Sistem kriptografi hybrid yang mana merupakan sistem kriptografi modern yang baru dari sebelum nya yaitu sistem kriptografi simetris dan asimetris. Sistem kriptografi hybrid merupakan gabungan dari kedua sistem sebelumnya yang mana tujuan penggunaan nya adalah memanfaatkan kelebihan kedua sistem sebelumnya untuk mengatasi kelemahan kedua sistem tersebut. Sistem ini menggunakan algoritma kriptografi simetris AES (Advanced Encryption System) dan algoritma kriptografi asimetris RSA (Rivest Shamir Adleman).

### Flowchart Algoritma Kriptografi Hybrid

## Analisis Fungsional

### DFD

## Perancangan Antarmuka

### Perancangan Antarmuka Main Menu

Perancangan antarmuka halaman utama atau main menu ditunjukkan pada gambar berikut.



Gambar perancangan antarmuka *Main Menu*

Gambar diatas merupakan perancangan tampilan antarmuka halaman utama dari program kriptografi hybrid. Terdapat 3 tombol yaitu tombol session key exchange untuk melakukan kegiatan pertukaran kunci sesi, tombol message encryption & decryption untuk melakukan kegiatan enkripsi dan dekripsi pesan dan terakhir ada tombol quit untuk keluar dari program.

### Perancangan Antarmuka Session Key Exchange

Perancangan antarmuka untuk pertukaran kunci sesi ditunjukkan pada gambar berikut.



Gambar perancangan antarmuka *Session Key Exchange*

Pada gambar diatas ditampilkan perancangan antarmuka untuk menu kegiatan session key exchange atau pertukaran kunci sesi. Di menu ini terdapat 4 tombol yaitu tombol generate session key untuk membuat kunci sesi baru, tombol generate public & private key untuk membuat kunci publik dan kunci privat baru, tombol encrypt session key untuk menenkrip kunci sesi yang sudah dibuat dan tombol decrypt session key untuk mendekrip kunci sesi yang sudah dienkrip sebelumnya.

### Perancangan Antarmuka Generate Session Key

Perancangan antarmuka untuk pembuatan kunci sesi baru ditunjukkan pada gambar berikut.



Gambar perancangan antarmuka *Generate Session Key*

Pada gambar diatas ditampilkan perancangan antarmuka untuk membuat kunci sesi baru dimana ketika tombol generate session key diklik maka pada entry generated session key akan dibuat kunci sesi baru secara otomatis dan random.

### Perancangan Antarmuka Generate Public & Private Key

Perancangan antarmuka untuk pembuatan kunci publik dan privat baru ditunjukkan pada gambar berikut.



Gambar perancangan antarmuka *Generate Public & Private Key*

Pada gambar diatas ditunjukkan perancangan antarmuka untuk pembuatan kunci publik dan privat. Ketika tombol diklik maka sebuah kunci publik baru dan sebuah kunci privat baru akan dibuat secara otomatis dan random.

### Perancangan Antarmuka Encrypt Session Key

Perancangan antarmuka untuk menu kegiatan enkripsi kunci sesi dengan kunci publik ditunjukkan pada gambar berikut.



Gambar perancangan antarmuka *Encrypt Session Key*

Pada gambar diatas ditunjukkan perancangan antarmuka untuk kegiatan mengenkripsi kunci sesi dengan kunci publik. Kunci sesi dimasukkan ke dalam entry session key lalu kunci publik dimasukkan ke dalam text widget public key lalu klik tombol encrypt dan kunci sesi akan dienkrip dengan kunci publik.

### Perancangan Antarmuka Decrypt Session Key

Perancangan antarmuka untuk menu kegiatan dekripsi kunci sesi dengan kunci privat ditunjukkan pada gambar berikut.



Gambar perancangan antarmuka *Decrypt Session Key*

Pada gambar diatas ditunjukkan perancangan antarmuka untuk kegiatan mendekripsi kunci sesi dengan kunci privat. Kunci sesi yang sudah dienkrip dimasukkan ke dalam text widget encrypted session key lalu kunci privat dimasukkan ke dalam text widget private key lalu klik tombol decrypt dan kunci sesi akan didekrip dengan kunci privat.

### Perancangan Antarmuka Message Encryption & Decryption

Perancangan antarmuka untuk menu kegiatan enkripsi dan dekripsi pesan ditunjukkan pada gambar berikut.



