# IMPLEMENTASI HYBRID KRIPTOGRAFI MODERN RIVEST SHAMIR ADLEMAN DAN NOEKEON UNTUK PENGAMANAN DATA

**SKRIPSI**

**OLEH :**

**ARYA KUSUMA**
**NPM : 18010005**

**PROGRAM STUDI INFORMATIKA**
**FAKULTAS ILMU KOMPUTER**
**UNIVERSITAS DEHASEN**
**BENGKULU**
**2025**

# IMPLEMENTASI HYBRID KRIPTOGRAFI MODERN RIVEST SHAMIR ADLEMAN DAN NOEKEON UNTUK PENGAMANAN DATA

**SKRIPSI**

OLEH :

**ARYA KUSUMA**
**NPM : 18010005**

Diajukan Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Program Studi Informatika

**PROGRAM STUDI INFORMATIKA**
**FAKULTAS ILMU KOMPUTER**
**UNIVERSITAS DEHASEN**
**BENGKULU**
**2025**

**IMPLEMENTASI HYBRID KRIPTOGRAFI MODERN RIVEST SHAMIR ADLEMAN DAN NOEKEON UNTUK PENGAMANAN DATA**

**SKRIPSI**

OLEH :

**ARYA KUSUMA**
**NPM : 18010005**

**Disetujui Oleh :**

| Pembimbing Utama | Pembimbing Pendamping |
|---|---|
| **Prahasti, S.Kom, M.Kom** | **Desi Mahdalena, S.Kom, M.Kom** |
| **NIDN. 02.140482.02** | **NIDN. 02.081294.03** |

**Mengetahui,**
**Ketua Program Studi Informatika**

**Devi Sartika, S.Kom, M.Kom**
**NIDN. 02.030386.05**

iii

**IMPLEMENTASI HYBRID KRIPTOGRAFI MODERN RIVEST SHAMIR ADLEMAN DAN NOEKEON UNTUK PENGAMANAN DATA**

**SKRIPSI**

OLEH :

**ARYA KUSUMA**
**NPM : 18010005**

Telah dipertahankan di depan Tim Penguji Universitas Dehasen Bengkulu Pada :

Hari : Kamis
Tanggal : 12 Juni 2025
Tempat : Ruang Sidang Universitas Dehasen Bengkulu

Skripsi Telah Diperiksa dan Disahkan Oleh :

| Penguji | Nama | NIDN | Tanda Tangan |
|---|---|---|---|
| Ketua | Prahasti, S.Kom, M.Kom | 02.140482.02 | |
| Anggota | Desi Mahdalena, S.Kom, M.Kom | 02.081294.03 | |
| Anggota | Indra Kanedi, S.Kom, M.Kom | 02.100581.01 | |
| Anggota | Devi Sartika, S.Kom, M.Kom | 02.030386.05 | |

**Mengetahui,**

**Dekan**
**Fakultas Ilmu Komputer**

**Khairil, S.Kom, M.Kom**
**NIDN : 02.130475.01**



iv

# DAFTAR RIWAYAT HIDUP

Penulis bernama arya kusuma, dilahirkan di desa air bening pada tanggal 21 oktober 1997. Anak pertama dari dua bersaudara, ayah bernama hamdani dan ibu bernama yopi wardani.

Penulis menempuh pendidikan dimulai dari sekolah dasar (SD) Negeri 46 air bening pada tahun 2005 dan lulus pada tahun 2011. Kemudian melanjutkan ke tingkat sekolah menengah pertama (smp) Negeri 1 Bermani Ulu Raya pada tahun 2011 dan lulus pada tahun 2013, kemudian melanjutkan pendidikan Sekolah Menengah Atas (SMA) Negeri 3 Curup Utara pada tahun 2013, tetapi hanya satu tahun berpindah ke Sekolah Menengah Kejuruan (SMK) Negeri 4 Rejang Lebong dan lulus pada tahun 2018.

Selanjutnya melanjutkan pendidikan perguruan tinggi di Universitas Dehasen (UNIVED) Kota Bengkulu dengan mengambil jurusan Studi Informatika pada Fakultas Ilmu Komputer untuk jenjang Strata 1 (S1).

## MOTTO DAN PERSEMBAHAN

**Motto :**

- ❖ Memulai dengan penuh keyakinan, Menjalankan dengan ikhlas, Menyelesaikan dengan kebahagiaan
- ❖ Ambil lah kebaikan dari apa yang didengar, jangan melihat siapa yang mengatakannya.
- ❖ Tak perlu khawatir akan bagaimana alur cerita pada jalan ini, perankan saja, Tuhan ialah sebaik-baiknya sutradara.

**Persembahan :**

Kupersembahkan kado kecil ini dengan sepenuh hati untuk :

1. Ayah dan ibu (hamdani dan yopi wardani) yang telah memberikan kasih sayang dan selalu memberi dukungan, dan tiada henti memberikan do'a dan selalu bersabar sampai saya menyelesaikan pendidikan ini
2. Untuk adikku (Putri Sari Rezaki) terimakasih buat dukungan dan supportnya.
3. Terimakasih untuk Kakek dan Nenek yang telah memberikan dukungan dan do'a terbaik hingga saat ini
4. Untuk seseorang yang selalu menyemangati (Lola Oktapiani, S.Pd) yang selalu memberikan semangat dan motivasi
5. Para dosen dan pembimbing (Ibu Devi Sartika, M.Kom, Ibu Prahasti,M.Kom, dan Ibu Desi Mahdalena,M.Kom) yang telah banyak memberikan bimbingan kepada saya dalam menyelesaikan skripsi ini.
6. Almamater kuning yang aku banggakan

# ABSTRAK

## IMPLEMENTASI HYBRID KRIPTOGRAFI MODERN RIVEST SHAMIR ADLEMAN DAN NOEKEON UNTUK PENGAMANAN DATA

**Oleh :**

Arya Kusuma [1]
Prahasti, S.Kom, M.Kom[2]
Desi Mahdalena, S.Kom, M.Kom[3]

Perkembangan teknologi informasi telah mengubah berbagai sektor secara signifikan, termasuk bisnis, industri, kesehatan, dan pendidikan. Pentingnya keamanan dan integritas data merupakan hal yang krusial bagi sebuah sistem. Enkripsi adalah solusi umum untuk memastikan keamanan data dan meminimalkan risiko penyalahgunaan data. Metode seperti Rivest Shamir Adleman (RSA) dan Noekeon telah digunakan untuk meningkatkan keamanan data.

RSA adalah metode kriptografi kunci publik yang membutuhkan dua kunci berbeda untuk penyandian dan penguraian. Noekeon adalah cipher 128-bit dengan 16 bit dan 16 bit. Penelitian ini bertujuan untuk menganalisis arsitektur sistem dan metode enkripsi dan dekripsi untuk data teks dengan menggunakan algoritma RSA dan Noekeon. Hasil penelitian menunjukkan bahwa penggabungan algoritma RSA dan Noekeon dapat meningkatkan keamanan data

Hasil implementasi kombinasi pengamanan pesan teks menggunakan RSA dan *Noekeon* menunjukkan hasil yang cukup baik, dimana pesan teks yang ter enkripsi memiliki keunggulan dalam proses autentikasi dimana menerapkan kunci *public* dan kunci *private* dari algoritma RSA serta memiliki keunggulan dalam kompleksitas chiperteks dari algoritma *Noekeon*

Kata kunci : *Hybrid Kripto, Rivest Shamir Adleman, Noekeon*

1. Mahasiswa
2. Pembimbing

## *ABSTRACT*

### *The Implementation of Modern Hybrid Cryptography Rivest Shamir Adleman and Noekeon for Data Security*

***By:***
***Arya Kusuma*** [1]
***Prahasti*** [2]
***Desi Mahdalena*** [3]

*The development of information technology has significantly changed various sectors, including business, industry, health, and education. The importance of data security and integrity is crucial for a system. Encryption is a common solution to ensure data security and minimize the risk of data misuse. Methods such as Rivest Shamir Adleman (RSA) and Noekeon have been used to improve data security. RSA is a public-key cryptography method that requires two different keys for encryption and decryption. Noekeon is a 128-bit cipher with 16 bits and 16 bits. This study aims to analyze the system architecture and encryption and decryption methods for text data using the RSA and Noekeon algorithms. The results show that combining RSA and Noekeon algorithms can improve data security. The implementation of text message security using RSA and Noekeon shows quite good results, where encrypted text messages have advantages in the authentication process by applying public and private keys from RSA algorithm and have advantages in the complexity of the ciphertext from the Noekeon algorithm.*

*Keywords: Hybrid Cryptography, Rivest Shamir Adleman, Noekeon.*

*1. Student*
*2. Supervisors*

# SURAT PERNYATAAN ORISINILITAS

## PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Arya Kusuma

NPM : 18010005

Prodi : Informatika

Menyatakan dengan sesungguhnya bahwa :

1. Selama melakukan penelitian dan pembuatan skripsi ini saya tidak melakukan pelanggaran etika akademik dalam bentuk apapun atau pelanggaran lainnya yg bertentang dengan etika akademik
2. Skripsi yang saya buat merupakan karya ilmiah saya sebagai penulis, bukan jiplakan atau karya orang lain
3. Apabila di kemudian hari ditemukan bukti yang meyakinkan bahwa dalam proses pembuatan skripsi ini terdapat pelanggaran etika akademik atau skripsi ini hasil jiplakan atau skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi akademik yang diteteapkan oleh Universitas Dehasen Bengkulu

Demikian pernyataan ini saya buat dengan sebenarnya untuk di pergunakan bilamana perlu

Bengkulu, 12 Juni 2024

Yang menyatakan,


**ARYA KUSUMA**

**NPM.18010005**

# KATA PENGANTAR

Puji Syukur saya panjatkan kehadirat Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia-NYA, sehingga skripsi yang berjudul "**Implementasi Hybrid Kriptografi Modern Rivest Shamir Adleman Dan Noekeon Untuk Pengamanan Data**" dapat diselesaikan dalam waktu yang telah ditetapkan.

Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah banyak membantu penulis selama penyusunan skripsi ini kepada :

1. Bapak Prof. Dr. Husaini, SE., M.Si, Ak, CA, CRP selaku Rektor Universitas Dehasen Bengkulu
2. Bapak Khairil, S.Kom, M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Dehasen Bengkulu.
3. Ibu Devi Sartika, M.Kom selaku Ketua Prodi Informatika Universitas Dehasen Bengkulu.
4. Ibu Prahasti, S.Kom, M.Kom selaku pembimbing utama yang telah membimbing dengan sabar dan memberikan masukan serta saran kepada penulis
5. Ibu Desi Mahdalena, S.Kom, M.Kom Selaku pembimbing pendamping yang telah memberikan masukan dan saran kepada penulis.
6. Buat teman-teman yang tidak bisa disebutkan satu persatu baik formal dan non formal, terima kasih atas bantuannya selama penyelesaian penulisan skripsi ini

Penulis menyadari masih banyak kekurangan dalam penulisan skripsi ini, namun penulis mengharapkan saran dan kritik yang sifatnya membangun guna menunjang perkembangan ilmu pengetahuan khususnya ilmu komputer.

Bengkulu,  Mei 2025

Arya Kusuma

# DAFTAR ISI

**Halaman**

# DAFTAR GAMBAR

**Gambar** **Halaman**

# DAFTAR TABEL

**Tabel** **Halaman**

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang

Perkembangan pesat teknologi informasi telah membawa perubahan signifikan di berbagai sektor, mulai dari bisnis, industri, kesehatan, hingga pendidikan. Hal ini menuntut adanya peningkatan sistem keamanan data untuk melindungi informasi sensitif dari ancaman yang semakin kompleks. Masalah keamanan dan kerahasiaan data merupakan salah satu aspek penting dari suatu sistem informasi. Salah satu solusi yang banyak digunakan adalah enkripsi, yaitu proses mengubah data menjadi bentuk kode atau acak yang tidak mudah dipahami oleh pihak-pihak yang tidak berwenang. Dengan demikian, kerahasiaan dan integritas informasi dapat terjaga, meminimalisir risiko penyalahgunaan data oleh oknum yang tidak bertanggung jawab

Masalah dalam keamanan tersebut membuat pengguna memiliki kebutuhan tersendiri akan keamanan data. Kebutuhan akan keamanan data yang lebih baik telah melahirkan berbagai jenis metode dan teknik yang dapat digunakan pada kegiatan pengamanan pesan. Beberapa diantaranya adalah metode *Rivest Shamir Adleman* (RSA) dan metode Noekeon. RSA merupakan metode kriptografi pertama yang menggunakan *public key* dalam prosesnya. RSA merupakan metode kriptografi asimetris yang beroperasi pada mode blok. RSA membutuhkan dua kunci yang berbeda pada proses enkripsi dan dekripsi nya sehingga proses enkripsi dan dekripsi hanya dapat

dilakukan oleh pihak yang memiliki kunci yang sesuai. Walaupun kunci enkripsi diketahui oleh pihak yang tidak berhak, pesan tidak dapat di dekripsi menggunakan kunci tersebut. Sedangkan metode Noekeon merupakan cipher blok berulang dengan panjang blok dan panjang kuncinya 128 bit dan terdiri dari transformasi round berulang, diikuti transformasi output. Noekeon memiliki 16 putaran pengulangan, dalam setiap putarannya dilakukan empat buah transformasi yaitu theta, shift offset Pi1 dan Pi2, dan gamma

Penggunaan RSA dan Noekeon secara berurutan akan memanfaatkan kelebihan dari masing-masing algoritma dengan menggabungkan kedua algoritma dalam satu proses yang berkesinambungan dalam mengamankan data. Kekuatan kunci RSA yang dipadukan dengan iterasi perulangan yang dimiliki oleh Noekeon secara hipotesis sistem mampu memberikan bentuk pengamanan data yang lebih baik dibandingkan dengan menggunakan hanya satu dari masing-masing algoritma

Berdasarkan latar belakang diatas maka penulis tertarik untuk melakukan penelitian yang diberi judul **"Implementasi Hybrid Kriptografi Modern Rivest Shamir Adleman Dan Noekeon Untuk Pengamanan Data".**

## 1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang di atas, rumusan masalah yang akan dibahas adalah bagaimana implementasi hybrid kriptografi modern RSA dan Noekeon untuk pengamanan data.

## 1.3 Batasan Masalah

Agar tidak memperluas materi penulisan maka batasan-batasan dan ruang lingkup penulisan antara lain adalah

1. Penelitian ini menerapkan menggunakan berkas *file* teks sebagai *input* proses enkripsi dan dekripsi
2. Bahasa pemrograman yang digunakan untuk membangun sistem adalah Visual Studio.Net 2010.

## 1.4 Tujuan Penelitian

### 1.4.1 Tujuan Umum

Tujuan umum dilakukannya penelitian ini adalah sebagai salah satu syarat untuk menyelesaikan pendidikan Strata Satu (S1) Pada Program Studi Informatika Fakultas Ilmu Komputer

### 1.4.2 Tujuan Khusus

Tujuan khusu dari peneliitan ini untuk mengetahui bentuk implementasi proses dan tahapan dari enkripsi dan dekripsi pada kombinasi RSA dan Noekeon dan kemudian menganalisa hasil dari penerapan kriptografi hybrid tersebut

## 1.5 Manfaat Penelitian

Adapun manfaat dari penelitian antara lain :

1. Memberikan informasi secara rinci kepada pembaca akan fungsi enkripsi dan dekripsi dengan menggunakan Algoritma RSA dan NOEKEON dalam pengamanan data

2. Menghasilkan aplikasi yang dapat digunakan oleh pengguna untuk meng-enkripsi dan men-dekripsi data
3. Menghasilkan referensi pada bidang keamanan digital dan kriptografi

# BAB II

# LANDASAN TEORI

## 2.1 Pengertian Implementasi

Implementasi berasal dari bahasa Inggris "*to implement*" yang artinya mengimplementasikan. Implementasi bukan hanya suatu aktivitas, tetapi implementasi juga merupakan suatu kegiatan yang direncanakan serta dilaksanakan dengan serius dan mengacu pada norma-norma tertentu, guna mencapai tujuan kegiatan (Wahidin, et.all, 2021)

Implementasi adalah pelaksanaan dan penerapan, dimana kedua hal ini bermaksud untuk mencari bentuk tentang hal yang disepakati terlebih dahulu. Tujuan dari implementasi sebuah sistem adalah untuk menyelesaikan desain sistem yang telah disetujui, menguji serta mendokumentasikan program-program dan prosedur sistem yang diperlukan, memastikan bahwa personil yang terlibat dapat mengoperasikan sistem yang baru dan memastikan bahwa konversi sistem lama ke sistem baru dapat berjalan dengan baik dan benar (Gunawan & Kirman, 2019).

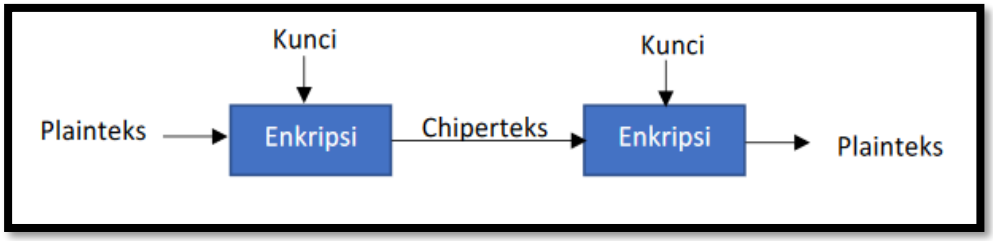
## 2.2 Pengertian Kriptografi

Kriptografi telah dikenal dan dipakai cukup lama sejak kurang lebih tahun 1900 sebelum masehi pada prasasti- prasasti kuburan. Kriptografi sendiri berasal dari kata "*Crypto*" yang berarti rahasia dan "*graphy*" yang berarti tulisan. Jadi, dapat dikatakan kriptografi adalah tulisan yang tersembunyi. Dengan adanya tulisan yang tersembunyi ini, orang-orang yang tidak mengetahui bagaimana tulisan tersebut disembunyikan tidak

akan mengetahui bagaimana cara membaca maupun menerjemahkan tulisan tersebut. *Criptography* berasal dari bahasa Yunani. Menurut bahasanya, istilah tersebut terdiri dari kata kripto dan graphia. Kripto berarti *secret* (rahasia) dan graphia berarti *writing* (tulisan) (Rahmat, Jumadi, & Lianda, 2024)

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasian data, keabsahan data, integritas data, serta autentikasi data. Tetapi tidak semua aspek keamanan informasi dapat diselesaikan dengan kriptografi. Definisi lain menurut kriptografi yaitu seni untuk menjaga keamanan pesan (Dairi, Asih, & Khairunnisa, 2023).

