BAB III
METODE PENELITIAN

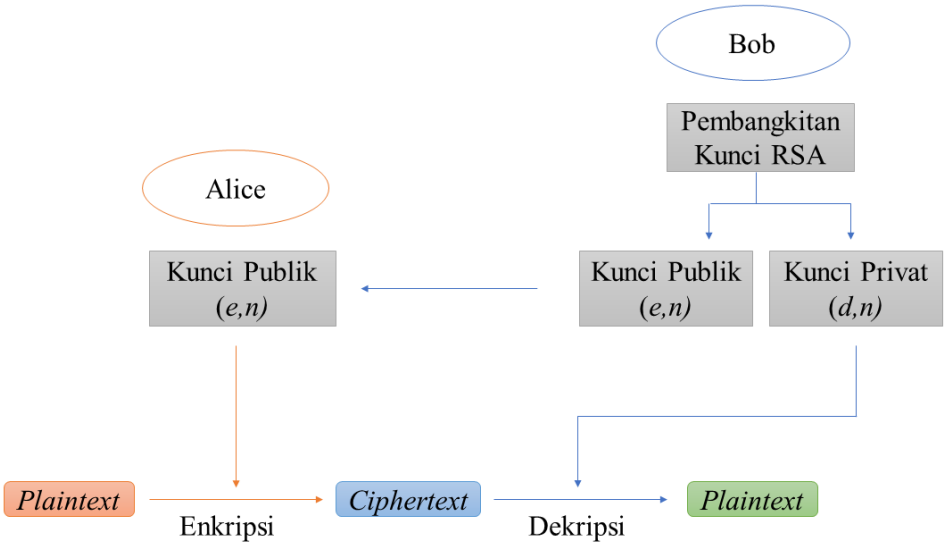
3.1 Identifikasi Masalah

Email adalah salah satu alat komunikasi yang banyak digunakan, tetapi juga memiliki risiko serangan cyber dan kebocoran data. Serangan seperti *phishing* dan penggunaan alat-alat baru seperti *hyperscrape* yang telah terjadi, mengancam keamanan pengguna *email*. Selain itu, ketika membuat akun di platform *online*, seringkali persetujuan privasi terabaikan oleh pengguna dan memungkinkan akses pihak ketiga ke pesan *email*. Oleh karena itu, penting untuk menerapkan tindakan pengamanan, seperti kriptografi, untuk melindungi pesan-pesan rahasia yang dikirim melalui *email*. Kriptografi adalah salah satu cara yang dapat digunakan untuk mengamankan suatu pesan rahasia. Dengan menggunakan teknik enkripsi dan dekripsi serta kunci yang tepat, kriptografi mengubah pesan menjadi kode yang tidak dapat dimengerti (cipherteks) dan hanya dapat diubah kembali menjadi pesan asli (plainteks) oleh penerima yang dituju.

Dalam penelitian ini, akan digunakan penggabungan algoritma kriptografi yang menggabungkan keunggulan dari algoritma RSA yang ditingkatkan dengan algoritma AES. Algoritma AES dikenal karena kekuatan dan efisiensinya dalam melindungi data dengan ukuran yang lebih besar, sementara algoritma RSA terkenal karena kemampuannya dalam menjaga keamanan kunci publik dan pribadi serta melakukan proses enkripsi dan dekripsi yang aman.

3.2 Model Dasar

Skema algoritma kriptografi RSA berdasarkan yang dipaparkan dalam BAB II bagian 2.6 akan dtunjukkan dalam Gambar 3.1.