Gambar perancangan antarmuka *Message Encrypt & Decrypt*

Pada gambar diatas ditunjukkan perancangan antarmuka untuk menu mengenkripsi dan mendekripsi pesan. Terdapat 2 tombol yaitu tombol encrypt message dan tombol decrypt message. Tombol encrypt message untuk membuka menu untuk kegiatan mengenkripsi pesan. Tombol decrypt message untuk membuka menu untuk kegiatan mengdekripsi pesan.

### Perancangan Antarmuka Encrypt Message

Perancangan antarmuka untuk menu kegiatan mengenkripsi pesan ditunjukkan pada gambar berikut.



Gambar perancangan antarmuka *Message Encrypt & Decrypt*

Pada gambar diatas ditunjukkan perancangan antarmuka untuk menu kegiatan mengenkripsi pesan. Masukan kunci sesi ke dalam entry session key lalu masukan pesan yang akan dienkrip dengan kunci sesi. Setelah itu klik tombol encrypt dan pesan tersebut akan dienkrip dan hasil nya ditampilkan di text widget message.

### Perancangan Antarmuka Decrypt Message

Perancangan antarmuka untuk menu kegiatan mendekripsi pesan ditunjukkan pada gambar berikut.



Gambar perancangan antarmuka *Message Encrypt & Decrypt*

Pada gambar diatas ditunjukkan perancangan antarmuka untuk menu kegiatan mendekripsi pesan. Masukan kunci sesi ke entry session key lalu masukan pesan yang akan didekripsi dengan kunci sesi. Klik tombol decrypt maka pesan tersebut akan didekrip dan ditampilkan di text widget decrypted message.

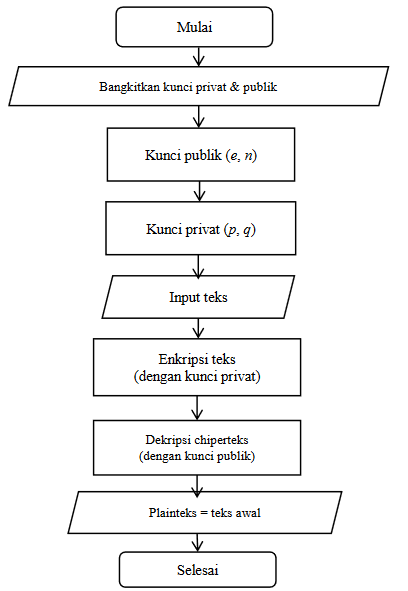
## Perancangan Arsitektural

### Structure Chart

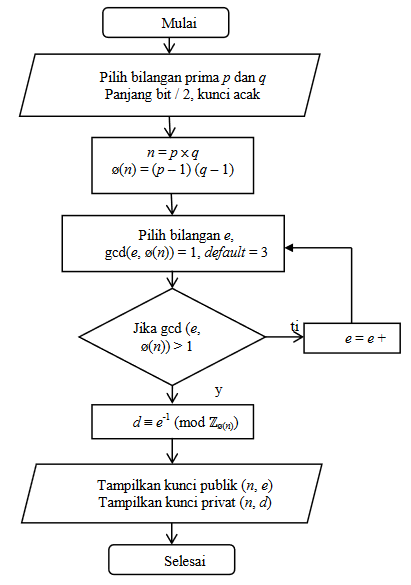
## Perancangan Prosedural

### Pseudocode

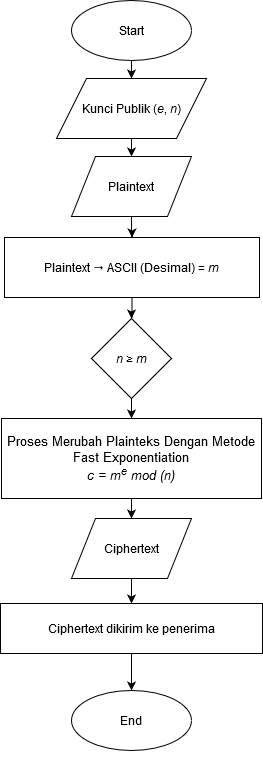
1. **Gambaran Perangkat Lunak**
2. Flowchart Algoritma RSA



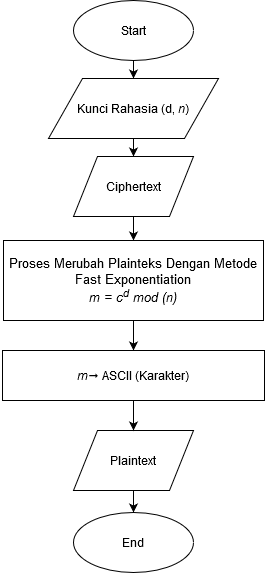
1. Flowchart Pembangkitan Kunci Untuk RSA