Terdapat dua proses dalam kriptografi yaitu proses enkripsi dan proses dekripsi. Enkripsi merupakan proses untuk melakukan perubahan kode dari yang sebelumnya dapat dipahami oleh manusia untuk membacanya menjadi sesuatu tidak dapat dipahami (*unreadable*) bagi manusia, pengubahan itu bisa mengubah teks asli menjadi susunan karakter maupun simbol dengan susunan yang jauh berbeda dari teks aslinya. (Ridho, Mutia, & Sinaga, 2022). *Sedangkan* dekripsi adalah kebalikan dari kegiatan enkripsi karena tujuan dari deskripsi mengembalikan pesan yang tersandi atau informasi palsu ke pesan asli. Pada proses mengembalikan isi pesan tersamar harus menggunakan kode yang telah disiapkan sebelumnya. Kegiatan perubahan isi pesan dari *plaintext* ke *ciphertext* disebut enkripsi, dan prosedur mengembalikan teks dari *ciphertext* ke *plaintext* disebut dekripsi




**Gambar 2.1 Diagram Enkripsi dan Dekripsi**

Prinsip-prinsip yang mendasari kriptografi yakni:

1. *Secrecy* (kerahasiaan), layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka maupun menghapus informasi yang telah disandi.
2. *Authentication*, berhubungan dengan identifikasi atau pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Dimana informasi yang dikirimkan melalui kanal harus diautentifikasi keaslian, isi datanya, waktu pengiriman dan lain-lain.
3. Hak Akses terhadap suatu *file* atau fasilitas lain dalam sebuah sistem pemrosesan informasi masih dalam area lain dimana gagasan kriptografi telah diterapkan

### 2.2.1 Tujuan Kriptografi

Kriptografi bertujuan untuk layanan keamanan yang memiliki beberapa aspek keamanan yang memiliki beberapa aspek keamanan, antara lain sebagai berikut :

a. *Authentication*

Layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data atau informasi. Fasilitas yang

berkaitan untuk melakukan identifikasi terlebih dahulu antara pengirim dan penerima pesan.

b. *Integrity*

Keuntungan yang didapatkan dalam menggunakan teknik kriptografi yaitu menjamin bahwa pesan akan diterima dalam keadaan masih utuh dan belum mengalami perubahan selama proses pengiriman. Layanan yang mampu mengenali/mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan) data yang tidak sah (oleh pihak lain).

c. *Confidelity*

Layanan agar isi pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima/pihak-pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.

d. *Non-repudiation*

Layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya

### 2.2.2 Jenis Kriptografi

1. Kriptografi Klasik

Kriptografi klasik digunakan sebelum era komputerisasi dan kebanyakan menggunakan teknik kunci simetris. Teknik ini melibatkan pengacakan huruf pada kata terang atau *plaintext*

menggunakan penggantian huruf atau substitusi dan pengacakan posisi huruf atau transposisi. Teknik substitusi adalah mengganti karakter dalam *plaintext* menjadi karakter lain yang hasilnya adalah *ciphertext*. Sedangkan transposisi adalah teknik mengubah *plaintext* menjadi *ciphertext* dengan cara melakukan permutasi pada karakternya. Contoh algoritma kriptografi klasik yaitu: *Caesar Cipher, Vigenere Cipher*, dan *Hill Cipher* (Sasono, et,all, 2023)

2. Kriptografi Modern

Kriptografi Modern merupakan suatu perbaikan dari teknik yang digunakan pada kriptografi klasik. Algoritma di kriptografi modern ini menggunakan pengolahan dan penggunaan simbol biner yang dibentuk dari kode ASCII (*American Standard Code for Information Interchange*) karena berjalan mengikuti operasi komputer digital, sehingga membutuhkan pengetahuan dasar matematika untuk menguasainya. Algoritma ini juga memiliki tingkat kesulitan yang lebih kompleks yang menyebabkan kriptanalis sangat sulit memecahkan *ciphertext* tanpa mengetahui kuncinya. Adapun jenis kunci dalam kriptografi modern terdiri dari 3 yaitu: simetris, asimetris, dan hibrida. Pada kriptogarfi modern terdapat berbagai macam algoritma yang memiliki tujuan untuk mengamankan informasi yang dikirim melalui jaringan computer, contoh kriptografi modern yaitu MD5, RC4, AES dan lain-lain (Sasono, et,all, 2023).

## 2.3 Metode RSA (*Rivest Shamir Adleman*)

Metode RSA pertama kali diperkenalkan pada tahun 1976 oleh tiga peneliti yang berasal dari *Massachusetts Institute of Technology*, yaitu Ron Rivest, Adi Shamir, dan Len Adleman. Nama RSA sendiri diambil dari inisial ketiga peneliti tersebut (Zachary, Sylviani, & Kurniadi, 2024). Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi factor-faktor prima. Pemfaktoran dilakukan unutk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi factor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin (Firmansyah & Permana, 2019).

Pada algoritma RSA terdapat 3 langkah utama yaitu *key generation* (pembangkit kunci), enkripsi dan dekripsi. Kunci pada RSA mencakup dua buah kunci, yaitu *public key* dan *private key*. *Public key* digunakan untuk melakukan enkripsi, dan dapat diketahui oleh orang lain. Sedangkan *private key* tetap dirahasiakan dan digunakan untuk melakukan dekripsi (Wahyudi, et.all, 2022).

Berikut adalah tahapan dalam algoritma kriptografi RSA (Dairi, Asih, & Khairunnisa, 2023) :

1. Pembangkitan Kunci Pada RSA

   Pengkodean RSA membutuhkan dua kunci yang berbeda untuk proses enkripsi dan dekripsi. Bilangan yang dipilih sebagai kunci adalah bilangan prima yang besar dengan pemfaktoran sebuah bilangan hasil dari perkalian dari dua bilangan prima yang besar menjadi dua bilangan prima yang sesuai akan sangat sulit.

2. Dipilih dua buah bilangan prima sembarang yang besar, p dan q. Nilai p dan q harus dirahasiakan
3. Dihitung n = p x q Besaran n tidak perlu dirahasiakan sebaiknya $p \neq q$, sebab jika p = q maka n = p2 sehingga p dapat diperoleh dengan menarik akar pangkat dua dari n
4. Dihitung m = (p – 1)(q – 1)
5. Pilih sebuah bilangan bulat untuk kunci publik yang disebut e, yang relatif prima terhadap m. Relatif prima terhadap m artinya faktor pembagi keduanya adalah 1, secara matematis disebut gcd (e,m) = 1
6. Hitung kunci untuk dekripsi (d) dengan rumus e.d mod m = 1

Maka hasil dari algoritma diatas yaitu :

a. Kunci publik adalah pasangan (e, n)
b. Kunci privat adalah pasangan (d, n)

Catatan : n tidak bersifat rahasia, namun ia diperlukan pada perhitungan enkripsi atau dekripsi

## 2.4 Metode Noekeon

Algoritma Noekeon adalah salah satu jenis algoritma kriptografi blok cipher Algoritma noekeon ini termasuk keluarga dua cipher blok yang dirancang oleh Joan Daemen, Michael Peeters, Gilles Van Assche dan Vinjent Rijmen yang dibuat pada proyek Nessie pada September 2000. Dua cipher tersebut adalah *direct mode* dan *indirect mode*. Noekeon juga merupakan cipher blok berulang dengan panjang blok dan panjang kunci masing-masing 128 bit, terdiri dari transformasi round sederhana yang berulang, diikuti dengan transformasi output. Algoritma noekeon memiliki 16

putaran (Nr) iterasi, dalam setiap putarannya dilakukan empat buah transformasi yaitu theta, shift offset yang terdiri dua buah transformasi Pi1 dan Pi2 dan gamma (Milawati,, Silalahi, & Tampubolon, 2021).

1. Penjadwalan kunci

   Penjadwalan kunci dilakukan dengan cara mengkonversi kunci utama (*cipher key*) 128 bit menjadi sebuah *working-key* 128 bit. Karena sifat Noekeon yang simetri, maka setiap roundnya menggunakan *working-key* yang sama. Dalam Noekeon ada mode saat penjadwalan kunci tidak lakukan yang disebut dengan "*direct mode*" artinya working-key adalah *cipher-key* itu sendiri. Mode yang kedua mode "*indirect mode*" yang melakukan proses penjadwalan kunci untuk mengeliminasi pola serangan *related-key*.

   Pada *indirect-key*, sebelum kunci diaplikasikan terhadap pesan pada oprasi theta, kunci dirubah dahulu menjadi sebuah kunci yang lain dengan tetap menggunakan fungsi yang sama dalam Noekeon. Kemudian baru kunci tersebut diaplikasikan pada pesan pada oprasi theta dan seterusnya sebanyak 16 putaran.

   a. State

      Setiap informasi round dioprasikan pada sebuah state yang terdiri dari empat buah 35-bit word yaitu [0]

b. Theta

Theta adalah pemetaan linier yang menggunakan working-key k dan dilakukan pada oprasi state a. tahap ini memerlukan 12 langkah dalam penyelesaiannya. Adapun langkah tersebut yaitu:

Langkah pertama oprasi *xor* antara word a0 dan a2. Kemudian hasil oprasi dilakukan pergeseran bit, yaitu ke kanan 8-bit dan ke kiri 8-bit. Hasil pergeseran tersebut di-xor dengan hasil langkah pertama. Berikutnya yaitu proses perubahan word a1 dan a3, dengan meng-*xor*-kan a1 dengan langkah kedua. Setelah itu keempat *word* dari *plaintext* masing-masing di xor-kan dengan keempat buah kunci word dan akan menghasilkan *word* [a0,a1,a2,a3] baru. Dari *word* baru tersebut, a1 dan a3 di-*xor* dan hasilnya dilakukan dua buah pergeseran 8-bit masing-masing kekanan dan kekiri. Terakhir a0 dan a2 masing-masing akan di-*xor* dengan hasil pergeseran tersebut. Dari proses Theta ini akan dihasilkan word [a0,a1,a2,a3] yang baru untuk proses selanjutnya.

c. Shift Offset

Pergeseran pada tahap ini terdiri dari 2 kali pergeseran yaitu Pi1 dan Pi2 yang masing-masing berkebalikan arah dimana pergeseran pada Pi1 adalah:

Pi1: A0 tidak bergeser

A1 digeser 1 bit kekiri

A2 digeser 5 bit kekiri

A3 digeser 2 bit kekiri

Pi2: A0 tidak bergeser

A1 digeser 1 bit kekanan

A2 digeser 5 bit kekanan

A3 digeser 2 bit kekanan

d. Gamma

Gamma merupakan pemetaan non linier, dengan tiga langkah:

1. Transformasi non linear sederhana
2. Transformasi linier sederhana
3. Transformasi non linear sederhana

Dalam tahap ini Noekeon akan menghasilkan S-Box yang terdiri dari 4 buah word 32-bit (a0,a1,a2,a3).

e. Round Constant

Untuk menghilangkan sifat linear pada setiap putaran Noekeon, dilakukan operasi round constant yang merupakan *shift register* (mod 0x80, untuk state [0] yang dilakukan terhadap 8-bit terbawah dalam 32-bit word state awal.

2. Enkripsi dan Dekripsi

Dalam setiap algoritma kriptografi akan melakukan proses enkripsi dan dekripsi pada data yang akan diproses. Adapun di algoritma Noekeon proses enkripsi dan dekripsi tersebut denga langkah berikut:

a. Enkripsi

Tahap ini diawali dengan adanya masukan dari pengguna berupa teks dan kunci. Lalu teks tersebut diubah menjadi bi-bit dan bentuk blok sepanjang 128 bit. Yang masing-masing blok dank kunci dibagi menjadi 4 buah word 32 bit (a0,a1,a2,a3) untuk plaintext dan

(k0,k1,k2,k3) untuk kunci. Bila ternyata dalam suatu blok jumlah bitnya kurang dari 128 bit,maka akan dilakukan padding dengan menambahkan bit dummies.

b. Dekripsi

Proses selanjutnya yang dilakukan oleh Noekeon adalah dekripsi. Keunggulan algoritma Noekeon terletak pada kesederhanaan kode program dan sirkuit perangkat kerasnya. Kode atau sirkuit yang sama digunakan dalam enkripsi maupun dekripsinya, hanya penerapan pada theta yang berbeda. Pada enkripsi, theta adalah (k,a). Namun pada dekripsi, menjadi theta (NullVektor,a). Kebalikan dari theta adalah theta itu sendiri. Namun dengan pengaplikasian *null vector* sebagai *working-key*.

## 2.5 Tinjaun Umum Visual Basic.Net

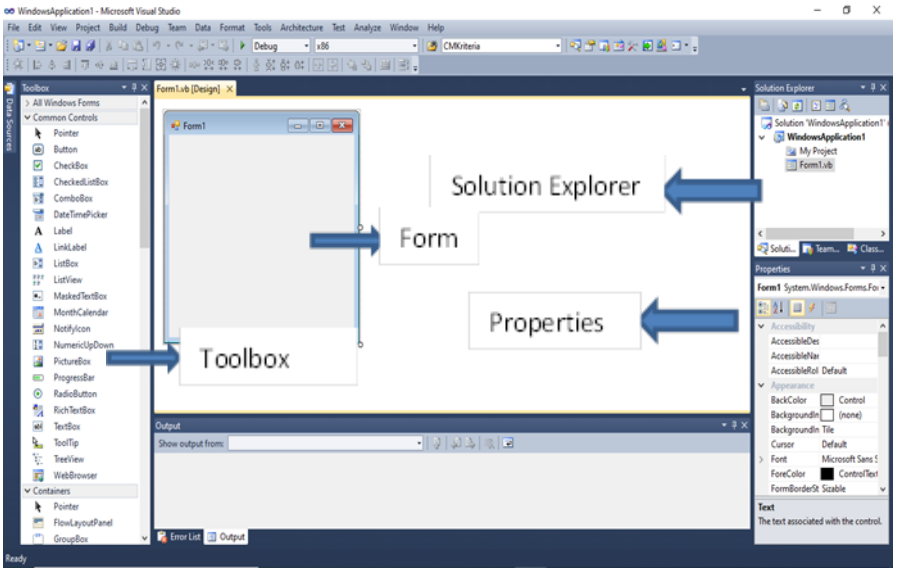
Microsoft Visual Basic.Net adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET Framework, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para programmer dapat membangun aplikasi *Windows Forms*, Aplikasi web berbasis ASP.NET, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++, Visual C#, atau Visual J#),* atau juga dapat diperoleh secara terpadu dalam Microsoft Visual Studio (R.H Sianipar, 2019).

Visual Basic.Net merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi *Windows*. Visual Basic merupakan salah satu

bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming* = OOP),

### 2.5.1 Menu Utama *Integrated Development Environment*

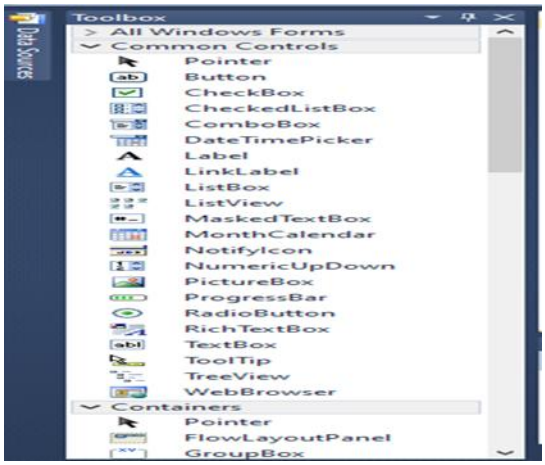
Di dalam menu utama IDE tersedia perintah-perintah dan disertai pula dengan submenu-submenunya. Pada umumnya menu juga dapat ditampilkan dalam bentuk toolbar, tetapi tidak semua opsi tersedia pada saat itu juga. Adakalanya opsi-opsi tersebut tidak dapat diterapkan pada tempat IDE. Ini berarti opsi tersebut dalam keadaan *invisible* atau *disabled*.




**Gambar 2.2 Komponen Visual Basic 2010**

### 2.5.2 *Toolbox Windows Form*

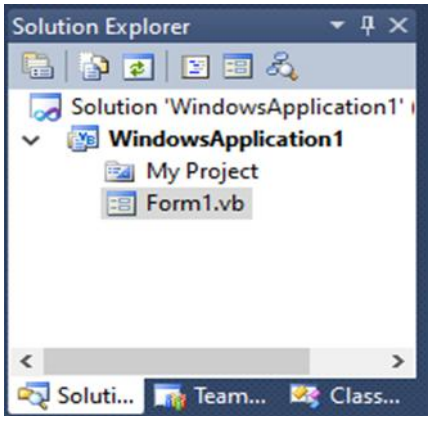
*Toolbox* berisi berbagai control yang dapat anda gunakan untuk mendesain antarmuka grafis. *Toolbox* mempunyai pengaturan automatic hiding sehingga akan tertutup jika tidak diperlukan.




**Gambar 2.3 *Toolbox***

### 2.5.3 Jendela *Explorer*

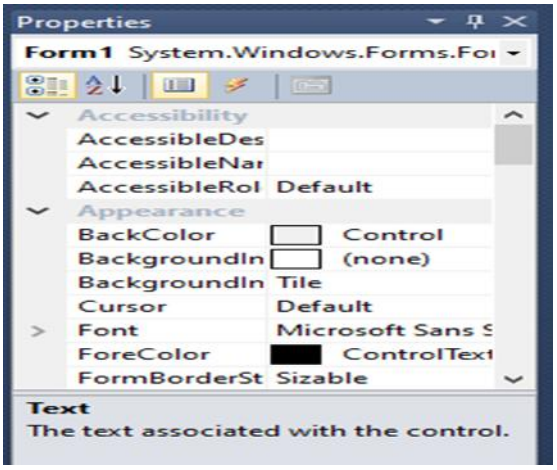
Jendela *explorer* merupakan tempat ditampilkannya daftar-daftar komponen secara hirarki. Dalam Jendela explorer dimungkinkan adanya beberapa proyek, dan dalam proyek ini masih ada beberapa item lagi seperti *form*, *module*, dan lain-lain.




**Gambar 2.4 Jendela *Explorer***

### 2.5.4 Jendela *Properties*

Jendela propertis ini berfungsi untuk menampilkan semua *property* dari komponen yang dipilih beserta settingannya. Dengan jendela ini kita dapat mengatur property dari masing-masing kontrol yang telah dibuat.




**Gambar 2.5 Jendela *Properties***

## 2.6 Pengertian UML (*Unified Modeling Language)*

*Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Samsudin & Islami, 2023).

UML merupakan singkatan dari *Unified Modeling Language* yang merupakan sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainya (Siagian & Triandi, 2024)

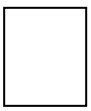
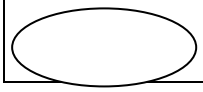
### 2.6.1 Use Case Diagram

*Use Case Diagram* merupakan gambaran *graphical* dari beberapa atau semua aktor, *Use Case*, dan interaksi yang memperkenalkan suatu sistem. *Use Case* secara sederhana sesungguhnya merupakan sebuah sarana bantu untuk mendefinisikan apa yang ada di luar sistem (aktor) dan apa yang harus dilakukan oleh sistem yang sedang dikembangkan (Siagian & Triandi, 2024).

Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna (Limantoro & Kristiadi, 2021). Use Case digram yaitu model hasil analisis perancangan sistem yang bertujuan untuk mendeskripsikan kebutuhan sistem. Kebutuhan sistem tersebut akan diterapkan oleh pengguna sehingga perancangan sistem dapat tergambarkan.

Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

**Tabel 2.1 Simbol Use Case Diagram**

| Gambar | Nama | Keterangan |
|---|---|---|
|  | *Actor* | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinterasi dengan *Use Case*. |
|  | *Depedency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
|  | *Generalization* | Hubungan dimana objek anak(*Descended)* berbagi prilaku dan struktur data dari objek yang diatasnya objek induk. |
|  | *Include* | Menspesifikasikan bahwa use case sumber secara explicit. |
|  | *Extend* | Menspesifikasikan bahwa use case target memperluas prilaku pada sebuah titik diberikan. |
|  | *Assosiation* | Apa yang memhubungkan objek satu dengan objek yang lainnya. |
|  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
|  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang |

| Gambar | Nama | Keterangan |
|---|---|---|
| | | menghasilkan suatu hasil yang terukur dari sebuah *actor*. |
|  | *Colaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya. |
|  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

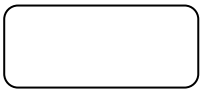
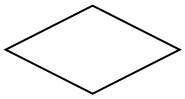
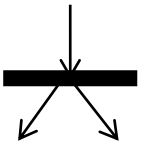
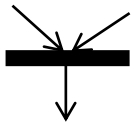
### 2.6.2 *Activity Diagram*

Diagram aktivitas menggambarkan fungsi yang direncanakan, cara kerja masing-masing fungsi, serta hasil yang diharapkan dari fungsi tersebut. Diagram Activity memodelkan peristiwa yang terjadi dalam *Use Case* (Hasan & Safrizal, 2024)

Diagram aktivitas (*Activity Diagram***)** adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem (Limantoro & Kristiadi, 2021), *Activity* diagram digunakan untuk mendokumentasikan alur kerja pada sebuah sistem,yang dimulai dari pandangan business level hingga ke operational level. Pada dasarnya, *activity* diagram merupakan variasi dari statechart diagram. Activity diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku

paralel sedangkan flowchart tidak bisa. Berikut adalah notasi *activity* diagram..