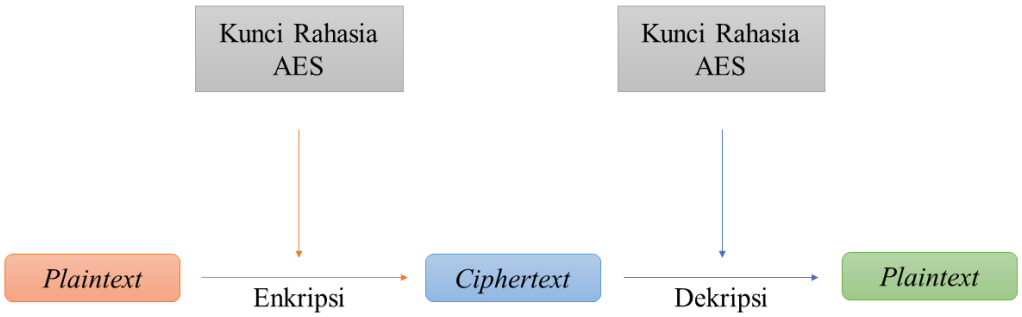


Gambar 3.1 Skema Algoritma Kriptografi RSA

Pada skema dalam Gambar 3.1 di atas memperlihatkan bahwa sebelum melakukan proses enkripsi dan dekripsi, harus dibangkitkan dulu suatu pasangan kunci publik dan kunci privat. Bob terlebih dahulu memilih bilangan p dan q. Kemudian bob menghitung nilai n dan $\phi(n)$. Bob memilih kunci publik e, yang relatif prima terhadap $\phi(n)$, kemudian membangkitkan kunci privat dengan persamaan $e \cdot d \equiv 1 \; mod \; \phi(n)$. Diperoleh pasangan kunci publik (e, n) dan kunci privat (d, n). Langkah selanjutnya, Bob mengirimkan kunci publik kepada Alice. Alice menggunakan kunci publik tersebut untuk mengenkripsi plainteks yang hendak dikirimkan. Cipherteks yang diperoleh dari proses enkripsi plainteks menggunakan kunci publik RSA dikirimkan ke Bob, lalu Bob dapat mendekripsi cipherteks tersebut dengan kunci privat yang dimiliki.

Keunggulan utama dari RSA adalah bahwa algoritma ini memungkinkan pertukaran pesan secara aman tanpa perlu pertukaran kunci rahasia terlebih dahulu. RSA juga tahan terhadap serangan *brute-force* karena kompleksitas matematika yang terlibat dalam proses enkripsi dan dekripsi. Namun, RSA juga memiliki kelemahan. Proses enkripsi dan dekripsi dalam RSA cenderung memakan waktu yang lama, terutama ketika digunakan untuk mengenkripsi atau mendekripsi data dengan ukuran yang besar. Oleh karena itu, RSA umumnya digunakan untuk pertukaran kunci dan pengamanan data yang relatif kecil, sedangkan algoritma simetris seperti AES digunakan untuk mengenkripsi data dalam skala besar.

Adapun untuk proses enkripsi dan dekripsi menggunakan algoritma AES yang telah dipaparkan pada BAB II bagian 2.7 akan ditunjukkan dengan skema dalam Gambar 3.2.



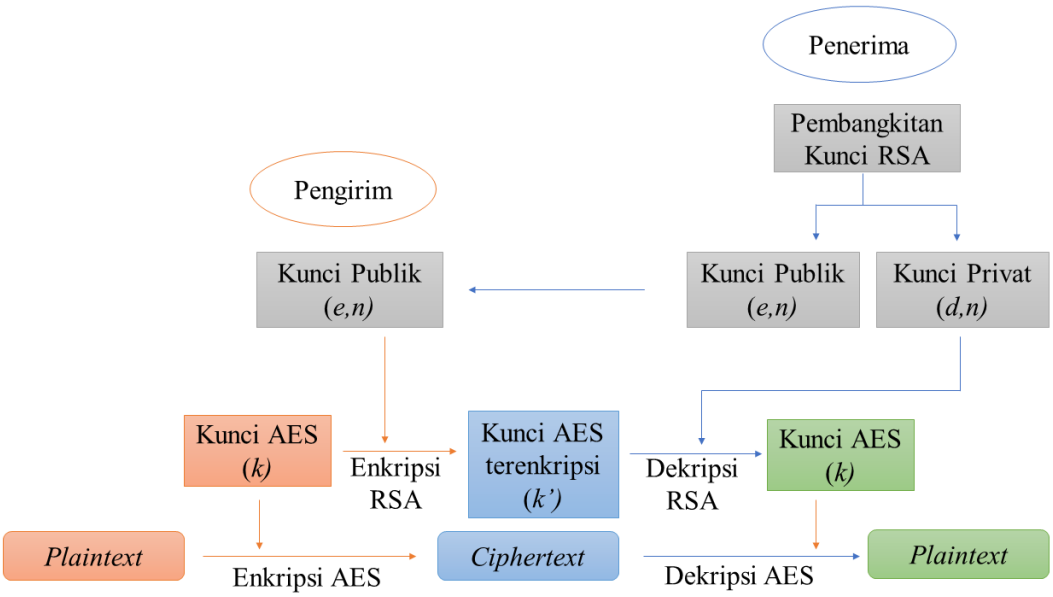
Gambar 3.2 Skema Algoritma Kriptografi AES