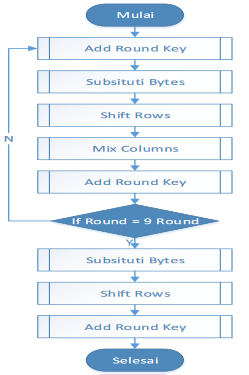
1. Flowchart Algoritma Enkripsi RSA



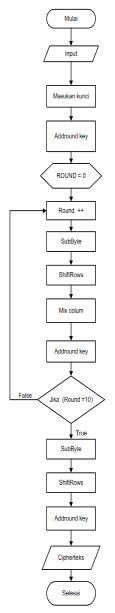
1. Flowchart Algoritma Dekripsi RSA



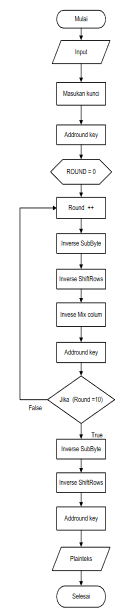
1. Flowchart Algoritma AES



1. Flowchart Pembangkitan Kunci Untuk AES
2. Flowchart Algoritma Enkripsi AES



1. Flowchart Algoritma Dekripsi AES



1. **Analisis Fungsional**
2. DFD
3. **Perancangan Antarmuka**
4. Rancangan Interface Halaman Utama
5. **Perancangan Arsitektural**
6. Structure Chart
7. **Perancangan Prosedural**
8. Pseudocode

# BAB IV IMPLEMENTASI DAN PENGUJIAN

## Implementasi

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

### Lingkup dan Batasan

Lingkup dan batasan dari perangkat lunak yang telah penulis buat adalah sebagai berikut :

1. Perangkat lunak dibuat dengan menggunakan arsitektur *microservices*.

### Kebutuhan Sumber Daya

Untuk menggunakan perangkat lunak ini dibutuhkan spesifikasi yang mampu mendukung pengopersiannya, beberapa komponen yang dibutuhkan adalah sebagai berikut :

1. Kebutuhan Minimum Hardware

Untuk menjalankan perangkat lunak yang telah dibuat, dibutuhkan beberapa spesifikasi kebutuhan hardware sebagai berikut :

1. Processor : Quadcore 3,0 GHz
2. RAM : 8 GB
3. SSD : 256 GB
4. VGA : 2 GB
5. Monitor : 22’ inch FHD
6. Keyboard : Wireless Keyboard
7. Mouse : Wireless Mouse
8. Kebutuhan Minimum Software

Perangkat Lunak yang telah dibuat juga membutuhkan software yang mendukung dalam proses pembuatan maupun dalam penggunaan perangkat lunak tersebut.

Kebutuhan akan software atau perangkat lunak untuk mengembangkan aplikasi ini adalah sebagai berikut :

1. Sistem operasi : Windows 10 32/64 Bit
2. Text Editor : Visual Studio Code
3. Command Line : Command Prompt
4. Bahasa Pemrograman : Python 3.7.3
5. Spesifikasi Brainware

Spesifikasi kebutuhan brainware untuk mengembangkan aplikasi ini yaitu :

1. Memahami dasar python
2. Memahami dasar kriptografi simetris
3. Memahami dasar kriptografi asimetris
4. Memahami dasar penggunaan aplikasi desktop

### Implementasi Aplikasi

## Pengujian

### Lingkup dan Lingkungan

### Kebutuhan Sumber Daya

### Hasil Pengujian

# BAB V KESIMPULAN DAN SARAN

## Kesimpulan

Dari penelitian dan implementasi kriptografi hybrid tersebut maka dapat diambil kesimpulan sebagai berikut:

1. Penerapan sistem kriptografi hybrid untuk mengatasi kelemahan sistem kriptografi simetris dan asimetris dapat diaplikasikan dan diimplementasikan ke dalam aplikasi desktop berbasis python.

## Saran

Setelah melakukan implementasi kriptografi hybrid, penulis memberikan saran yang ditujukan untuk tahap pengembangan ke depannya. Berikut poin-poin saran yang disampaikan:

1. Menambahkan fitur untuk mengenkripsi dan mendekripsi tidak hanya sebuah pesan tetapi data – data seperti dokumen, gambar dll.
2. Penambahan fitur untuk memilih tingkat kekuatan kunci baik kunci simetris (session key) maupun asimetris (public key dan private key).