**Tabel 2.2 Simbol Activity Diagram**

| Simbol | Nama Simbol | Keterangan |
|---|---|---|
|  | Titik Awal | Status awal aktivitas sistem |
|  | Titik Akhir | Status akhir yang dilakukan oleh sistem |
|  | Activity | Aktivitas yang dilakukan oleh sistem, biasanya diawali dengan kata kerja |
|  | Decission (percabangan) | Pilihan untuk mengambil keputusan |
|  | Fork; | Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu. |
|  | Join | Digunakan untuk menunjukkan kegiatan yang digabungkan |

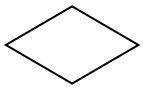
### 2.6.3 *Class Diagram*

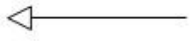
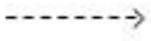
*Class diagram* ialah menjelaskan secara garis besar mengenai kelas-kelas perancangan sistem dari sudut pandang struktur sistem yang dapat memperjelas fungsi-fungsinya. attribute dan operasi merupakan bagian dari class diagram yang dapat memberi gambaran

hubungan antara perancangan dan perangkat lunaknya sehingga sesuai dengan pembuatan programnya (Sutrisno & Karnadi, 2021)

Diagram kelas adalah representasi yang menggambarkan struktur sistem, menjelaskan kelas-kelas yang ada, serta hubungan antara satu kelas dengan kelas lainnya. Class diagram menggambarkan model untuk merancang atribut dan fungsi yang dipakai untuk membuat sistem yang baru (Hasan & Safrizal, 2024)

***Tabel 2.3 Simbol Class Diagram***

| No | Simbol | Nama | Keterangan |
|---|---|---|---|
| 1 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek |
| 3 |  | *Assosiasi* | Hubungan statis antar *class*yang menggambarkan *class* yang memiliki atribut berupa class lain atau *class*yang harus mengetahui eksistensi *class* lain |

| No | Simbol | Nama | Keterangan |
|---|---|---|---|
| 4 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*) |
| 5 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independet) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri<br><br>(independent) |

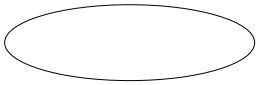
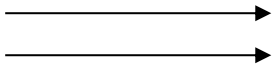
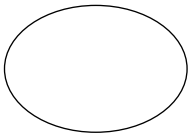
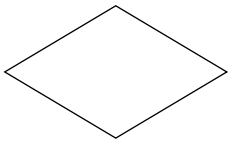
## 2.7 Flowchart

Bagan alir dokumen (*document flowchart*) adalah bagan (*chart*) yang menunjukkan aliran (*flow*) di dalam program atau prosedur sistem secara logika, digunakan terutama sebagai alat bantu komunikasi dan untuk dokumentasi. Bagan alir sistem (*system flowchart*) merupakan bagan yang menunjukkan arus pekerjaan dari sistem secara keseluruhan, menjelaskan urutan dari prosedur yang ada di dalam sistem serta menunjukkan apa yang dikerjakan di dalam sistem. (Muafi, Wijaya, & Aziz, 2020).

*Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta

pernyataannya, gambaran ini dinyatakan dengan simbol (Ayumida, Azis, & Fiano, 2020). Berikut ini mengenai beberapa fungsi flowchart :

**Tabel 2.4 Simbol dan Fungsi *Flowchart***

| SIMBOL | KETERANGAN |
|---|---|
|  | *Star*/Mulai<br>*End*/ Selesai |
|  | Simbol arus/ flow yang menyatakan jalannya proses |
|  | Simbol *connector*, (menyatakan sambungan dari proses ke proses lainnya dalam hal yang sama |
|  | Simbol *process* yaitu menyatakan suatu tindakan |
|  | Simbol manual, manyatakan suatu tindakan |
|  | Simbol *decision*, menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan |
|  | Simbol *keying* operation menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai *keyboard* |
|  | Simbol *input*/*output* menyatakan proses input/output |

# BAB III

# METODOLOGI PENELITIAN

## 3.1 Waktu dan Tempat Penelitian

Waktu penelitian dimulai dari bulan Oktober 2024 sampai dengan bulan Mei 2025. Penelitian dilakukan secara mandiri dengan melakukan uji coba terhadap file dan menganalisa dari penerepan algoritma yang digunakan.

## 3.2 Metode Penelitian

Dalam melakukan penelitian ini, penulis menggunakan metode terapan (*applied research*). Penelitian terapan atau *applied research* dilakukan berkenaan dengan kenyataan-kenyataan praktis, penerapan, dan pengembangan ilmu pengetahuan yang dihasilkan oleh penelitian dasar dalam kehidupan nyata. Penelitian terapan berfungsi untuk mencari solusi tentang masalah-masalah tertentu. Tujuan utama penelitian terapan adalah pemecahan masalah sehingga hasil penelitian dapat dimanfaatkan untuk kepentingan manusia baik secara individu atau kelompok maupun untuk keperluan industri atau politik dan bukan untuk wawasan keilmuan semata.

Dalam melaksanakan penelitian terapan ini terdapat 5(lima) langkah, diantaranya :

a. Melakukan sesuatu yang sedang diperlukan, dipelajari, diukur, dan diperiksa kelemahannya.
b. Mencari satu dari kelemahan-kelemahan yang diperoleh dipilih untuk penelitian.
c. Mencari dan memberikan solusi dalam melakukan pemecahan masalah

d. Kemudian dilakukan modifikasi sehingga penyelesaian dapat dilakukan untuk diterapkan.

e. Pemecahan dipertahankan dan menempatkannya dalam suatu kesatuan sehingga jadi bagian permanen dalam satu sistem.

## 3.3 Perangkat Lunak dan Perangkat Keras

### 3.3.1 Perangkat lunak (*Software*)

a. Sistem operasi Windows 10

b. Bahasa pemrograman Visual Studio.Net 2010

c. Microsoft Visio

### 3.3.2 Perangkat Keras *( (Hardware*)

a. Laptop Dell Intel Core i3

b. SSD 512

c. Processor Intel Corei3

d. Printer Epson

## 3.4 Metode Pengumpulan Data

Metode pengumpulan data digunakan untuk memperoleh data yang dapat mendukung permasalahan yang akan dibahas. Sehubungan dengan hal ini maka digunakan metode pengumpulan data yang meliputi :

A. Observasi

Dalam pengumpulan data melalui observasi, penulis mengamati dan menganalisa bagaimana cara sistem melakukan enkripsi dan dekripsi yang berbentuk data.

B. Studi Pustaka

Studi pustaka adalah suatu metode pengumpulan data yang diambil dari perpustakaan yang berupa karya ilmiah, jurnal, buku-buku serta dari *internet* yang berhubungan dengan penulisan ini. Tujuan dari studi pustaka ini adalah untuk mendalami dan memperoleh keterangan yang lengkap terhadap objek yang diteliti.

## 3.5 Analisa Perancangan Sistem

Analisa dan perancangan pada sistem merupakan sebuah bentuk persiapan yang dilakukan sebelum proses perancangan sistem dan implementasi algoritma dilakukan. Analisis sendiri dilakukan dengan cara yang berbeda-beda namun memiliki kesamaan pada persiapan kebutuhan sistem..

### 3.5.1 Analisa Sistem Aktual

Pada penelitian ini akan dilakukan analisis terhadap arsitektur sistem dan metode enkripsi dan dekripsi pada pesan teks yang digunakan pada aplikasi yang dibangun yaitu algoritma RSA dan Noekeon. Analisis akan dimulai dengan analisis kebutuhan atau spesifikasi dari sistem sehingga berdasarkan spesifikasi tersebut dapat dibangun rancangan dari aristektur sistem yang akan dibangun.

Analisis yang akan dilaksanakan akan didasarkan pada tujuan utama yaitu untuk menambahkan kekuatan pengamanan keamanan data dengam menggabungkan dua algoritma yang memiliki perbedaan konsep kriptografi yaitu RSA dan Noekeon. Dengan memanfaatkan karakteristik dari masing-masing algoritma dimana RSA dengan

kunci public dan kunci privatenya dan Noekeon dengan blok chipernya, yang mana kedua algoritma ini akan digabungkan dalam satu buah sistem yang akan mengamankan data teks. Setelah proses analisa kebutuhan sistem maka dapat diambil beberapa kebutuhan inti dari sistem seperti berikut:

1. Sistem yang akan dibangun oleh penulis adalah sistem yang didasarkan pada platform VB.Net untuk kemudahan dalam penggunaaan sistem
2. Sistem yang dibuat harus mampu merubah data sehinga tidak lagi merepresentasikan bentuk awal dan tidak mampu dibaca secara langsung, dan kemudian sistem juga harus mampu mengembalikan bentuk data yang sudah diacak tadi kembali menjadi teks yang dapat dibaca

### 3.5.2 Analisa Sistem Baru

Analisa sistem baru yang akan dilakukan pada penelitian ini terbagi menjadi analisa metode RSA dalam keamanan pesan teks, perancangan proses, pembuatan flowchart operasional dan sistem, perancangan UML sistem, dan perancangan bentuk interface sistem. Berikut penjabaran dari masing–masing perancangan yang dilakukan pada penelitian ini

**A. Analisis Algoritma RSA dan Noekeon**

RSA merupakan algoritma kriptografi yang menggunakan dua kunci berbeda pada proses enkripsi dan dekripsi-nya. RSA menganut sistem algoritma kunci publik yang saat ini telah

digunakan secara luas. RSA pertama kali dipublikasikan pada tahun 1977. RSA merupakan metode kriptografi asimetris yang beroperasi pada mode blok. RSA membutuhkan dua kunci yang berbeda pada proses enkripsi dan dekripsi nya sehingga proses enkripsi dan dekripsi hanya dapat dilakukan oleh pihak yang memiliki kunci yang sesuai. Sedangkan metode Noekeon merupakan *cipher blok* berulang dengan panjang blok dan panjang kuncinya masing-masing 128 bit, yang terdiri dari aplikasi transformasi *round* sederhana yang berulang, diikuti dengan sebuah transformasi output. Noekeon memiliki 16 putaran (Nr) iterasi, dalam setiap putarannya dilakukan empat buah transformasi yaitu theta, *Shift offset* yang terdiri dari dua buah transformasi Pi1 dan Pi2, dan gamma

**Contoh Kasus :**

Diketahui kunci :
Publik : 5-161
Private : 53-161
Plainteks : UNHAR
Pembangkitan Kunci RSA :
Memilih p bilangan prima secara acak : 29
Memilih q bilangan prima secara acak : 3
Menghitung n = (p*q) = 87
Menghitung m = (p-1) * (q-1) = 56
Menghitung nilai e yang memenuhi GCD(e,m) atau pembagi sama yang terbesar = 1
e = 2, GCD(2,56) = 2 tidak memenuhi
e = 3, GCD(3,56) = 1 memenuhi
Menghitung nilai d yang memenuhi (d x e) mod m = 1

d = 2, (2*3) mod 56 = 6 tidak memenuhi

d = 3, (3*3) mod 56 = 9 tidak memenuhi

d = 4, (4*3) mod 56 = 12 tidak memenuhi

d = 5, (5*3) mod 56 = 15 tidak memenuhi

d = 6, (6*3) mod 56 = 18 tidak memenuhi

d = 7, (7*3) mod 56 = 21 tidak memenuhi

d = 8, (8*3) mod 56 = 24 tidak memenuhi

d = 9, (9*3) mod 56 = 27 tidak memenuhi

d = 10, (10*3) mod 56 = 30 tidak memenuhi

d = 11, (11*3) mod 56 = 33 tidak memenuhi

d = 12, (12*3) mod 56 = 36 tidak memenuhi

d = 13, (13*3) mod 56 = 39 tidak memenuhi

d = 14, (14*3) mod 56 = 42 tidak memenuhi

d = 15, (15*3) mod 56 = 45 tidak memenuhi

d = 16, (16*3) mod 56 = 48 tidak memenuhi

d = 17, (17*3) mod 56 = 51 tidak memenuhi

d = 18, (18*3) mod 56 = 54 tidak memenuhi

d = 19, (19*3) mod 56 = 1 memenuhi

diperoleh

kunci public (e,n) = (3-87)

kunci private (d,n) = (19-87)

Plainteks : UNHAR

Enkripsi RSA :

Enkripsi RSA :

Ch[0] = 85 ^ 3 Mod 87 = 79

Ch[1] = 78 ^ 3 Mod 87 = 54

Ch[2] = 72 ^ 3 Mod 87 = 18

Ch[3] = 65 ^ 3 Mod 87 = 53

Ch[4] = 82 ^ 3 Mod 87 = 49

Hasil Enkripsi RSA = 79#54#18#53#49

Penjadwalan Kunci Noekeon :

RC[0] = 128

RC[1] = 26

RC[2] = 52

RC[3] = 104

RC[4] = 208

RC[5] = 186

RC[6] = 110

RC[7] = 220

RC[8] = 162

RC[9] = 94

RC[10] = 188

RC[11] = 98

RC[12] = 196

RC[13] = 146

RC[14] = 62

RC[15] = 124

RC[16] = 248

248Penjadwalan Kunci :

Kunci Input : 8700000000000000

Kunci(Bit) : 00111000;00110111;00110000;00110000;
00110000;00110000;00110000;00110000;
00110000;00110000;00110000;00110000;
00110000;00110000;00110000;00110000;

A0 : 00111000;00110111;00110000;00110000 = 943140912

A1 : 00110000;00110000;00110000;00110000 = 808464432

A2 : 00110000;00110000;00110000;00110000 = 808464432

A3 : 00110000;00110000;00110000;00110000 = 808464432

**Putaran (0)Penjadwalan Kunci :**

A0 = A0 Xor RC(0) = 943141040

Theta Inv Proses :

A0 : 943141040

A1 : 808464432

A2 : 808464432

A3 : 808464432

T = A0 Xor A2

= 134676608

Temp = T<<8 Xor T>>8 = 2265483016

A1 = T Xor Temp Xor A1 = 3208624056

A3 = T Xor Temp Xor A3 = 3208624056

T = A1 Xor A3 = 0

Temp = T<<8 Xor T>>8 = 0

A0 = T Xor Temp Xor A0 = 943141040

A2 = T Xor Temp Xor A2 = 808464432

A1 = A1 << 1 = 2122280817

A2 = A2 << 5 = 101058054

A3 = A3 << 2 = 4244561634

Gamma Proses :

A0 : 943141040

A1 : 2122280817

A2 : 101058054

A3 : 4244561634

A1 = A1 Xor ((Not A3) And (Not A2)) = 2138984040

A0 = A0 Xor (A2 And A1) = 1043412656

A0 = A3 = 4244561634

A3 = A0 = 1043412656

A2 = A2 Xor A3 Xor A1 Xor A0 = 3149373500

A1 = A1 Xor (Not A3 And Not A2) = 1060505387

A0 = A0 Xor (A2 And A1) = 3351830218

A1 = A1 >> 1 = 2677736341

A2 = A2 >> 5 = 3856514305

A3 = A3 >> 2 = 260853164

**Putaran (1)Penjadwalan Kunci :**

A0 = A0 Xor RC(1) = 3351830224

Theta Inv Proses :

A0 : 3351830224

A1 : 2677736341

A2 : 3856514305

A3 : 260853164

T = A0 Xor A2

= 571827153

Temp = T<<8 Xor T>>8 = 3292644417

A1 = T Xor Temp Xor A1 = 2043651077

A3 = T Xor Temp Xor A3 = 3923307068

T = A1 Xor A3 = 2417446457

Temp = T<<8 Xor T>>8 = 786312926

A0 = T Xor Temp Xor A0 = 2030157367

A2 = T Xor Temp Xor A2 = 1528094182

A1 = A1 << 1 = 4087302154

A2 = A2 << 5 = 1654373579

A3 = A3 << 2 = 2808326387

Gamma Proses :

A0 : 2030157367

A1 : 4087302154

A2 : 1654373579

A3 : 2808326387

A1 = A1 Xor ((Not A3) And (Not A2)) = 3952806670

A0 = A0 Xor (A2 And A1) = 463124029

A0 = A3 = 2808326387

A3 = A0 = 463124029

A2 = A2 Xor A3 Xor A1 Xor A0 = 905554187

A1 = A1 Xor (Not A3 And Not A2) = 731859918

A0 = A0 Xor (A2 And A1) = 2264572409

A1 = A1 >> 1 = 365929959

A2 = A2 >> 5 = 1504693576

A3 = A3 >> 2 = 1189522831

**Putaran (2)Penjadwalan Kunci :**

A0 = A0 Xor RC(2) = 2264572365

Theta Inv Proses :

A0 : 2264572365

A1 : 365929959

A2 : 1504693576

A3 : 1189522831

T = A0 Xor A2

= 3746917509

Temp = T<<8 Xor T>>8 = 3501445299

A1 = T Xor Temp Xor A1 = 438901201

A3 = T Xor Temp Xor A3 = 1224741305

T = A1 Xor A3 = 1395198056

Temp = T<<8 Xor T>>8 = 1096499547

A0 = T Xor Temp Xor A0 = 2492000510

A2 = T Xor Temp Xor A2 = 1272808571

A1 = A1 << 1 = 877802402

A2 = A2 << 5 = 2075168617

A3 = A3 << 2 = 603997925

Gamma Proses :

A0 : 2492000510

A1 : 877802402

A2 : 2075168617

A3 : 603997925

A1 = A1 Xor ((Not A3) And (Not A2)) = 3021800368

A0 = A0 Xor (A2 And A1) = 2761485278

A0 = A3 = 603997925

A3 = A0 = 2761485278

A2 = A2 Xor A3 Xor A1 Xor A0 = 1328882146

A1 = A1 Xor (Not A3 And Not A2) = 2757695409

A0 = A0 Xor (A2 And A1) = 538265413

A1 = A1 >> 1 = 3526331352

A2 = A2 >> 5 = 309963023

A3 = A3 >> 2 = 2837854967

**Putaran (15)Penjadwalan Kunci :**

A0 = A0 Xor RC(15) = 2391970765

Theta Inv Proses :

A0 : 2391970765

A1 : 3733178967

A2 : 559874642

A3 : 2670652982

T = A0 Xor A2

= 2949484959

Temp = T<<8 Xor T>>8 = 1379815998

A1 = T Xor Temp Xor A1 = 594545142

A3 = T Xor Temp Xor A3 = 1658662295

T = A1 Xor A3 = 1101866081

Temp = T<<8 Xor T>>8 = 3429223525

A0 = T Xor Temp Xor A0 = 56261577

A2 = T Xor Temp Xor A2 = 2895637078

A1 = A1 << 1 = 1189090284

A2 = A2 << 5 = 2466073301

A3 = A3 << 2 = 2339681885

Gamma Proses :

A0 : 56261577

A1 : 1189090284

A2 : 2466073301

A3 : 2339681885

A1 = A1 Xor ((Not A3) And (Not A2)) = 585241294

A0 = A0 Xor (A2 And A1) = 28997901

A0 = A3 = 2339681885

A3 = A0 = 28997901

A2 = A2 Xor A3 Xor A1 Xor A0 = 986814283

A1 = A1 Xor (Not A3 And Not A2) = 3873837694

A0 = A0 Xor (A2 And A1) = 2847187991

A1 = A1 >> 1 = 1936918847

A2 = A2 >> 5 = 1507232954

A3 = A3 >> 2 = 1080991299

A0 = A0 Xor RC(16) = 2847188207

DK(0) = A0 = 2847188207

DK(1) = A1 = 1936918847

DK(2) = A2 = 1507232954

DK(3) = A3 = 1080991299

Theta Inv Proses :

A0 : 2847188207

A1 : 1936918847

A2 : 1507232954

A3 : 1080991299

T = A0 Xor A2

= 4032964693

Temp = T<<8 Xor T>>8 = 936916952

A1 = T Xor Temp Xor A1 = 3033077426

A3 = T Xor Temp Xor A3 = 2278851022

T = A1 Xor A3 = 857577340

Temp = T<<8 Xor T>>8 = 1638162852

A0 = T Xor Temp Xor A0 = 4211954231

A2 = T Xor Temp Xor A2 = 191855202

EK(0) = A0 = 4211954231

EK(1) = A1 = 3033077426

EK(2) = A2 = 191855202

EK(3) = A3 = 2278851022

Pembentukan Blok Input :