Pada algoritma kriptografi AES, pengirim melakukan proses enkripsi AES dengan kunci dan plainteks yang diinginkan, kemudian cipherteks dan kunci rahasia tersebut dikirimkan kepada penerima untuk didekripsi. Seperti yang telah dijelaskan sebelumnya, proses enkripsi dalam AES melibatkan transformasi data secara berulang menggunakan serangkaian putaran (*round*s) yang melibatkan substitusi, pergeseran, dan pencampuran bit., yang membuatnya sangat sulit untuk dipecahkan tanpa pengetahuan kunci yang tepat. Selain itu, AES juga menerapkan operasi *bitwise*, yaitu operasi pada level bit (1 dan 0) dalam representasi biner dari data, memungkinkan manipulasi langsung terhadap bit-bit tersebut. Operasi *bitwise* memungkinkan implementasi perangkat keras khusus yang dioptimalkan untuk AES, karena perangkat keras dapat di-desain untuk melakukan operasi *bitwise* secara paralel dengan kecepatan tinggi. Namun, salah satu kekhawatiran yang muncul adalah ketika kunci rahasia AES harus ditransmisikan dari satu pihak ke pihak lain secara aman. Jika kunci tersebut jatuh ke tangan yang salah atau disadap oleh pihak yang tidak berwenang, maka keamanan data yang dienkripsi menggunakan kunci tersebut dapat terancam.

3.3 Pengembangan Model Dasar

Implementasi penggabungan algoritma kriptografi RSA yang ditingkatkan dan AES bertujuan untuk meningkatkan keamanan pada pesan rahasia. Dari permasalahan transmisi kunci yang ditemukan saat implementasi algoritma AES, maka algoritma kriptografi RSA akan diterapkan agar pertukaran kunci AES lebih aman. Selain itu, karena algoritma kriptografi RSA memiliki tahap-tahap komputasi

yang berat, algoritma ini akan lebih cocok diterapkan pada kunci karena kunci yang akan digunakan berukuran tetap yaitu 16 *bytes* (AES-128). Penggabungan kedua algoritma kriptografi ini disebut juga dengan kriptografi *hybrid*. Lebih lanjut, cara implementasi kriptografi *hybrid* ini ditunjukkan dalam skema pada Gambar 3.3.



Gambar 3.3 Skema Penggabungan Algoritma RSA yang Ditingkatkan dan AES

Dalam implementasi pengembangan modelnya, penerima membangkitkan kunci dengan algoritma RSA yang ditingkatkan yaitu dengan cara menambahkan satu bilangan prima pada prosesnya, hingga diperoleh kunci publik dan kunci privat. Kemudian, kunci publik tersebut dikirimkan kepada pihak yang akan mengirim pesan. Pengirim pesan terlebih dahulu mengenkripsi pesan yang akan dikirimkan menggunakan algoritma AES menjadi cipherteks, kemudian kunci yang digunakan untuk enkripsi AES akan dienkripsi oleh algoritma RSA yang ditingkatkan menggunakan kunci publik dari penerima. Kemudian, kunci AES yang telah dienkripsi dan cipherteks yang diperoleh dikirimkan ke pihak penerima. Selanjutnya, pihak penerima terlebih dulu mendekripsi kunci AES yang terenkripsi dengan menggunakan kunci privat yang dimiliki, kemudian setelah terdekripsi, kunci AES tersebut digunakan untuk mendekripsi cipherteks.