Blok Input (0) : 00110111;00111001;00100011;00110101

Blok Input (1) : 00110100;00100011;00110001;00111000

Blok Input (2) : 00100011;00110101;00110011;00100011

Blok Input (3) : 00110100;00111001;00100000;00100000

End (4) BLok

**Enkripsi :**

A0 : 926491445

A1 : 874721592

A2 : 590689059

A3 : 876159008

**Putaran (0) Enkripsi :**

A0 = A0 Xor RC(0) = 926491573

Theta Proses :

A0 : 926491573

A1 : 874721592

A2 : 590689059

A3 : 876159008

K(0) : 4211954231

K(1) : 3033077426

K(2) : 191855202

K(3) : 2278851022

T = A0 Xor A2

= 336334998

Temp = T<<8 Xor T>>8 = 2583992836

A1 = T Xor Temp Xor A1 = 3123428266

A3 = T Xor Temp Xor A3 = 3123817138

A0 = A0 Xor Kunci(0) = 3425988994

A1 = A1 Xor Kunci(1) = 249736472

A2 = A2 Xor Kunci(2) = 677005633

A3 = A3 Xor Kunci(3) = 1038429052

T = A1 Xor A3 = 856131172

Temp = T<<8 Xor T>>8 = 1672831925

A0 = T Xor Temp Xor A0 = 2626065491

A2 = T Xor Temp Xor A2 = 2028514448

A1 = A1 << 1 = 499472944

A2 = A2 << 5 = 487952911

A3 = A3 << 2 = 4153716208

Gamma Proses :

A0 : 2626065491

A1 : 499472944

A2 : 487952911

A3 : 4153716208

A1 = A1 Xor ((Not A3) And (Not A2)) = 498014768

A0 = A0 Xor (A2 And A1) = 2172880467

A0 = A3 = 4153716208

A3 = A0 = 2172880467

A2 = A2 Xor A3 Xor A1 Xor A0 = 1991091100

A1 = A1 Xor (Not A3 And Not A2) = 369050128

A0 = A0 Xor (A2 And A1) = 3812208608

A1 = A1 >> 1 = 184525064

A2 = A2 >> 5 = 3820317980

A3 = A3 >> 2 = 3764445588

**Putaran (1) Enkripsi :**

A0 = A0 Xor RC(1) = 3812208634

Theta Proses :

A0 : 3812208634

A1 : 184525064

A2 : 3820317980

A3 : 3764445588

K(0) : 4211954231

K(1) : 3033077426

K(2) : 191855202

K(3) : 2278851022

T = A0 Xor A2

= 9224934

Temp = T<<8 Xor T>>8 = 1791126210

A1 = T Xor Temp Xor A1 = 1622214956

A3 = T Xor Temp Xor A3 = 2318289328

A0 = A0 Xor Kunci(0) = 406126029

A1 = A1 Xor Kunci(1) = 3564642206

A2 = A2 Xor Kunci(2) = 3906606974

A3 = A3 Xor Kunci(3) = 234539134

T = A1 Xor A3 = 3649230816

Temp = T<<8 Xor T>>8 = 1645109774

A0 = T Xor Temp Xor A0 = 2746763299

A2 = T Xor Temp Xor A2 = 1398186640

A1 = A1 << 1 = 2834317117

A2 = A2 << 5 = 1792299530

A3 = A3 << 2 = 938156536

Gamma Proses :

A0 : 2746763299

A1 : 2834317117

A2 : 1792299530

A3 : 938156536

A1 = A1 Xor ((Not A3) And (Not A2)) = 686863160

A0 = A0 Xor (A2 And A1) = 2338871851

A0 = A3 = 938156536

A3 = A0 = 2338871851

A2 = A2 Xor A3 Xor A1 Xor A0 = 4272397025

A1 = A1 Xor (Not A3 And Not A2) = 685807148

A0 = A0 Xor (A2 And A1) = 525054936

A1 = A1 >> 1 = 342903574

A2 = A2 >> 5 = 267730135

A3 = A3 >> 2 = 3805943434

**Putaran (2) Enkripsi :**

A0 = A0 Xor RC(2) = 525054956

Theta Proses :

A0 : 525054956

A1 : 342903574

A2 : 267730135

A3 : 3805943434

K(0) : 4211954231

K(1) : 3033077426

K(2) : 191855202

K(3) : 2278851022

T = A0 Xor A2

= 280923963

Temp = T<<8 Xor T>>8 = 2241824159

A1 = T Xor Temp Xor A1 = 2169586098

A3 = T Xor Temp Xor A3 = 2012945454

A0 = A0 Xor Kunci(0) = 3829850587

A1 = A1 Xor Kunci(1) = 899176192

A2 = A2 Xor Kunci(2) = 77219509

A3 = A3 Xor Kunci(3) = 4029652448

T = A1 Xor A3 = 3317156576

Temp = T<<8 Xor T>>8 = 1460623119

A0 = T Xor Temp Xor A0 = 1996389428

A2 = T Xor Temp Xor A2 = 2518866778

A1 = A1 << 1 = 1798352384

A2 = A2 << 5 = 3294325586

A3 = A3 << 2 = 3233707907

Gamma Proses :

A0 : 1996389428

A1 : 1798352384

A2 : 3294325586

A3 : 3233707907

A1 = A1 Xor ((Not A3) And (Not A2)) = 1345334828

A0 = A0 Xor (A2 And A1) = 921589300

A0 = A3 = 3233707907

A3 = A0 = 921589300

A2 = A2 Xor A3 Xor A1 Xor A0 = 1648059593

A1 = A1 Xor (Not A3 And Not A2) = 3643844398

A0 = A0 Xor (A2 And A1) = 2156811147

A1 = A1 >> 1 = 1821922199

A2 = A2 >> 5 = 1259461414

A3 = A3 >> 2 = 230397325

**Putaran (15) Enkripsi :**

A0 = A0 Xor RC(15) = 3744102091

Theta Proses :

A0 : 3744102091

A1 : 3691262960

A2 : 2818282888

A3 : 4250518133

K(0) : 4211954231

K(1) : 3033077426

K(2) : 191855202

K(3) : 2278851022

T = A0 Xor A2

= 2027021123

Temp = T<<8 Xor T>>8 = 2459669147

A1 = T Xor Temp Xor A1 = 911098408

A3 = T Xor Temp Xor A3 = 387167149

A0 = A0 Xor Kunci(0) = 606545148

A1 = A1 Xor Kunci(1) = 2189907098

A2 = A2 Xor Kunci(2) = 2895441898

A3 = A3 Xor Kunci(3) = 2428973667

T = A1 Xor A3 = 306210553

Temp = T<<8 Xor T>>8 = 3111434612

A0 = T Xor Temp Xor A0 = 2400451441

A2 = T Xor Temp Xor A2 = 127938663

A1 = A1 << 1 = 84846901

A2 = A2 << 5 = 4094037216

A3 = A3 << 2 = 1125960078

Gamma Proses :

A0 : 2400451441

A1 : 84846901

A2 : 4094037216

A3 : 1125960078

A1 = A1 Xor ((Not A3) And (Not A2)) = 233806628

A0 = A0 Xor (A2 And A1) = 2333471569

A0 = A3 = 1125960078

A3 = A0 = 2333471569

A2 = A2 Xor A3 Xor A1 Xor A0 = 836807963

A1 = A1 Xor (Not A3 And Not A2) = 1239784320

A0 = A0 Xor (A2 And A1) = 1123827854

A1 = A1 >> 1 = 619892160

A2 = A2 >> 5 = 3650028904

A3 = A3 >> 2 = 1657109716

A0 = A0 Xor RC(16) = 1123827830

Theta Proses :

A0 : 1123827830

A1 : 619892160

A2 : 3650028904

A3 : 1657109716

K(0) : 4211954231

K(1) : 3033077426

K(2) : 191855202

K(3) : 2278851022

T = A0 Xor A2

= 2608022814

Temp = T<<8 Xor T>>8 = 1843293662

A1 = T Xor Temp Xor A1 = 3529499904

A3 = T Xor Temp Xor A3 = 2489865236

A0 = A0 Xor Kunci(0) = 3119583809

A1 = A1 Xor Kunci(1) = 1721168818

A2 = A2 Xor Kunci(2) = 3537927946

A3 = A3 Xor Kunci(3) = 331142618

T = A1 Xor A3 = 1965696616

Temp = T<<8 Xor T>>8 = 1112752723

A0 = T Xor Temp Xor A0 = 2391307898

A2 = T Xor Temp Xor A2 = 3852016433

Store Output :

Output(0) : 142

Output(1) : 136

Output(2) : 118

Output(3) : 122

Store Output :

Output(4) : 102

Output(5) : 150

Output(6) : 243

Output(7) : 178

Store Output :

Output(8) : 229

Output(9) : 153

Output(10) : 27

Output(11) : 49

Store Output :

Output(12) : 19

Output(13) : 188

Output(14) : 213

Output(15) : 218

Hasil Enkripsi Dalam karakter ASCII :

**Žˆvzf–ó²å™_1_¼ÕÚ**

Dekripsi RSA Noekeon :

Kunci Dekripsi Noekeon : 87

Dekripsi Noekeon :

Chipertext :

**Žˆvzf–ó²å™_1_¼ÕÚ**

Pembentukan Blok Input :

Blok Input (0) : 10001110;10001000;01110110;01111010

Blok Input (1) : 01100110;10010110;11110011;10110010

Blok Input (2) : 11100101;10011001;00011011;00110001

Blok Input (3) : 00010011;10111100;11010101;11011010

End (4) BLok

A0 : 2391307898

A1 : 1721168818

A2 : 3852016433

A3 : 331142618

**Putaran (16) Dekripsi :**

Theta Proses :

A0 : 2391307898

A1 : 1721168818

A2 : 3852016433

A3 : 331142618

K(0) : 2847188207

K(1) : 1936918847

K(2) : 1507232954

K(3) : 1080991299

T = A0 Xor A2

= 1796304203

Temp = T<<8 Xor T>>8 = 1510365702

A1 = T Xor Temp Xor A1 = 1468122367

A3 = T Xor Temp Xor A3 = 581690007

A0 = A0 Xor Kunci(0) = 658297493

A1 = A1 Xor Kunci(1) = 619892160

A2 = A2 Xor Kunci(2) = 3159332747

A3 = A3 Xor Kunci(3) = 1657109716

T = A1 Xor A3 = 1178054932

Temp = T<<8 Xor T>>8 = 603399159

A0 = T Xor Temp Xor A0 = 1123827830

A2 = T Xor Temp Xor A2 = 3650028904

A0 = A0 Xor RC(16) = 1123827854

A1 = A1 << 1 = 1239784320

A2 = A2 << 5 = 836807963

A3 = A3 << 2 = 2333471569

Gamma Proses :

A0 : 1123827854

A1 : 1239784320

A2 : 836807963

A3 : 2333471569

A1 = A1 Xor ((Not A3) And (Not A2)) = 233806628

A0 = A0 Xor (A2 And A1) = 1125960078

A0 = A3 = 2333471569

A3 = A0 = 1125960078

A2 = A2 Xor A3 Xor A1 Xor A0 = 4094037216

A1 = A1 Xor (Not A3 And Not A2) = 84846901

A0 = A0 Xor (A2 And A1) = 2400451441

A1 = A1 >> 1 = 2189907098

A2 = A2 >> 5 = 127938663

A3 = A3 >> 2 = 2428973667

**Putaran (15) Dekripsi :**

Theta Proses :

A0 : 2400451441

A1 : 2189907098

A2 : 127938663

A3 : 2428973667

K(0) : 2847188207

K(1) : 1936918847

K(2) : 1507232954

K(3) : 1080991299

T = A0 Xor A2

= 2293484310

Temp = T<<8 Xor T>>8 = 2772673859

A1 = T Xor Temp Xor A1 = 2943826639

A3 = T Xor Temp Xor A3 = 3174521910

A0 = A0 Xor Kunci(0) = 648503198

A1 = A1 Xor Kunci(1) = 3691262960

A2 = A2 Xor Kunci(2) = 1584839901

A3 = A3 Xor Kunci(3) = 4250518133

T = A1 Xor A3 = 559804805

Temp = T<<8 Xor T>>8 = 3637565648

A0 = T Xor Temp Xor A0 = 3744102091

A2 = T Xor Temp Xor A2 = 2818282888

A0 = A0 Xor RC(15) = 3744102071

A1 = A1 << 1 = 3087558625

A2 = A2 << 5 = 4285706516

A3 = A3 << 2 = 4117170647

Gamma Proses :

A0 : 3744102071

A1 : 3087558625

A2 : 4285706516

A3 : 4117170647

A1 = A1 Xor ((Not A3) And (Not A2)) = 3095405001

A0 = A0 Xor (A2 And A1) = 1730828215

A0 = A3 = 4117170647

A3 = A0 = 1730828215

A2 = A2 Xor A3 Xor A1 Xor A0 = 3586117309

A1 = A1 Xor (Not A3 And Not A2) = 2965373321

A0 = A0 Xor (A2 And A1) = 1709640030

A1 = A1 >> 1 = 3630170308

A2 = A2 >> 5 = 4004380277

A3 = A3 >> 2 = 3653932525

**Putaran (14) Dekripsi :**

Theta Proses :

A0 : 1709640030

A1 : 3630170308

A2 : 4004380277

A3 : 3653932525

K(0) : 2847188207

K(1) : 1936918847

K(2) : 1507232954

K(3) : 1080991299

T = A0 Xor A2

= 2336945963

Temp = T<<8 Xor T>>8 = 1635541372

A1 = T Xor Temp Xor A1 = 844535443

A3 = T Xor Temp Xor A3 = 872154042

A0 = A0 Xor Kunci(0) = 3428036017

A1 = A1 Xor Kunci(1) = 1092984748

A2 = A2 Xor Kunci(2) = 3078320847

A3 = A3 Xor Kunci(3) = 1938988537

T = A1 Xor A3 = 850854485

Temp = T<<8 Xor T>>8 = 3794854448

A0 = T Xor Temp Xor A0 = 483675604

A2 = T Xor Temp Xor A2 = 1744605866

A0 = A0 Xor RC(14) = 483675626

A1 = A1 << 1 = 2185969496

A2 = A2 << 5 = 4287780172

A3 = A3 << 2 = 3460986853

Gamma Proses :

A0 : 483675626

A1 : 2185969496

A2 : 4287780172

A3 : 3460986853

A1 = A1 Xor ((Not A3) And (Not A2)) = 2188293962

A0 = A0 Xor (A2 And A1) = 2664847522

A0 = A3 = 3460986853

A3 = A0 = 2664847522

A2 = A2 Xor A3 Xor A1 Xor A0 = 761318721

A1 = A1 Xor (Not A3 And Not A2) = 3261568342

A0 = A0 Xor (A2 And A1) = 3458921125

A1 = A1 >> 1 = 1630784171

A2 = A2 >> 5 = 158008938

A3 = A3 >> 2 = 2813695528

**Putaran (1) Dekripsi :**

Theta Proses :

A0 : 2746763299

A1 : 3564642206

A2 : 1398186640

A3 : 234539134

K(0) : 2847188207

K(1) : 1936918847

K(2) : 1507232954

K(3) : 1080991299

T = A0 Xor A2

= 4042189491

Temp = T<<8 Xor T>>8 = 1562008858

A1 = T Xor Temp Xor A1 = 2039261239

A3 = T Xor Temp Xor A3 = 2685304791

A0 = A0 Xor Kunci(0) = 168619212

A1 = A1 Xor Kunci(1) = 184525064

A2 = A2 Xor Kunci(2) = 176172586

A3 = A3 Xor Kunci(3) = 3764445588

T = A1 Xor A3 = 3936305308

Temp = T<<8 Xor T>>8 = 61473706

A0 = T Xor Temp Xor A0 = 3812208634

A2 = T Xor Temp Xor A2 = 3820317980

A0 = A0 Xor RC(1) = 3812208608

A1 = A1 << 1 = 369050128

A2 = A2 << 5 = 1991091100

A3 = A3 << 2 = 2172880467

Gamma Proses :

A0 : 3812208608

A1 : 369050128

A2 : 1991091100

A3 : 2172880467

A1 = A1 Xor ((Not A3) And (Not A2)) = 498014768

A0 = A0 Xor (A2 And A1) = 4153716208

A0 = A3 = 2172880467

A3 = A0 = 4153716208

A2 = A2 Xor A3 Xor A1 Xor A0 = 487952911

A1 = A1 Xor (Not A3 And Not A2) = 499472944

A0 = A0 Xor (A2 And A1) = 2626065491

A1 = A1 >> 1 = 249736472

A2 = A2 >> 5 = 2028514448

A3 = A3 >> 2 = 1038429052

Theta Proses :

A0 : 2626065491

A1 : 249736472

A2 : 2028514448

A3 : 1038429052

K(0) : 2847188207

K(1) : 1936918847

K(2) : 1507232954

K(3) : 1080991299

T = A0 Xor A2

= 3832428739

Temp = T<<8 Xor T>>8 = 2916920796

A1 = T Xor Temp Xor A1 = 1196439559

A3 = T Xor Temp Xor A3 = 1951907427

A0 = A0 Xor Kunci(0) = 892481724

A1 = A1 Xor Kunci(1) = 874721592

A2 = A2 Xor Kunci(2) = 557719594

A3 = A3 Xor Kunci(3) = 876159008

T = A1 Xor A3 = 1708312

Temp = T<<8 Xor T>>8 = 34669073

A0 = T Xor Temp Xor A0 = 926491573

A2 = T Xor Temp Xor A2 = 590689059

A0 = A0 Xor RC(0) = 926491445

Store Output :

Output(0) : 55

Output(1) : 57

Output(2) : 35

Output(3) : 53

Store Output :

Output(4) : 52

Output(5) : 35

Output(6) : 49

Output(7) : 56

Store Output :

Output(8) : 35

Output(9) : 53

Output(10) : 51

Output(11) : 35

Store Output :

Output(12) : 52

Output(13) : 57

Output(14) : 32

Output(15) : 32

Output :

79#54#18#53#49

**Kunci Dekripsi RSA : 19-87**

Chipertext :

79#54#18#53#49 dipecah berdasarkan karakter "#" menjadi : 79, 54, 18, 53, dan 49

Dekripsi RSA :

P[0] = 79 ^ 19 Mod 87 = 85

P[1] = 54 ^ 19 Mod 87 = 78

P[2] = 18 ^ 19 Mod 87 = 72

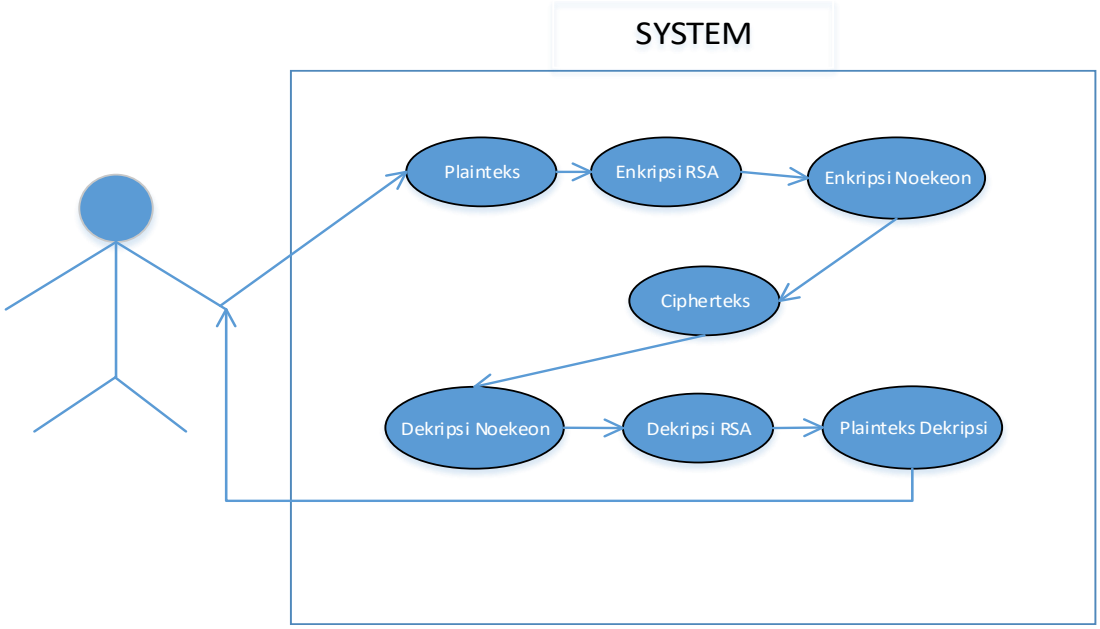
P[3] = 53 ^ 19 Mod 87 = 65

P[4] = 49 ^ 19 Mod 87 = 82

Plainteks : UNHAR

## B. *Use Case Diagram*

*Use case diagram* merupakan gambaran proses pada sistem dari sudu pandang *user*. *Use case diagram* untuk sistem yang akan dibangun dapat dilihat pada gambar 3.1

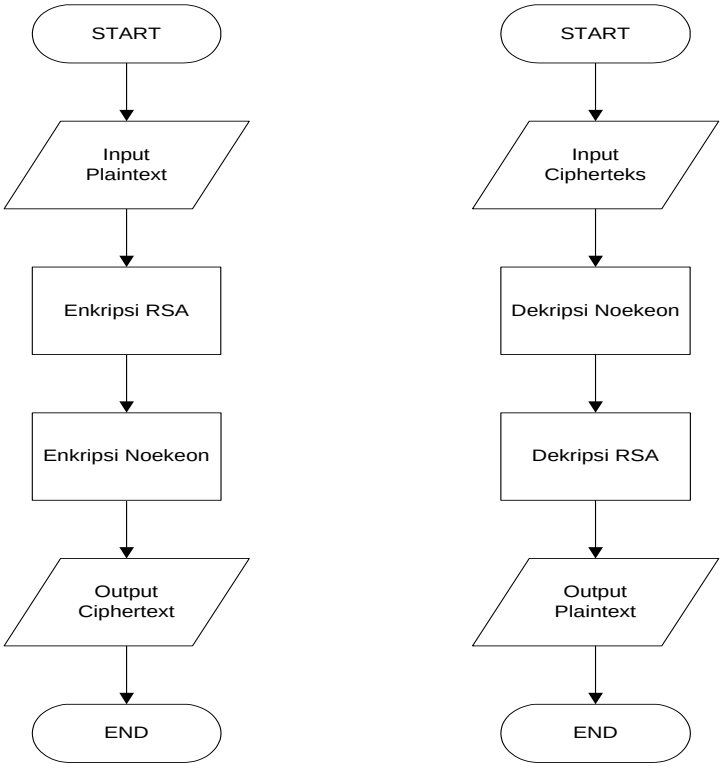



**Gambar 3.1 Use Case Diagram**

*Use Case* pada gambar 3.1, terlihat di mana pengirim melakukan proses enkripsi dan penyimpanan *file* yang berisi *ciphertext*, sedangkan penerima melakukan proses membuka *file* dan proses dekripsi. Proses enkripsi, pertama-tama pengirim meng*input plaintext*. Setelah meng*input text* selanjutnya pengirim melakukan proses enkripsi dengan algoritma RSA dan Noekeon menghasilkan *ciphertext*. *Ciphertext* disimpan dalam direktori di mana pada penelitian ini ekstensi yang digunakan adalah *.txt. Pada proses dekripsi, penerima membuka *file* yang berisi *ciphertext* dan melakukana proses dekripsi dengan algoritma Noekeon dan RSA menghasilkan *plaintext* asli

**C. Flowchart Sistem**

*Flowchart* dari sistem yang akan dibangun dapat dilihat pada Gambar 3.2




**(a)** **(b)**

**Gambar 3.2 *Flowchart* Sistem (a) Enkripsi, (b) Dekripsi**

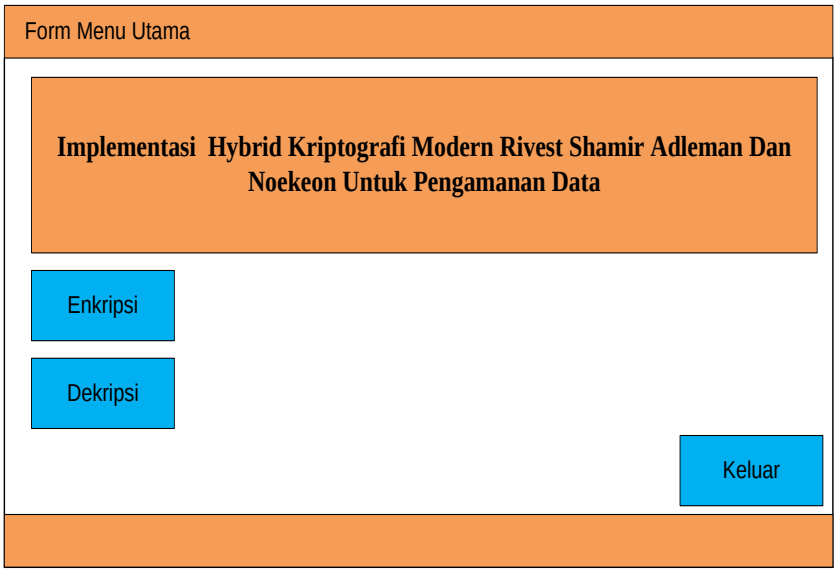
*Flowchart system* diatas menjelaskan mengenai proses yang akan dilalui secara garis besar ada sistem, yaitu pada proses enkripsi akan terlebih dahulu menggunakan algoritma RSA dan kemudian Noekeon. Pada proses dekripsi terlebih dahulu akan digunakan algoritma Noekeon dan kemudian akan diteruskan dengan menggunakan algoritma RS.

**D. Perancangan Antarmuka**

Perancangan ini bertujuan untuk merancang tampilan dari suatu perangkat lunak yang akan di buat yang sesuai dengan kebutuhan pengguna. Berikut perancangan antarmuka aplikasi

enkripsi dan dekripsi data menggunakan metode RSA dan Noekeon

**1. Rancangan Menu Utama Aplikasi**

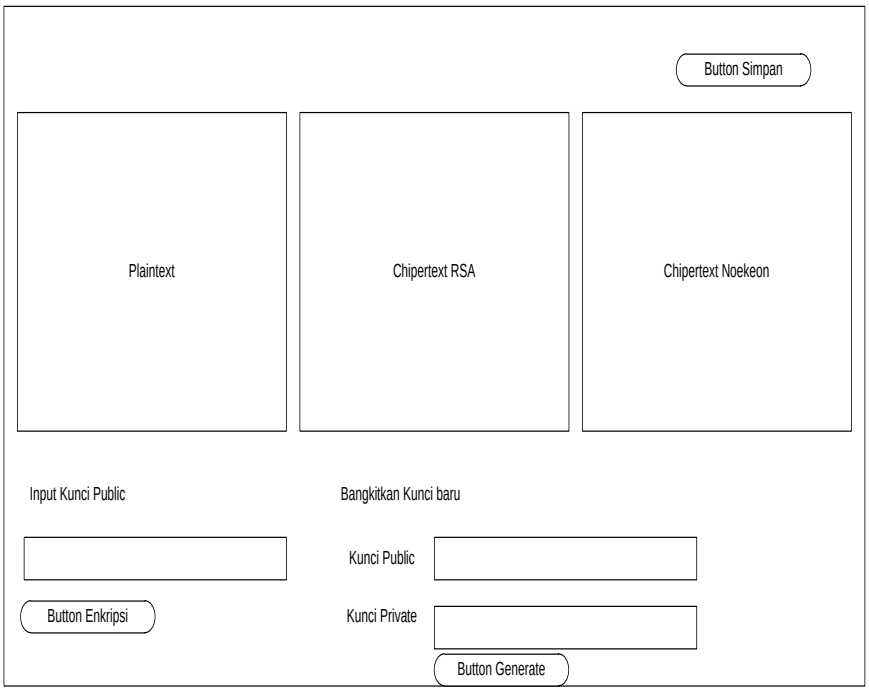



**Gambar 3.3 Rancangan Menu Utama Aplikasi**

Rancangan Antamuka menu utama memiliki menu navigasi yang dapat diakses dengan memilih menu yang terdapat pada antarmuka utama. Menu navigasi pada antarmuka menu utama dapat dijabarkan sebagai berikut :

a. Enkripsi : tombol ini akan mengarahkan pengguna pada halaman enkripsi
b. Dekripsi : tombol ini akan mengarahkan pengguna pada halaman dekripsi

## 2. Rancangan *Form* Enkripsi




**Gambar 3.4 Rancangan Form Enkripsi**

Keterangan dari gambar 3.4 adalah :

1. Plainteks

   Komponen halaman ini berguna untuk menampilkan pesan yang dipilih oleh pengguna untuk dienkripsi.

2. Chiperteks

   Komponen halaman ini berguna untuk menampilkan pesan yang telah dienkripsi melalui proses enkripsi.

3. *Button*_simpan

   Opsi ini berguna untuk menampilkan dialog kepada pengguna untuk menyimpan pesan hasil enkripsi ke media penyimpanan.

4. *Button*_enkripsi

   Opsi ini berguna untuk melakukan operasi enkripsi pada pesan teks yang dipilih kepada pengguna dan pesan hasil enkripsi kemudian ditampilkan pada komponen chiperteks.

5. Text_kunci

   Komponen ini merupakan komponen yang menampung input kunci dari pengguna yang akan digunakan pada proses enkripsi pesan.

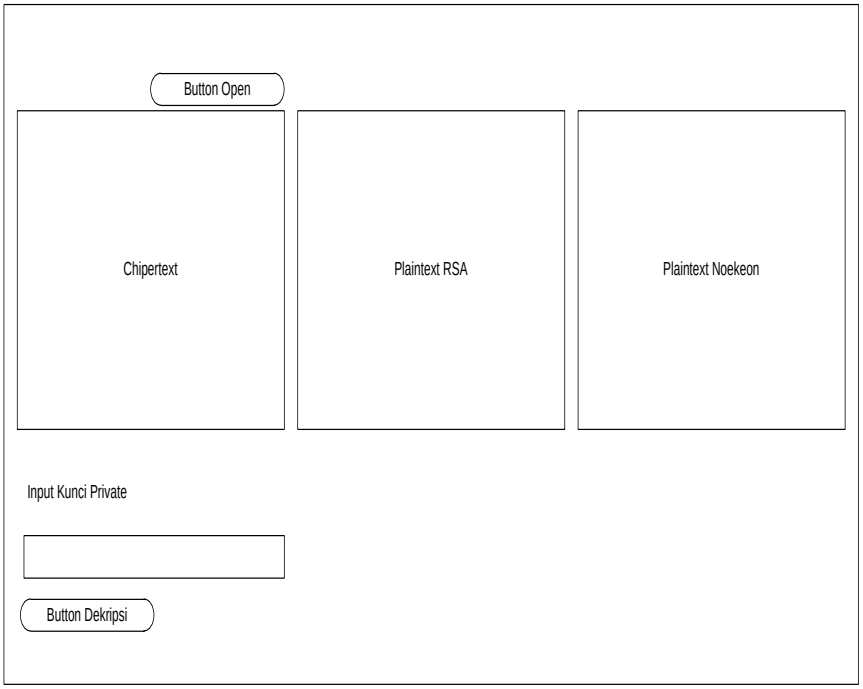
6. Text_kunci_publik

   Komponen ini merupakan komponen yang menampung kunci public dari proses pembangkitan kunci.

7. Text_kunci_*private*

   Komponen ini merupakan komponen yang menampung kunci private dari proses pembangkitan kunci

### 3. Rancangan Form Dekripsi




**Gambar 3.5 Halaman Dekripsi**

Keterangan dari gambar 3.13 adalah :

1. Plainteks RSA

   Komponen halaman ini berguna untuk menampilkan pesan input yang dipilih oleh pengguna untuk didekripsi RSA.

2. Plainteks Noekeon

Komponen halaman ini berguna untuk menampilkan pesan input yang dipilih oleh pengguna untuk didekripsi Noekeon

3. Chiperteks

   Komponen halaman ini berguna untuk menampilkan pesan yang telah didekripsi melalui proses dekripsi.

4. *Button_open*

   Opsi ini berguna untuk menampilkan dialog kepada pengguna untuk memilih pesan input dan memuat pesan tersebut di komponen chipertext.

5. Button_dekripsi

   Opsi ini berguna untuk melakukan operasi dekripsi pada pesan digital yang dipilih kepada pengguna dan pesan hasil dekripsi kemudian ditampilkan pada komponen plainteks.

6. Text_kunci

   Komponen ini merupakan komponen yang menampung input kunci dari pengguna yang akan digunakan pada proses dekripsi pesan

## 3.6 Perancangan Pengujian

Pengujian sistem merupakan proses eksekusi sistem dengan tujuan mencari kesalahan atau kelemahan dari program tersebut. Proses tersebut dilakukan dengan mengevaluasi kemampuan program. Suatu program yang diuji akan dievaluasi apakah keluaran atau output yang dihasilkan telah sesuai dengan yang diinginkan atau tidak. Metode pengujian yang dipakai dalam sistem ini adalah metode *black box.*

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi *input* yang akan mengerjakan seluruh keperluan fungsional program. Tujuan metode ini mencari kesalaman pada:

1. Fungsi yang salah atau hilang
2. Kesalahan pada interface
3. Kesalahan pada struktur data atau akses *database*
4. Kesalahan performansi

# BAB IV

# HASIL DAN PEMBAHASAN

## 4.1 Hasil Aplikasi

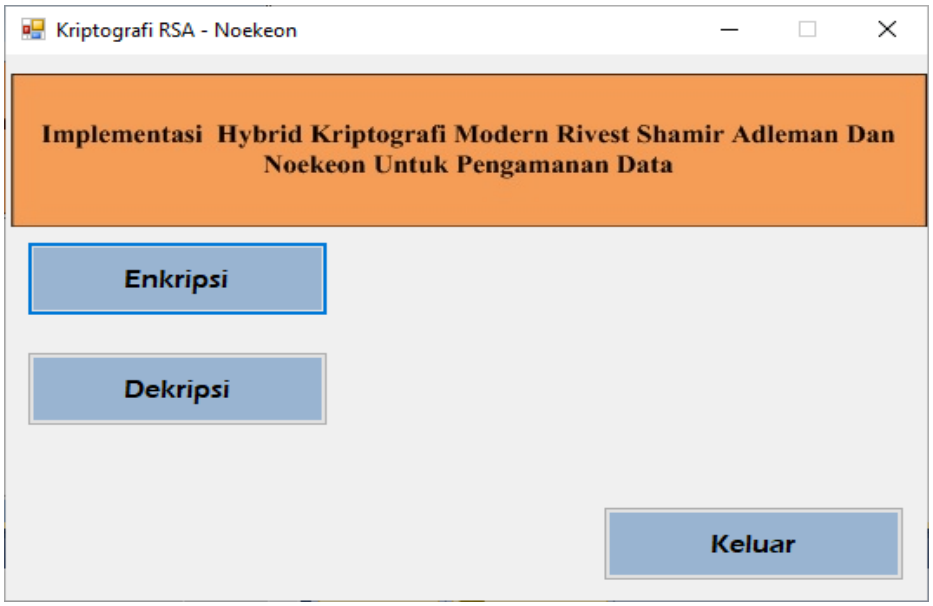
Berdasarkan analisa dan perancangan yang sudah dipaparkan pada bab sebelumnya, maka pada bab ini penulis akan memberikan penjelasan terkait dari perancangan yag dilakukan dan pembuatan sistem berdasarkan rancangan tersebut. Pembahasan pada bab ini sendiri diutamakan terhadap kemampuan dari sistem yang dibuat oleh penulis, dan juga melakukan pengujian terhadap fugsional sistem untuk meyakinkan bahwa sistem yang telah dibangun terbebas dari kesaahan, bug, dan berbagai macam halyang dapat mempengaruhi kinerja dan juga kemampuan sistem dari tujuan pembuatannya. Adapun sistem yang dikembangkan pada penelitian ini terdiri dari beberapa *form* yaitu *form* utama, *form* Enkripsi, *form* dan *form* Dekripsi berdasarkan metode yang digunakan.

## 4.2 Implementasi Sistem

Tujuan dari pembuatan sistem ini adalah menyediakan sebuah penyedian sistem yang mampu memberikan keamanan tambahan bagi data (teks). Data (teks) yang pertama kali akan di enkripsi dengan menggunakan algoritma kriptografi RSA dan kemudian hasil enkripsi tersebut akan di enkripsi kembali dengan menggunakan algoritma kriptografi Noekeon, dan kebalikannya pada saat proses dekripsi. Secara garis besar sistem memiliki dua fungsi utama, yaitu fungsi Enkripsi dan Dekripsi bagi data (teks). Pada

sistem yang dibuat oleh penulis. Berikut akan dijelaskan mengenai proses yang ada didalam sistem

## A. *Form* Menu Utama Aplikasi




**Gambar 4.1 *Form* Menu Utama Aplikasi**

Pada tampilan *form* menu utama, terdapat informasi judul dari aplikasi yang dibangun. Tampilan halaman pembuka merupakan tampilan yang pertama sekali muncul pada saat aplikasi dijalankan.

Adapun bagian-bagian dari menu utama

1. ***Button* (Tombol) Enkripsi**

   Fungsi dari *button* (tombol) enkripsi adalah untuk menampilkan tampilan enkripsi yang digunakan untuk meng-enkripsi data dari pengguna

2. ***Button* (Tombol) Dekripsi**

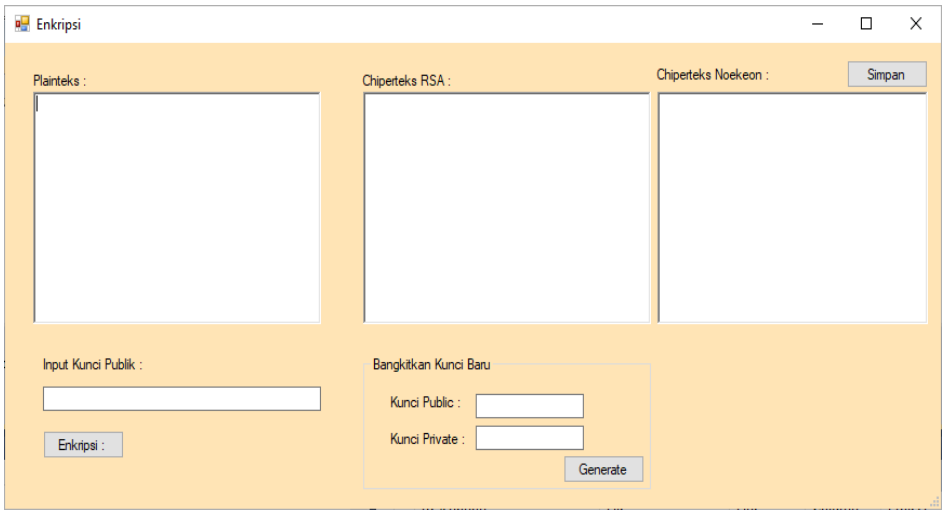
   Fungsi dari *button* (tombol) dekripsi adalah untuk menampilkan tampilan dekripsi yang digunakan untuk men-dekripsi data dari pengguna

### 3. *Button* (Tombol) Keluar

Fungsi dari *button* (tombol) keluar adalah untuk keluar dan menutup aplikasi.