3.4 Konstruksi Program Aplikasi

Pembuatan program aplikasi dari algoritma kombinasi hasil penggabungan algoritma RSA yang ditingkatkan dengan algoritma AES dilakukan dengan cara

mengubah model matematis algoritma kombinasi ke dalam bahasa pemrograman. Program yang digunakan adalah *Python*. Program ini akan memiliki beberapa pilihan menu mulai dari enkripsi, dekripsi hingga pengiriman pesan melalui *email*.

3.4.1 Input Output

Aplikasi yang akan dibuat memiliki fungsi untuk melakukan pembangkitan kunci, enkripsi atau dekripsi dan pengiriman *email*. Kemudian, data atau pesan berupa teks yang diinputkan secara manual ke dalam aplikasi pengirim *email*.

Program aplikasi yang akan dikonstruksi memiliki tampilan awal (menu utama) yang memiliki tiga pilihan yaitu, pembangkitan kunci RSA, enkripsi, dekripsi, dan pengiriman *email*. Input dari pembangkitan kunci RSA adalah bilangan prima *p, q, r*. Output dari pembangkitan kunci diperoleh kunci privat dan kunci publik. Input dari enkripsi pesan adalah plainteks, kunci AES, dan kunci publik RSA. Output dari enkripsi pesan adalah cipherteks dan kunci AES yang terenkripsi. Input dari dekripsi pesan adalah cipherteks, dan kunci privat RSA, dan kunci AES yang terenkripsi. Output dari dekripsi pesan adalah plainteks. Input dari pengiriman *email* meliputi *email* pengirim, password *email* pengirim, *email* tujuan, subjek *email*, dan pesan *email*. Output dari pengiriman *email* yaitu tulisan "Pesan Terkirim".

Tabel 3.1
Rancangan Input dan Output pada Program

	Input	Output
Pembangkitan Kunci RSA	- Bilangan prima *p* - Bilangan prima *q* - Bilangan prima *r*	- Kunci Publik RSA (*e,n*) - Kunci Privat RSA (*d,n*)
Enkripsi Pesan	- Plainteks - Kunci AES - Kunci Publik RSA	- Cipherteks - Kunci AES terenkripsi
Dekripsi Pesan	- Cipherteks - Kunci AES terenkripsi - Kunci Privat RSA	- Plainteks

3.4.2 Algoritma Deskriptif

Perbedaan algoritma RSA standar dengan algoritma RSA yang ditingkatkan terdapat pada proses pembangkitkan kunci. Proses pembangkitan kunci, enkripsi, dan dekripsi menggunakan penggabungan algoritma RSA yang ditingkatkan dan AES adalah sebagai berikut.

a. Pembangkitan Kunci

Proses pembangkitan kunci menggunakan algoritma RSA yang ditingkatkan adalah sebagai berikut.

1. Pilih tiga bilangan prima p_1, p_2, dan p_3.
2. Hitung $n = p_1 \times p_2 \times p_3$ ($p_i \neq p_j$, $i \neq j$ agar n tidak mudah difaktorkan).
3. Hitung $\phi(n) = (p_1 - 1) \times (p_2 - 1) \times (p_3 - 1)$.
4. Pilih kunci publik e, yang relatif prima terhadap $\phi(n)$.
5. Bangkitkan d yang memenuhi $e.\ d = 1\ mod\ \phi(n)$.

Pada program *python* yang akan dirancang, penerima hanya perlu memasukkan tiga bilangan prima yang dipilih lalu diperoleh kunci publik dan kunci privat.

b. Enkripsi

Dilanjutkan dengan proses enkripsi dan dekripsi dengan perhitungan yang telah dijelaskan pada BAB II bagian 2.5 dan 2.6, langkah-langkahnya sebagai berikut:

1. Pengirim menerima kunci publik dari penerima.
2. Pengirim mengenkripsi pesan yang akan dikirimkan dengan algoritma AES.
3. Pengirim mendapatkan cipherteks dari hasil enkripsi plainteks menggunakan AES, dan kunci AES yang terenkripsi oleh RSA.
4. Pengirim mengirimkan cipherteks dan kunci AES yang terenkripsi kepada penerima..