## B. *Form* Enkripsi

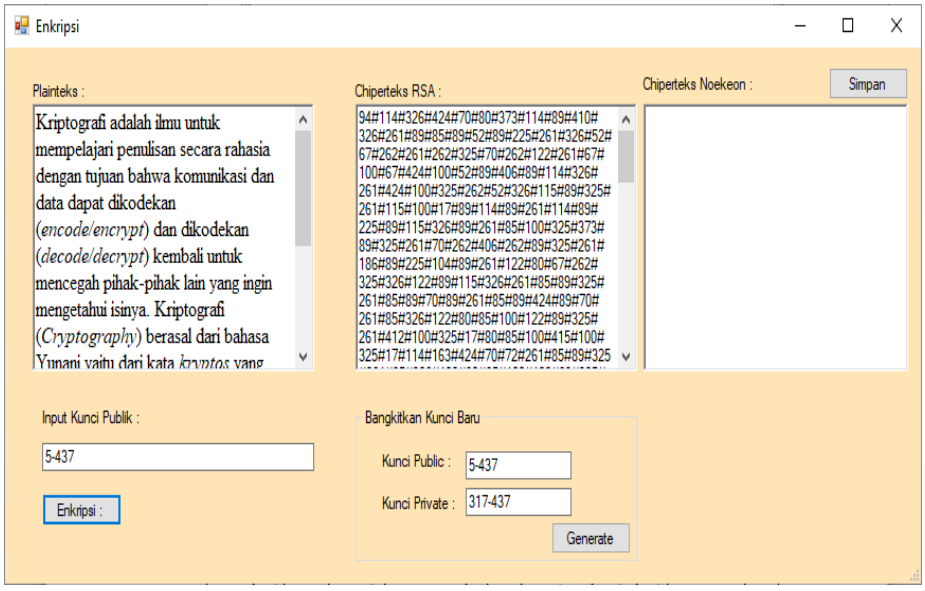
*Form* ini digunakan untuk melakukan enkripsi terhadap berkas digital yang di masukkan oleh pengguna. Proses enkripsi merupakan proses dimana data teks yang asli akan mengalami perubahan isi (konten data) sehingga tidak memungkinkan dibaca. Proses enkripsi sendiri adalah sebuah proses pyang memanfaatkan penggunaan password dan juga algoritma kriptografi (algoritma kriptografi yang dibahas adalah RSA dan Noekeon) dalam upaya untuk pengubahan plaintext menjadi chipertext atau enkripsi




**Gambar 4.2 *Form* Enkripsi**

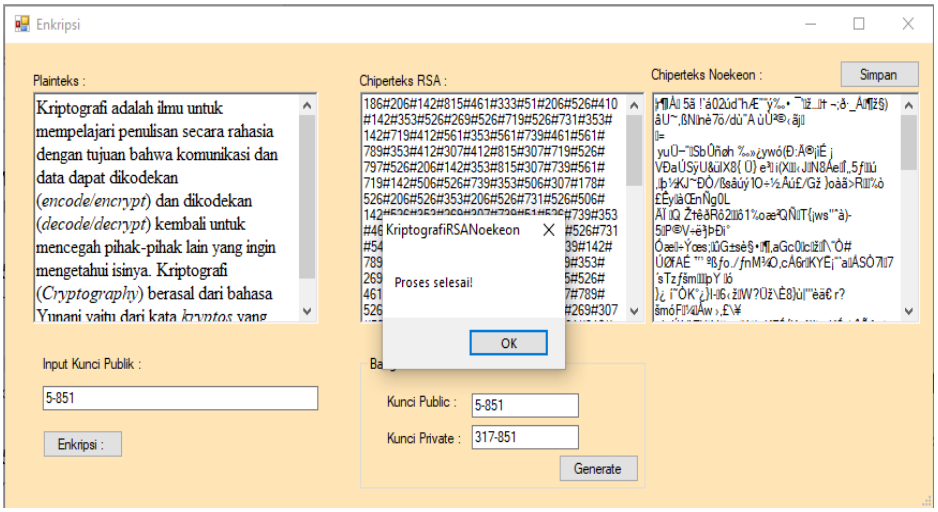
Seperti yang dapat kita lihat pada gambar 4.2 ada cukup banyak *textbox* yang dapat digunakan pada *form* ini. Pertama adalah *textbox* plainteks, pada *textbox* ini pengguna akan mengetikkan teks yang akan dienkripsi. Selanjutnya adalah *textbox chipertext* RSA, pada textbox ini

sendiri akan ditampilkan hasil enkripsi dari algoritma RSA. Sesuai dengn algoritm sistem dimana sistem akan melakukan enkripsi RSA terlebih dahulu maka pada aplikasi pun sistem akan menampilkan hasil enkripsi dari RSA. Selanjutnya pada *textbox chipertext noekeon*, adalah tempat dimana hasil enkripsi akhir akan ditampilkan, secara algoritma *textbox* ini akan menampilkan hasil enkripsi *noekeon* terhadap *chipertext* yang dihasilkan oleh algoritma RSA. Kemudian kita akan berpindah pada tiga *textbox* lainnya yaitu kunci *public*, kunci *private*, dan input kunci public. Pertama-tama pengguna akan *mengenerate* kunci public dan kunci private dengan menggunakan algoritma pembangkit kunci yang dimiliki oleh kriptografi RSA. Kemudian pengguna akan mengetikkan ulang atau meng-copy kunci public pada textbox input kunci *public*, Untuk lebih memperjelas kegunaan konten dalam halaman enkripsi maka akan dijelaskan pada gambar 4.3 berikut




**Gambar 4.3 *Antarmuka Proses Enkripsi***

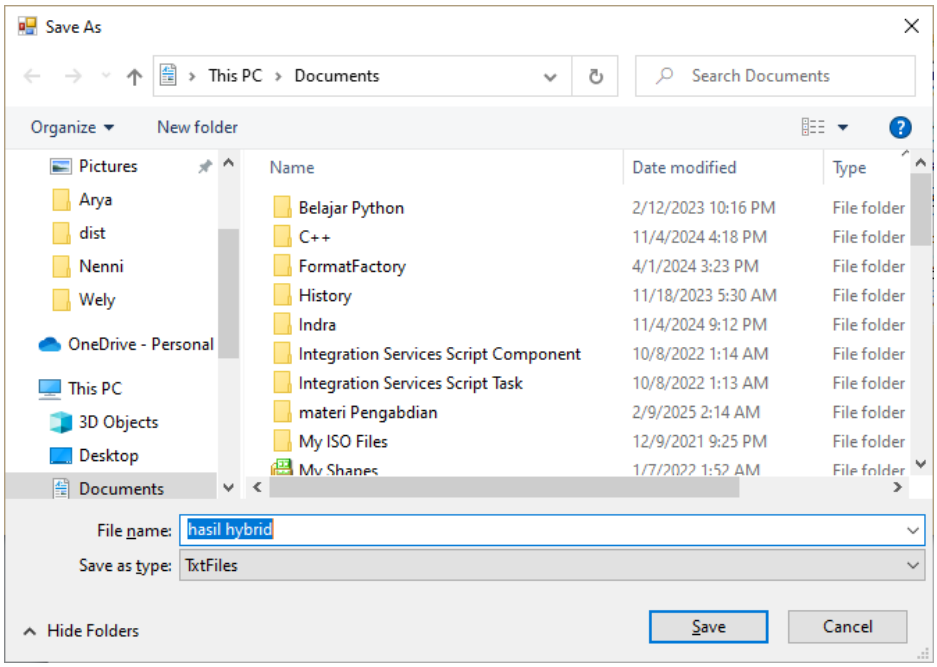
Pada gambar 4.3 diatas, kunci telah terlebih dahulu digenerate dengan menggunakan algoritma pembangkit kunci RSA dan akan menghasilkan kunci yang akan digunakan yaitu kunci public (5-437 ) dan kunci private (317-437). Langkah selanjutnya dalam penggunaan aplikasi ini adalah memasukkan text atau pesan yang akan dienkripsi. Seperti yang terlihat pada gambar diatas, penulis telah memasukkan teks yang diinginkan. Langkah selanjutnya adalah memasukkan kunci public kedalam textbox input kunci publik untuk mempersiapkan proses enkripsi. Kemudian penguna dapat menekan tombol enkripsi yang akan memerintahkan sistem untuk melaksanakan proses enkripsi *hybrid* RSA dan Noekeon




**Gambar 4.4 Antarmuka Proses Enkrips Selesai**

Seperti yang terlihat pada gambar 4.4 diatas proses enkripsi telah berhasil dilakukan. Tampak pada gambar meskipun tombol enkripsi hanya ditekan 1 kali sistem akan tetap mengeksekusi proses enkripsi sesuai dengan algoritma yang ditentukan yaitu RSA terlebih dahulu kemudian baru mengenkripsi hasil chipertext RSA dengan algoritm Noekeon sehingga menghasilkan chipertext yang melewati proses *double*

*encryption system* dengan menggunaan *hybrid* kripto sistem. Langkah selanjutnya adalah menyimpan data, seperti yang tampak pada gambar 4.5 file akan disimpan dengan nama “enkripsi” dan data hasil enkripsi akan disimpan dengan format .TXT. peyimpanan dengan format ini dikarenakan nanti pada saat akan melaksanakan proses dekripsi, sistem juga hanya akan meneripa inputan data dengan format .TXT sehingga penulis merasa perlu untuk membatasi format data penyimpanan hasil enkripsi

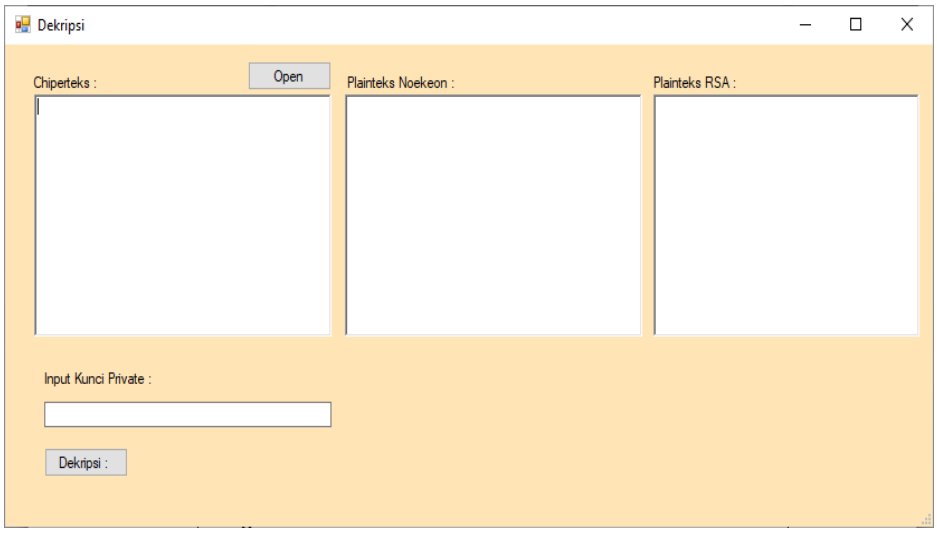



**Gambar 4.5 Kotak Dialgo Simpan *File Chipertext***

## C. *Form* Dekripsi

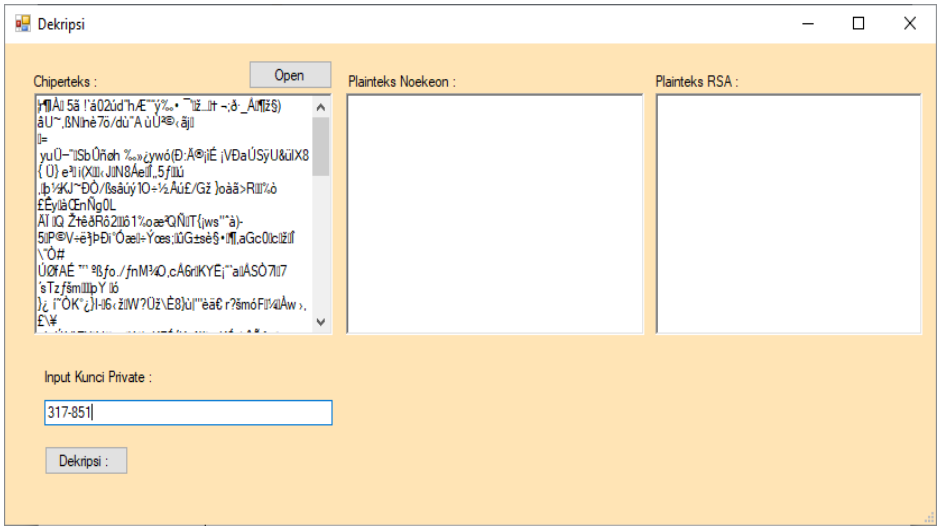
*Form* dekripsi sendiri adalah sebuah proses yang bertolak belakang dengan proses enkripsi. Proses dekripsi adalah proses yang akan mengembalikan kondisi data teks yang semula dalam kondisi yang tidak bisa dibaca, kembali menjadi kondisi semula sebelum data teks mengalami proses enkripsi. Proses dekripsi dilakukan dengan

menggunakan password yang sama dengan password yang digunakan dalam proses enkripsi dan juga tentunya dengan menggunakan algoritma yang sama



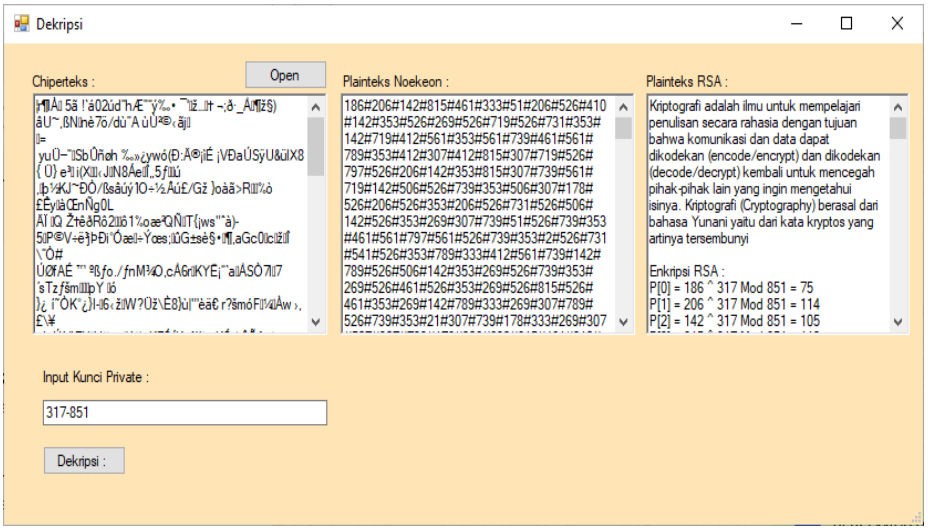

**Gambar 4.6 *Form* Dekripsi**

*File* chiperteks dapat dibuka dengan menggunakan tombol "Open" yang kemudian akan menampilkan dialog untuk memilih file chiperteks yang akan di – dekripsi. Setelah chiperteks dibuka selanjutnya pengguna dapat melakukan proses dekripsi dengan memasukkan kunci yang telah dibangkitkan sebelumnya dan melakukan proses dekripsi dengan menekan tombol "Dekripsi" sehingga proses dekripsi akan dilakukan dan akan menampilkan plainteks seperti yang terlihat pada gambar 4.7




**Gambar 4.7 *Form Dekripsi* dengan *File* Enkripsi dan Kunci *Private***

Hasil akhir dari proses dekripsi yang dijabarkan pada gambar 4.7 adalah teks dari file yang telah ter enkripsi dapat dikembalikan pada kondisi awal sama seperti sebelum file melewati proses enkripsi. Tampak pada gambar 4.8 dapat dilihat bahwasanya untuk dapat mengembalikan teks kedalam kondisi sebelum di enkripsi atau plaintext, sistem melakukan proses dekripsi secara terbalik apabila dibandingkan dengan proses enkripsi. Tampak pertama kali chiperteks akan di dekripsi dengan menggunakan algoritma Noekeon yang kemudin hasil dekripsi dari Noekeon akan dilanjutkan untuk melewati proses dekripsi pada algoritma RSA. Setelah kedua prose tersebut selesai dilakukan dapat dilihat pada textbox plaintext RSA adalah paintext awal yang dapat dibaca




**Gambar 4.8 *Form Dekripsi* Dengan File Enkripsi dan Kunci Private dan Hasil Dekripsi**

## 4.3 Pengujian Sistem

Pengujian yang dilakukan pada aplikasi ini adalah dengan menggunakan teknik *black box*, teknik *black box* ini merupakan teknik pengujian yang berfokus pada keluaran hasil dari respon, atau secara simpel untuk mengetahui apakah ada *error* atau ada fungsi yang tidak berjalan sesuai dengan harapan. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal, yaitu mampu mempresentasikan kajian pokok dari spesifikasi analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri. Berikut tabel pengujian *black box*

**Tabel 4.1 Pengujian *Black Box***

| Jenis Uji | Keterangan Uji | Jenis Pengujian |
|---|---|---|
| *Enkripsi* | Proses Enkripsi | *Black Box* |
| Key public dan Private | Generate key public dan private | *Black Box* |
| Dekripsi | Proses Dekripsi | *Black Box* |

| Kasus dan Hasil Uji | | | |
|---|---|---|---|
| **Data Masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| Enkripsi | Plaintext berhasil di enkripsi | File berhasil berubah sesuai dengan kunci yang digunakan | [x] diterima<br>[ ] ditolak |
| Key public dan Private | Menampilkan hasil kunci public dan private | Kunci public dan private tidak pernah sama setiap melakukan enkripsi | [x] diterima<br>[ ] ditolak |
| Dekripsi | File berhasil di dekripsi | Plaintext berhasil kembali menjadi plainteks dengan menggunakan kunci sama pada saat enkripsi | [x] diterima<br>[ ] ditolak |

# BAB V

# KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

Berdasarkan pembahasan dan pengujian program yang dilakukan, maka dapat di tarik kesimpulan sebagai berikut :

1. Pengamanan data teks dengan menerapkan algoritma RSA dan Neokeon dapat dikombinasikan dengan baik. Proses enkripsi dimulai terlebih dahulu menggunakan metode *RSA* yang kemudian hasil enkripsi tersebut di enkripsi lagi menggunakan metode *Noekeon* sehingga seperti proses enkripsi beruntun. Pada proses enkripsi Algoritma *Noekeon*, kunci yang digunakan adalah komponen *n* dari kunci RSA dimana komponen *n* pada kunci *public* dan kunci *private* bernilai sama sehingga dapat digunakan pada saat enkripsi dan dekripsi. Pada proses dekripsi, chiperteks hasil *Noekeon* di dekripsi terlebih dahulu menggunakan algoritma *Noekeon* yang mana hasil *plainteks* dari algoritma *Noekeon* di dekripsi lebih lanjut menggunakan algoritma RSA menggunakan kunci *privat*
2. Hasil implementasi kombinasi pengamanan pesan teks menggunakan RSA dan *Noekeon* menunjukkan hasil yang cukup baik, dimana pesan teks yang ter enkripsi memiliki keunggulan dalam proses autentikasi dimana menerapkan kunci *public* dan kunci *private* dari algoritma RSA serta memiliki keunggulan dalam kompleksitas chiperteks dari algoritma *Noekeon*

## 5.2 Saran

Saran - saran yang penulis kemukakan diharapkan dapat lebih meningkatkan hasil yang telah didapatkan. Berikut ini beberapa saran yang disampaikan oleh penulis adalah :

1. Aplikasi yang dikembangkan tidak hanya yang berbasis *desktop* tapi dapat juga dapat dikembangkan aplikasi yang berbasis *mobile*
2. Aplikasi dapat dikembangkan dengan menerapkan sistem *user registration* sehingga setiap *user* memiliki kunci *public* dan *private* yang tetap
3. Diharapkan pada penelitian mendatang dapat mengembangkan kombinasi – kombinasi lain yang lebih lengkap dan lebih aman

# DAFTAR PUSTAKA


Ayumida, S., Azis, M. S., & Fiano, Z. G. (2020). Implementasi Program Administrasi Pembayaran Berbasis Dekstop (Studi Kasus: SMA Negeri 1 Cikampek). *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, 72-83.

Dairi, M., Asih, M. S., & Khairunnisa. (2023). Implementasi Algoritma Kriptografi RSA Dalam Aplikasi Sistem Informasi Perpustakaaan. *Jurnal Ilmu Komputer dan Sistem Informasi (JIRSI)* , 214-223.

Firmansyah, R., & Permana, A. A. (2019). Implementasi Keamanan Pesan Teks Menggunakan Kriptografi Algoritma Rsa Dengan Metode Waterfall Berbasis Java. *JOUTICA* , 217-221.

Gunawan, & Kirman. (2019). Implementasi Algoritma Turbo Boyer Moore Untuk Pencarian Data Pada Transaksi Keuangan Duta Phonecell Sawah Lebar. *Jurnal Media Infotama*, 9-15.

Hasan, H. I., & Safrizal, S. (2024). Penerapan Algoritma Affine Cipher Untuk Keamanan Data Registrasi Siswa Baru. *Seminar Nasional Multi Disiplin Ilmu (SENADIMU)*, 1029-1042.

Limantoro, R. R., & Kristiadi, D. P. (2021). Pengembangan Sistem Informasi Pendataan Green Folder Menggunakan Metode Berorientasi Objek Dan UML Berbasis Web Pada TKHarvest Christian School. *JURNAL SISTEM INFORMASI DAN TEKNOLOG (SINTEK)*, 7-14.

Milawati,, Silalahi, N., & Tampubolon, K. (2021). Analisa Algoritma Noekeon Untuk Mengamankan File Video. *JURIKOM (Jurnal Riset Komputer),*, 80-88.

Muafi, Wijaya, A., & Aziz, V. A. (2020). Sistem Pakar Mendiagnosa Penyakit Mata Pada Anusia Menggunakan Metode Forward Chaining. *Jurnal Komputasi dan Teknologi Informasi*, 43-49.

Rahmat, E., Jumadi, J., & Lianda, D. (2024). Implementasi Kriptografi Untuk Keamanan Database Dengan Menggunakan Algoritma Twofish Pada Puskesmas Ilir Talo. *Journal of Science and Social Research*, 1320-1326.

Ridho, A., Mutia, C., & Sinaga, A. P. (2022). Analisis Enkripsi Dan Dekripsi Cipher Teks Menggunakan Kombinasi Gronsfeld Cipher Dengan Reverse Cipher. *JTIK (Jurnal Teknik Informatika Kaputam)*, 87-94.

Samsudin, A., & Islami, H. H. (2023). Sistem Pengaduan Masyarakat Menggunakan Metode Agile Extreme Programming. *Jurnal Infotex*, 214-226.

Sasono, D. M., Tahir, M., Angel M, F., Azizah, M., Utami, L. F., & Septiana, N. (2023). Perbandingan Kriptography Klasik Caesar Cipher Dengan Kriptography Modern Aes Dalam Tingkat Keamanan Jaringan Kompute. *Jurnal Informasi, Sains dan Teknologi*, 72-77.

Siagian, K. Z., & Triandi, B. (2024). Implementasi Aplikasi KeamananData Karyawan Pada PT. Jaya Diesel Menggunakan Metode Affine Chiper Dan Rsa Berbasis Web. *Jurna l Info Digit (JDI)*, 672-683.

Sutrisno, J., & Karnadi, V. (2021). Aplikasi Pendukung Pembelajaran Bahasa Inggris Menggunakan Media Lagu Berbasis Android. *JURNAL COMASIE*, 31-41.

Wahidin, U., Sarbini, M., Maulida, A., & Wangsadanureja, M. (2021). Implementasi Pembelajaran Agama Islam Berbasis Multimedia Di Pondok Pesantren. *Edukasi Islami: Jurnal Pendidikan Islam*, 21-32.

Wahyudi, Hartama, D., Kirana, I. O., Sumarno, & Gunawan, I. (2022). Implementasi Algoritma Kriptografi Rivest Shamir Adlemen untuk Mengamankan Data Ijazah pada SMK Swasta Prama Artha Kab. Simalungun. *Jurnal Ilmu Komputer dan Informatika (JIKI)*, 57-66.

Zachary, M. Z., Sylviani, S., & Kurniadi, E. (2024). Implementasi Algoritma RSA (Rivest-Shamir-Adleman) Pada Kriptografi Klasik. *Mathematical Sciences and Applications Journal*, 54-59.

# L A M P I R A N

## 1. Kode Program

### Noekeon

```
Public Class ClNoekeon
    Public Class BBlock
        Public B(3) As UInteger
    End Class

    Public LBlock As List(Of BBlock)

    Public SProses As String

    Private EK() As UInteger
    Private DK() As UInteger

    Public tOutput() As Byte

    'Public RC() As Byte = {Convert.ToByte("80", 16),
Convert.ToByte("1b", 16), _
    '    Convert.ToByte("36", 16), Convert.ToByte("6c", 16),
Convert.ToByte("d8", 16), _
    '    Convert.ToByte("ab", 16), Convert.ToByte("4d", 16),
Convert.ToByte("9a", 16), _
    '    Convert.ToByte("2f", 16), Convert.ToByte("5e", 16),
Convert.ToByte("bc", 16), _
    '    Convert.ToByte("63", 16), Convert.ToByte("c6", 16),
Convert.ToByte("97", 16), _
    '    Convert.ToByte("35", 16), Convert.ToByte("6a", 16),
Convert.ToByte("d4", 16)}

    Public RC() As Byte

    Public Sub GenerateRC()
        ReDim RC(16)

        Dim i As Integer

        RC(0) = Convert.ToByte("80", 16)
        For i = 1 To 16
            If (RC(i - 1) And Convert.ToByte("80", 16)) <> 0 Then
                RC(i) = Rotate_Left2(RC(i - 1), 1)
                RC(i) = RC(i) Xor Convert.ToByte("1b", 16)
            Else
                RC(i) = Rotate_Left2(RC(i - 1), 1)
            End If
            SProses = SProses & "RC[" & i & "] = " & RC(i)
        Next
    End Sub

    Public Sub GenerateBlock(ByVal tInput As String)
        Dim i As Long
        Dim j As Integer
        Dim tB As BBlock
        LBlock = New List(Of BBlock)

        Dim Tamp As String
```

```
        Do While Len(tInput) Mod 16 <> 0
            tInput = tInput & " "
        Loop

        j = 0
        i = 1
        tB = New BBlock
        SProses = SProses & "Pembentukan Blok Input : " & vbCrLf
        Do While i <= Len(tInput)
            SProses = SProses & "Blok Input (" & j & ") : " &
Dec2Bin8(Asc(Mid(tInput, i, 1))) & ";" & Dec2Bin8(Asc(Mid(tInput, i +
1, 1))) & ";" & Dec2Bin8(Asc(Mid(tInput, i + 2, 1))) & ";" &
Dec2Bin8(Asc(Mid(tInput, i + 3, 1))) & vbCrLf
            tB.B(j) = Bin2Dec(Dec2Bin8(Asc(Mid(tInput, i, 1))) &
Dec2Bin8(Asc(Mid(tInput, i + 1, 1))) & Dec2Bin8(Asc(Mid(tInput, i + 2,
1))) & Dec2Bin8(Asc(Mid(tInput, i + 3, 1))))

            j += 1
            If j > 3 Then
                SProses = SProses & "End (" & j & ") BLok " & vbCrLf &
vbCrLf
                LBlock.Add(tB)
                tB = New BBlock
                j = 0
            End If
            i = i + 4
        Loop
    End Sub

    Public Sub GenerateBlock2(ByVal tInput() As Byte, ByRef tProgress
As Integer)
        Dim i As Long
        Dim j As Integer
        Dim tB As BBlock
        LBlock = New List(Of BBlock)

        'Do While Len(tInput) Mod 16 <> 0
        '    tInput = tInput & " "
        'Loop
        i = 0
        'Do While i <= tInput.GetUpperBound(0)
        '    SProses = SProses & "Piksel Input : " & tInput(i) & ";" &
tInput(i + 1) & ";" & tInput(i + 2) & vbCrLf
        '    i += 3
        'Loop

        j = 0
        i = 0
        tB = New BBlock
        SProses = SProses & "Pembentukan Blok Input : " & vbCrLf
        Do While i <= tInput.GetUpperBound(0)
            SProses = SProses & "Blok Input (" & j & ") : " &
Dec2Bin8(tInput(i)) & ";" & Dec2Bin8(tInput(i + 1)) & ";" &
Dec2Bin8(tInput(i + 2)) & ";" & Dec2Bin8(tInput(i + 3)) & vbCrLf
            tB.B(j) = Bin2Dec(Dec2Bin8(tInput(i)) & Dec2Bin8(tInput(i +
1)) & Dec2Bin8(tInput(i + 2)) & Dec2Bin8(tInput(i + 3)))

            j += 1
            If j > 3 Then
```

```
                SProses = SProses & "End (" & j & ") BLok " & vbCrLf &
vbCrLf
                LBlock.Add(tB)
                tB = New BBlock
                j = 0
            End If
            tProgress = (1 / tInput.GetUpperBound(0)) * 100
            Application.DoEvents()
            i = i + 4
        Loop
    End Sub

    Private Sub Theta(ByRef A0 As UInteger, ByRef A1 As UInteger, ByRef
A2 As UInteger, ByRef A3 As UInteger, ByVal Kunci() As UInteger)
        Dim T As UInteger
        Dim Temp1 As UInteger

        SProses = SProses & "Theta Proses : " & vbCrLf
        SProses = SProses & "A0 : " & A0 & vbCrLf
        SProses = SProses & "A1 : " & A1 & vbCrLf
        SProses = SProses & "A2 : " & A2 & vbCrLf
        SProses = SProses & "A3 : " & A3 & vbCrLf
        SProses = SProses & "K(0) : " & Kunci(0) & vbCrLf
        SProses = SProses & "K(1) : " & Kunci(1) & vbCrLf
        SProses = SProses & "K(2) : " & Kunci(2) & vbCrLf
        SProses = SProses & "K(3) : " & Kunci(3) & vbCrLf

        SProses = SProses & "T = A0 Xor A2" & vbCrLf
        T = A0 Xor A2
        SProses = SProses & "  = " & T & vbCrLf

        Temp1 = Rotate_Left(T, 8) Xor Rotate_Right(T, 8)
        SProses = SProses & "Temp = T<<8 Xor T>>8 = " & Temp1 & vbCrLf

        T = T Xor Temp1
        A1 = A1 Xor T
        A3 = A3 Xor T
        SProses = SProses & "A1 = T Xor Temp Xor A1 = " & A1 & vbCrLf
        SProses = SProses & "A3 = T Xor Temp Xor A3 = " & A3 & vbCrLf

        A0 = A0 Xor Kunci(0)
        A1 = A1 Xor Kunci(1)
        A2 = A2 Xor Kunci(2)
        A3 = A3 Xor Kunci(3)
        SProses = SProses & "A0 = A0 Xor Kunci(0) = " & A0 & vbCrLf
        SProses = SProses & "A1 = A1 Xor Kunci(1) = " & A1 & vbCrLf
        SProses = SProses & "A2 = A2 Xor Kunci(2) = " & A2 & vbCrLf
        SProses = SProses & "A3 = A3 Xor Kunci(3) = " & A3 & vbCrLf

        T = A1 Xor A3
        Temp1 = Rotate_Left(T, 8) Xor Rotate_Right(T, 8)
        SProses = SProses & "T = A1 Xor A3 = " & T & vbCrLf
        SProses = SProses & "Temp = T<<8 Xor T>>8 = " & Temp1 & vbCrLf
        T = T Xor Temp1

        A0 = A0 Xor T
        A2 = A2 Xor T
        SProses = SProses & "A0 = T Xor Temp Xor A0 = " & A0 & vbCrLf
        SProses = SProses & "A2 = T Xor Temp Xor A2 = " & A2 & vbCrLf
    End Sub
```

```
    Private Sub Theta(ByRef A0 As UInteger, ByRef A1 As UInteger, ByRef
A2 As UInteger, ByRef A3 As UInteger)
        Dim T As UInteger
        Dim Temp1 As UInteger

        SProses = SProses & "Theta Inv Proses : " & vbCrLf
        SProses = SProses & "A0 : " & A0 & vbCrLf
        SProses = SProses & "A1 : " & A1 & vbCrLf
        SProses = SProses & "A2 : " & A2 & vbCrLf
        SProses = SProses & "A3 : " & A3 & vbCrLf

        SProses = SProses & "T = A0 Xor A2" & vbCrLf
        T = A0 Xor A2
        SProses = SProses & "  = " & T & vbCrLf
        'Temp1 = Rotate_Right(T, 8)
        Temp1 = Rotate_Left(T, 8) Xor Rotate_Right(T, 8)
        SProses = SProses & "Temp = T<<8 Xor T>>8 = " & Temp1 & vbCrLf

        T = T Xor Temp1
        A1 = A1 Xor T
        A3 = A3 Xor T
        SProses = SProses & "A1 = T Xor Temp Xor A1 = " & A1 & vbCrLf
        SProses = SProses & "A3 = T Xor Temp Xor A3 = " & A3 & vbCrLf

        T = A1 Xor A3
        Temp1 = Rotate_Left(T, 8) Xor Rotate_Right(T, 8)
        SProses = SProses & "T = A1 Xor A3 = " & T & vbCrLf
        SProses = SProses & "Temp = T<<8 Xor T>>8 = " & Temp1 & vbCrLf
        T = T Xor Temp1

        A0 = A0 Xor T
        A2 = A2 Xor T
        SProses = SProses & "A0 = T Xor Temp Xor A0 = " & A0 & vbCrLf
        SProses = SProses & "A2 = T Xor Temp Xor A2 = " & A2 & vbCrLf
    End Sub

    Private Sub Gamma(ByRef A0 As UInteger, ByRef A1 As UInteger, ByRef
A2 As UInteger, ByRef A3 As UInteger)
        Dim Temp1 As UInteger
        Dim T As UInteger

        SProses = SProses & "Gamma Proses : " & vbCrLf
        SProses = SProses & "A0 : " & A0 & vbCrLf
        SProses = SProses & "A1 : " & A1 & vbCrLf
        SProses = SProses & "A2 : " & A2 & vbCrLf
        SProses = SProses & "A3 : " & A3 & vbCrLf

        Temp1 = (Not A3) And (Not A2)
        A1 = A1 Xor Temp1
        SProses = SProses & "A1 = A1 Xor ((Not A3) And (Not A2)) = " &
A1 & vbCrLf

        Temp1 = A2 And A1
        A0 = A0 Xor Temp1
        SProses = SProses & "A0 = A0 Xor (A2 And A1) = " & A0 & vbCrLf

        T = A3
        A3 = A0
        A0 = T
        SProses = SProses & "A0 = A3 = " & A0 & vbCrLf
```

```
        SProses = SProses & "A3 = A0 = " & A3 & vbCrLf

        Temp1 = A0 Xor A1
        T = Temp1 Xor A3
        A2 = A2 Xor T
        SProses = SProses & "A2 = A2 Xor A3 Xor A1 Xor A0 = " & A2 &
vbCrLf

        Temp1 = (Not A3) And (Not A2)
        A1 = A1 Xor Temp1
        Temp1 = A2 And A1
        A0 = A0 Xor Temp1
        SProses = SProses & "A1 = A1 Xor (Not A3 And Not A2) = " & A1 &
vbCrLf
        SProses = SProses & "A0 = A0 Xor (A2 And A1) = " & A0 & vbCrLf
& vbCrLf
    End Sub

    Public Sub KeySchedule(ByVal tKunci As String)
        Dim i As Integer
        Dim A0 As UInteger
        Dim A1 As UInteger
        Dim A2 As UInteger
        Dim A3 As UInteger

        SProses = SProses & "Penjadwalan Kunci : " & vbCrLf
        Do While Len(tKunci) < 16
            tKunci = tKunci & "0"
        Loop

        SProses = SProses & "Kunci Input : " & tKunci & vbCrLf
        SProses = SProses & "Kunci(Bit) : " & Dec2Bin8(Asc(Mid(tKunci,
1, 1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 2, 1))) & ";" &
Dec2Bin8(Asc(Mid(tKunci, 3, 1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 4,
1))) & ";" & vbCrLf
        SProses = SProses & Dec2Bin8(Asc(Mid(tKunci, 5, 1))) & ";" &
Dec2Bin8(Asc(Mid(tKunci, 6, 1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 7,
1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 8, 1))) & ";" & vbCrLf
        SProses = SProses & Dec2Bin8(Asc(Mid(tKunci, 9, 1))) & ";" &
Dec2Bin8(Asc(Mid(tKunci, 10, 1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 11,
1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 12, 1))) & ";" & vbCrLf
        SProses = SProses & Dec2Bin8(Asc(Mid(tKunci, 13, 1))) & ";" &
Dec2Bin8(Asc(Mid(tKunci, 14, 1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 15,
1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 16, 1))) & ";" & vbCrLf
        A0 = Bin2Dec(Dec2Bin8(Asc(Mid(tKunci, 1, 1))) &
Dec2Bin8(Asc(Mid(tKunci, 2, 1))) & Dec2Bin8(Asc(Mid(tKunci, 3, 1))) &
Dec2Bin8(Asc(Mid(tKunci, 4, 1))))
        A1 = Bin2Dec(Dec2Bin8(Asc(Mid(tKunci, 5, 1))) &
Dec2Bin8(Asc(Mid(tKunci, 6, 1))) & Dec2Bin8(Asc(Mid(tKunci, 7, 1))) &
Dec2Bin8(Asc(Mid(tKunci, 8, 1))))
        A2 = Bin2Dec(Dec2Bin8(Asc(Mid(tKunci, 9, 1))) &
Dec2Bin8(Asc(Mid(tKunci, 10, 1))) & Dec2Bin8(Asc(Mid(tKunci, 11, 1))) &
Dec2Bin8(Asc(Mid(tKunci, 12, 1))))
        A3 = Bin2Dec(Dec2Bin8(Asc(Mid(tKunci, 13, 1))) &
Dec2Bin8(Asc(Mid(tKunci, 14, 1))) & Dec2Bin8(Asc(Mid(tKunci, 15, 1))) &
Dec2Bin8(Asc(Mid(tKunci, 16, 1))))
        SProses = SProses & "A0 : " & Dec2Bin8(Asc(Mid(tKunci, 1, 1)))
& ";" & Dec2Bin8(Asc(Mid(tKunci, 2, 1))) & ";" &
Dec2Bin8(Asc(Mid(tKunci, 3, 1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 4,
1))) & " = " & A0 & vbCrLf
```

```
        SProses = SProses & "A1 : " & Dec2Bin8(Asc(Mid(tKunci, 5, 1)))
& ";" & Dec2Bin8(Asc(Mid(tKunci, 6, 1))) & ";" &
Dec2Bin8(Asc(Mid(tKunci, 7, 1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 8,
1))) & " = " & A1 & vbCrLf
        SProses = SProses & "A2 : " & Dec2Bin8(Asc(Mid(tKunci, 9, 1)))
& ";" & Dec2Bin8(Asc(Mid(tKunci, 10, 1))) & ";" &
Dec2Bin8(Asc(Mid(tKunci, 11, 1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 12,
1))) & " = " & A2 & vbCrLf
        SProses = SProses & "A3 : " & Dec2Bin8(Asc(Mid(tKunci, 13, 1)))
& ";" & Dec2Bin8(Asc(Mid(tKunci, 14, 1))) & ";" &
Dec2Bin8(Asc(Mid(tKunci, 15, 1))) & ";" & Dec2Bin8(Asc(Mid(tKunci, 16,
1))) & " = " & A3 & vbCrLf

        For i = 0 To 15
            SProses = SProses & "Putaran (" & i & ")Penjadwalan Kunci :
" & vbCrLf
            A0 = A0 Xor RC(i)
            SProses = SProses & "A0 = A0 Xor RC(" & i & ") = " & A0 &
vbCrLf
            Theta(A0, A1, A2, A3)
            A1 = Rotate_Left(A1, 1)
            A2 = Rotate_Left(A2, 5)
            A3 = Rotate_Left(A3, 2)
            SProses = SProses & "A1 = A1 << 1 = " & A1 & vbCrLf
            SProses = SProses & "A2 = A2 << 5 = " & A2 & vbCrLf
            SProses = SProses & "A3 = A3 << 2 = " & A3 & vbCrLf

            Gamma(A0, A1, A2, A3)

            A1 = Rotate_Right(A1, 1)
            A2 = Rotate_Right(A2, 5)
            A3 = Rotate_Right(A3, 2)
            SProses = SProses & "A1 = A1 >> 1 = " & A1 & vbCrLf
            SProses = SProses & "A2 = A2 >> 5 = " & A2 & vbCrLf
            SProses = SProses & "A3 = A3 >> 2 = " & A3 & vbCrLf
        Next

        A0 = A0 Xor RC(16)
        SProses = SProses & "A0 = A0 Xor RC(16) = " & A0 & vbCrLf
        ReDim DK(3)
        ReDim EK(3)

        DK(0) = A0
        DK(1) = A1
        DK(2) = A2
        DK(3) = A3
        SProses = SProses & "DK(0) = A0 = " & DK(0) & vbCrLf
        SProses = SProses & "DK(1) = A1 = " & DK(1) & vbCrLf
        SProses = SProses & "DK(2) = A2 = " & DK(2) & vbCrLf
        SProses = SProses & "DK(3) = A3 = " & DK(3) & vbCrLf

        Theta(A0, A1, A2, A3)

        EK(0) = A0
        EK(1) = A1
        EK(2) = A2
        EK(3) = A3
        SProses = SProses & "EK(0) = A0 = " & EK(0) & vbCrLf
        SProses = SProses & "EK(1) = A1 = " & EK(1) & vbCrLf
        SProses = SProses & "EK(2) = A2 = " & EK(2) & vbCrLf
```

```
        SProses = SProses & "EK(3) = A3 = " & EK(3) & vbCrLf
    End Sub

    Public Sub Enkripsi(ByVal tKunci As String, ByVal tInput As String,
ByVal worker As System.ComponentModel.BackgroundWorker, _
                        ByVal e As
System.ComponentModel.DoWorkEventArgs, ByRef tProgress As Integer)
        Dim i As Long
        Dim j As Long
        Dim A0 As UInteger
        Dim A1 As UInteger
        Dim A2 As UInteger
        Dim A3 As UInteger
        Dim OIndeks As Long

        GenerateRC()
        KeySchedule(tKunci)
        GenerateBlock(tInput)

        ReDim tOutput((LBlock.Count * 16) - 1)
        OIndeks = 0

        SProses = SProses & "Enkripsi : " & vbCrLf
        For i = 0 To LBlock.Count - 1
            A0 = LBlock(i).B(0)
            A1 = LBlock(i).B(1)
            A2 = LBlock(i).B(2)
            A3 = LBlock(i).B(3)

            SProses = SProses & "A0 : " & A0 & vbCrLf
            SProses = SProses & "A1 : " & A1 & vbCrLf
            SProses = SProses & "A2 : " & A2 & vbCrLf
            SProses = SProses & "A3 : " & A3 & vbCrLf

            For j = 0 To 15
                SProses = SProses & "Putaran (" & j & ") Enkripsi : " &
vbCrLf
                A0 = A0 Xor RC(j)
                SProses = SProses & "A0 = A0 Xor RC(" & j & ") = " & A0
& vbCrLf
                Theta(A0, A1, A2, A3, EK)

                A1 = Rotate_Left(A1, 1)
                A2 = Rotate_Left(A2, 5)
                A3 = Rotate_Left(A3, 2)
                SProses = SProses & "A1 = A1 << 1 = " & A1 & vbCrLf
                SProses = SProses & "A2 = A2 << 5 = " & A2 & vbCrLf
                SProses = SProses & "A3 = A3 << 2 = " & A3 & vbCrLf

                Gamma(A0, A1, A2, A3)

                A1 = Rotate_Right(A1, 1)
                A2 = Rotate_Right(A2, 5)
                A3 = Rotate_Right(A3, 2)
                SProses = SProses & "A1 = A1 >> 1 = " & A1 & vbCrLf
                SProses = SProses & "A2 = A2 >> 5 = " & A2 & vbCrLf
                SProses = SProses & "A3 = A3 >> 2 = " & A3 & vbCrLf
            Next

            A0 = A0 Xor RC(16)
```

```
            SProses = SProses & "A0 = A0 Xor RC(16) = " & A0 & vbCrLf
            Theta(A0, A1, A2, A3, EK)

            Store_B(tOutput, OIndeks, A0)
            Store_B(tOutput, OIndeks, A1)
            Store_B(tOutput, OIndeks, A2)
            Store_B(tOutput, OIndeks, A3)

            tProgress = (i / LBlock.Count) * 100
            Application.DoEvents()
        Next
    End Sub

    Public Sub Dekripsi(ByVal tKunci As String, ByVal tInput() As Byte,
ByVal worker As System.ComponentModel.BackgroundWorker, _
                        ByVal e As
System.ComponentModel.DoWorkEventArgs, ByRef tProgress As Integer)
        Dim i As Long
        Dim j As Long
        Dim A0 As UInteger
        Dim A1 As UInteger
        Dim A2 As UInteger
        Dim A3 As UInteger
        Dim OIndeks As Long

        GenerateRC()
        KeySchedule(tKunci)
        GenerateBlock2(tInput, tProgress)

        ReDim tOutput((LBlock.Count * 16) - 1)
        OIndeks = 0

        For i = 0 To LBlock.Count - 1
            A0 = LBlock(i).B(0)
            A1 = LBlock(i).B(1)
            A2 = LBlock(i).B(2)
            A3 = LBlock(i).B(3)

            SProses = SProses & "A0 : " & A0 & vbCrLf
            SProses = SProses & "A1 : " & A1 & vbCrLf
            SProses = SProses & "A2 : " & A2 & vbCrLf
            SProses = SProses & "A3 : " & A3 & vbCrLf

            For j = 16 To 1 Step -1
                SProses = SProses & "Putaran (" & j & ") Dekripsi : " &
vbCrLf
                Theta(A0, A1, A2, A3, DK)
                A0 = A0 Xor RC(j)
                SProses = SProses & "A0 = A0 Xor RC(" & j & ") = " & A0
& vbCrLf

                A1 = Rotate_Left(A1, 1)
                A2 = Rotate_Left(A2, 5)
                A3 = Rotate_Left(A3, 2)
                SProses = SProses & "A1 = A1 << 1 = " & A1 & vbCrLf
                SProses = SProses & "A2 = A2 << 5 = " & A2 & vbCrLf
                SProses = SProses & "A3 = A3 << 2 = " & A3 & vbCrLf

                Gamma(A0, A1, A2, A3)
```

```
                A1 = Rotate_Right(A1, 1)
                A2 = Rotate_Right(A2, 5)
                A3 = Rotate_Right(A3, 2)
                SProses = SProses & "A1 = A1 >> 1 = " & A1 & vbCrLf
                SProses = SProses & "A2 = A2 >> 5 = " & A2 & vbCrLf
                SProses = SProses & "A3 = A3 >> 2 = " & A3 & vbCrLf
            Next

            Theta(A0, A1, A2, A3, DK)
            A0 = A0 Xor RC(0)
            SProses = SProses & "A0 = A0 Xor RC(0) = " & A0 & vbCrLf

            Store_B(tOutput, OIndeks, A0)
            Store_B(tOutput, OIndeks, A1)
            Store_B(tOutput, OIndeks, A2)
            Store_B(tOutput, OIndeks, A3)

            tProgress = (i / LBlock.Count) * 100
            Application.DoEvents()
        Next
    End Sub

    Private Sub Store_B(ByRef tOutput() As Byte, ByRef tIndeks As Long,
ByVal tInput As UInteger)
        Dim Temp1 As String
        Dim Temp2 As String

        Temp1 = Dec2Bin(tInput)
        SProses = SProses & "Store Output : " & vbCrLf
        tOutput(tIndeks) = Bin2Dec(Mid(Temp1, 1, 8))
        SProses = SProses & "Output(" & tIndeks & ") : " &
Bin2Dec(Mid(Temp1, 1, 8)) & vbCrLf
        tIndeks += 1

        Temp2 = Mid(Temp1, 9, 8)
        tOutput(tIndeks) = Bin2Dec(Temp2)
        SProses = SProses & "Output(" & tIndeks & ") : " &
Bin2Dec(Mid(Temp1, 9, 8)) & vbCrLf
        tIndeks += 1

        tOutput(tIndeks) = Bin2Dec(Mid(Temp1, 17, 8))
        SProses = SProses & "Output(" & tIndeks & ") : " &
Bin2Dec(Mid(Temp1, 17, 8)) & vbCrLf
        tIndeks += 1

        tOutput(tIndeks) = Bin2Dec(Mid(Temp1, 25, 8))
        SProses = SProses & "Output(" & tIndeks & ") : " &
Bin2Dec(Mid(Temp1, 25, 8)) & vbCrLf
        tIndeks += 1
    End Sub

    Public Function Rotate_Left(ByVal tInput As UInteger, ByVal tCount
As Integer) As UInteger
        Dim i As Integer
        Dim TS As String

        TS = Dec2Bin(tInput)
        For i = 1 To tCount
            TS = Mid(TS, 2) & Mid(TS, 1, 1)
        Next
```

```
        Return Bin2Dec(TS)
    End Function

    Public Function Rotate_Left2(ByVal tInput As UInteger, ByVal tCount
As Integer) As UInteger
        Dim i As Integer
        Dim TS As String

        TS = Dec2Bin8(tInput)
        For i = 1 To tCount
            TS = Mid(TS, 2) & Mid(TS, 1, 1)
        Next

        Return Bin2Dec(TS)
    End Function

    Public Function Rotate_Right(ByVal tInput As UInteger, ByVal tCount
As Integer) As UInteger
        Dim i As Integer
        Dim TS As String

        TS = Dec2Bin(tInput)
        For i = 1 To tCount
            TS = Mid(TS, 32) & Mid(TS, 1, 31)
        Next

        Return Bin2Dec(TS)
    End Function

    Private Function Dec2Bin(ByVal tValue As UInteger) As String
        Return Convert.ToString(tValue, 2).PadLeft(32, "0"c)
    End Function

    Private Function Dec2Bin8(ByVal tValue As UInteger) As String
        Return Convert.ToString(tValue, 2).PadLeft(8, "0"c)
    End Function

    Private Function Bin2Dec(ByVal tValue As String) As UInteger
        Dim BinaryNum As Long
        Dim BitCount As Short

        For BitCount = 1 To Len(tValue)
            BinaryNum = BinaryNum + (CLng(Mid(tValue, Len(tValue) -
BitCount + 1, 1)) * (2 ^ (BitCount - 1)))
        Next BitCount
        Return BinaryNum
    End Function
End Class
```

**RSA**

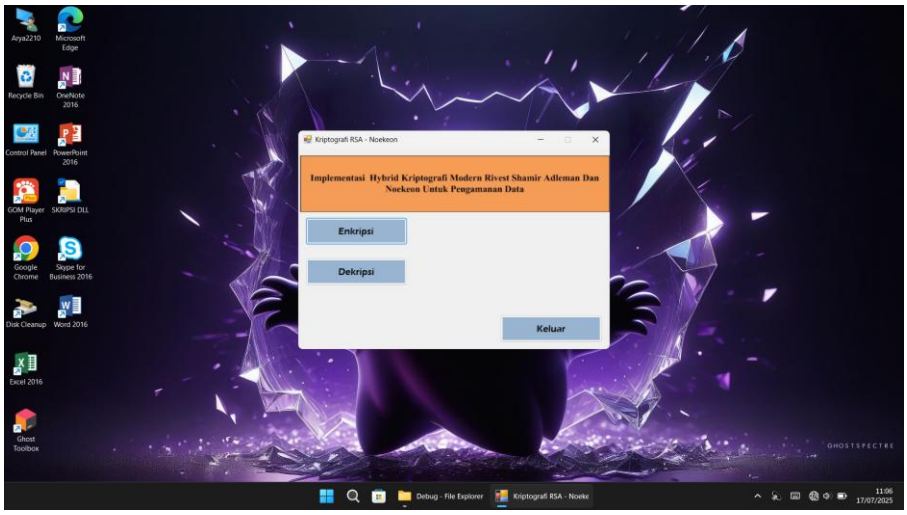
```
Public Class ClRSA
    Public Class AngkaKripto
        Public Angka As Long
    End Class

    Public PublicKey As String
    Public PrivateKey As String

    Public LAngkaKripto As List(Of AngkaKripto)
    Public tInput() As Byte
    Public tInput2() As Long
    Public Lapor As String

    Public Sub BangkitkanKunci()
        Dim p As Long
        Dim q As Long
        Dim n As Long
        Dim tot As Long
        Dim e As Long
        Dim d As Long

        p = Now.Second
        q = Now.Minute

        While p = 1
            p = Now.Second
        End While

        If q = 1 Then
            q += 1
        End If

        Do While Not PrimeNumberCheck(p)
            p += 1
        Loop

        Do While Not PrimeNumberCheck(q)
            q += 1
        Loop

        Lapor = "Memilih p bilangan prima secara acak : " & p
        Lapor &= vbCrLf & "Memilih q bilangan prima secara acak : " & q

        n = p * q

        Lapor &= vbCrLf & "Menghitung n = (p*q) = " & n

        tot = (p - 1) * (q - 1)

        Lapor &= vbCrLf & "Menghitung m = (p-1) * (q-1) = " & tot

        Lapor &= vbCrLf & "Menghitung nilai e yang memenuhi GCD(e,m)
atau pembagi sama yang terbesar = 1"
        e = 2
        Do While GCD(e, tot) <> 1
```

```
            Lapor &= vbCrLf & "e = " & e & ", GCD(" & e & "," & tot &
") = " & GCD(e, tot) & " tidak memenuhi"
            e += 1
        Loop
        Lapor &= vbCrLf & "e = " & e & ", GCD(" & e & "," & tot & ") =
" & GCD(e, tot) & " memenuhi"

        Lapor &= vbCrLf & "Menghitung nilai d yang memenuhi (d x e) mod
m = 1"
        d = 2
        Do While (((d * e) Mod tot) <> 1)
            Lapor &= vbCrLf & "d = " & d & ", (" & d & "*" & e & ") mod
" & tot & " = " & (d * e) Mod tot & " tidak memenuhi"
            d += 1
        Loop
        Lapor &= vbCrLf & "d = " & d & ", (" & d & "*" & e & ") mod " &
tot & " = " & (d * e) Mod tot & " memenuhi"

        Lapor &= vbCrLf & "diperoleh kunci public (e,n) = (" & e & "-"
& n & ") dan kunci private (d,n) = (" & d & "-" & n & ")"
        PublicKey = e & "-" & n
        PrivateKey = d & "-" & n
    End Sub

    Public Sub Enkripsi(ByVal worker As
System.ComponentModel.BackgroundWorker, _
                        ByVal e As
System.ComponentModel.DoWorkEventArgs, ByRef tProgress As Integer)

        Dim tAngkaEnkripsi As AngkaKripto
        Dim i As Long
        Dim kunci As Long
        Dim modulo As Long
        Dim ArrStr() As String

        ArrStr = PublicKey.Split("-")

        kunci = Val(ArrStr(0))
        modulo = Val(ArrStr(1))

        LAngkaKripto = New List(Of AngkaKripto)
        Lapor = "Enkripsi RSA : " & vbCrLf
        For i = 0 To tInput.Length - 1
            tAngkaEnkripsi = New AngkaKripto
            tAngkaEnkripsi.Angka = Pangkat(tInput(i), kunci, modulo)
            Lapor &= "Ch[" & i & "] = " & tInput(i) & " ^ " & kunci & "
Mod " & modulo & " = " & tAngkaEnkripsi.Angka & vbCrLf
            LAngkaKripto.Add(tAngkaEnkripsi)
            tProgress = (i / tInput.Length) * 100
        Next
    End Sub

    Public Sub Dekripsi(ByVal worker As
System.ComponentModel.BackgroundWorker, _
                        ByVal e As
System.ComponentModel.DoWorkEventArgs, ByRef tProgress As Integer)

        Dim tAngkaEnkripsi As AngkaKripto
        Dim i As Long
        Dim kunci As Long
```

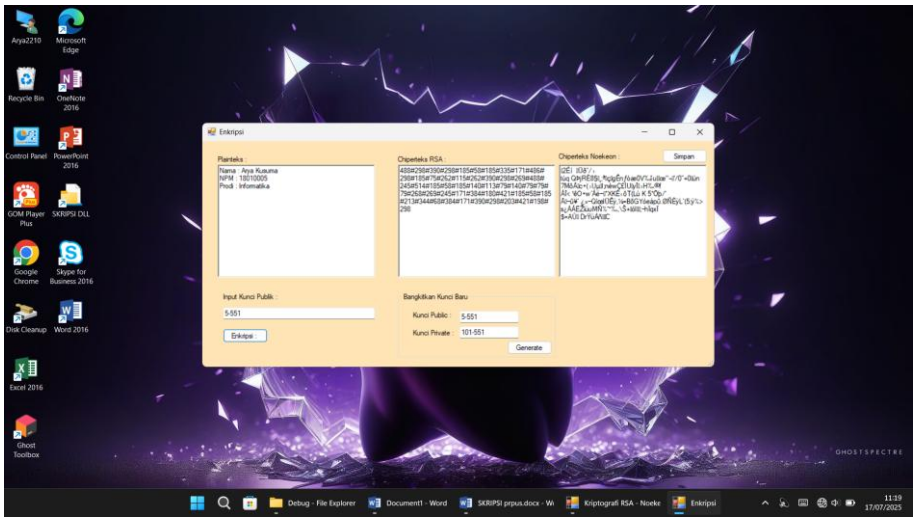
```
        Dim modulo As Long
        Dim ArrStr() As String

        ArrStr = PrivateKey.Split("-")

        kunci = Val(ArrStr(0))
        modulo = Val(ArrStr(1))

        LAngkaKripto = New List(Of AngkaKripto)
        Lapor = "Enkripsi RSA : " & vbCrLf
        For i = 0 To tInput2.Length - 1
            tAngkaEnkripsi = New AngkaKripto
            tAngkaEnkripsi.Angka = Pangkat(tInput2(i), kunci, modulo)
            Lapor &= "P[" & i & "] = " & tInput2(i) & " ^ " & kunci & "
Mod " & modulo & " = " & tAngkaEnkripsi.Angka & vbCrLf
            LAngkaKripto.Add(tAngkaEnkripsi)
            tProgress = (i / tInput2.Length) * 100
        Next
    End Sub

    Public Function Pangkat(ByVal tAngka As Long, ByVal tPangkat As
Long, ByVal tMod As Long) As Long
        Dim Tamp As Long
        Dim i As Long

        Tamp = tAngka
        For i = 2 To tPangkat
            Tamp = (Tamp * tAngka) Mod tMod
        Next

        Return Tamp
    End Function

    Public Function GCD(ByVal a As Long, ByVal b As Long) As Long
        Dim tmp As Long

        Do
            tmp = a Mod b
            a = b
            b = tmp
        Loop While b > 0
        Return a
    End Function

    Public Function PrimeNumberCheck(ByVal number As Integer) As
Boolean

        Dim primeI As Integer
        Dim primeFlag As Boolean

        primeFlag = True
        For primeI = 2 To number / 2
            If number Mod primeI = 0 Then
                Return False
            End If
        Next

        Return primeFlag
    End Function
End Class
```

## 2. Tampilan Aplikasi

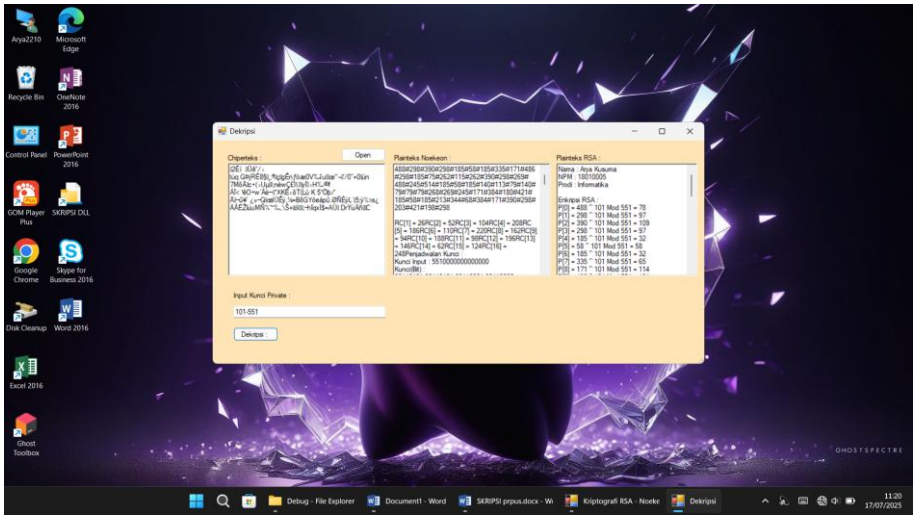
### - Halaman Depan




### - Halaman Enkripsi



- **Halaman Dekripsi**




- **Bentuk Data Terenkripsi**