c. Dekripsi

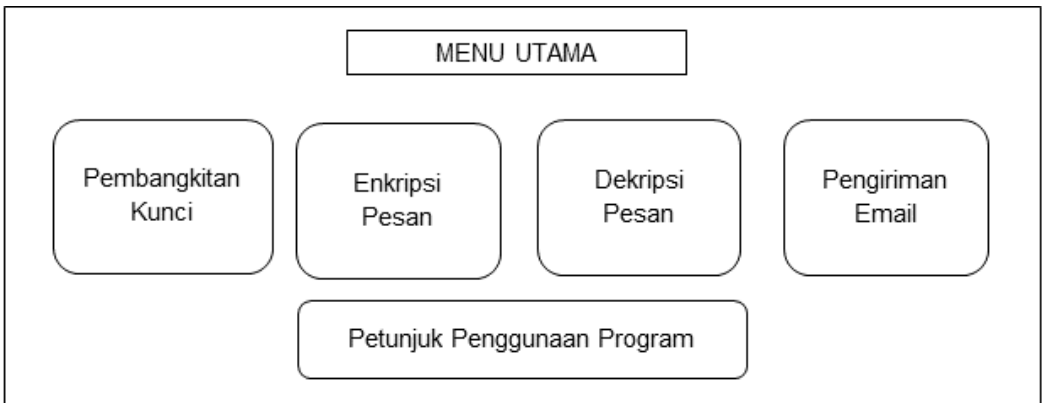
Kemudian pada proses dekripsi, penerima melakukan langkah-langkah berikut:

1. Penerima mendapatkan cipherteks dan kunci AES yang terenkripsi dari pengirim.

2. Penerima memiliki kunci privat dari tahap pembangkitan kunci yang sebelumnya telah dilakukan.
3. Penerima mendekripsi kunci AES yang terenkripsi menggunakan kunci privat yang dimiliki, menghasilkan kunci AES.
4. Penerima menggunakan kunci AES tersebut untuk mendekripsi cipherteks.

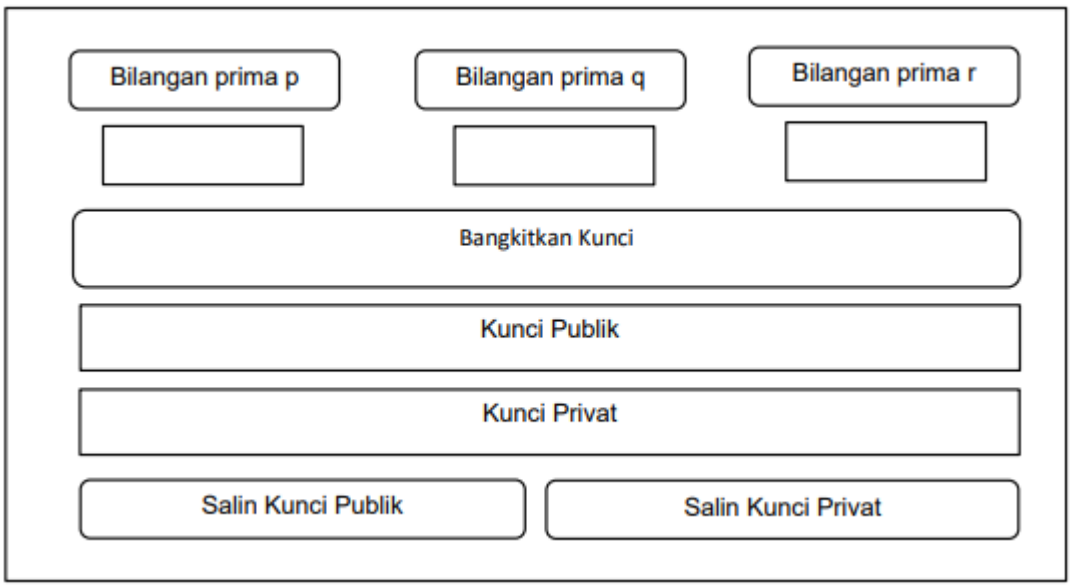
3.4.3 Rancangan Tampilan Program

a. Menu Utama



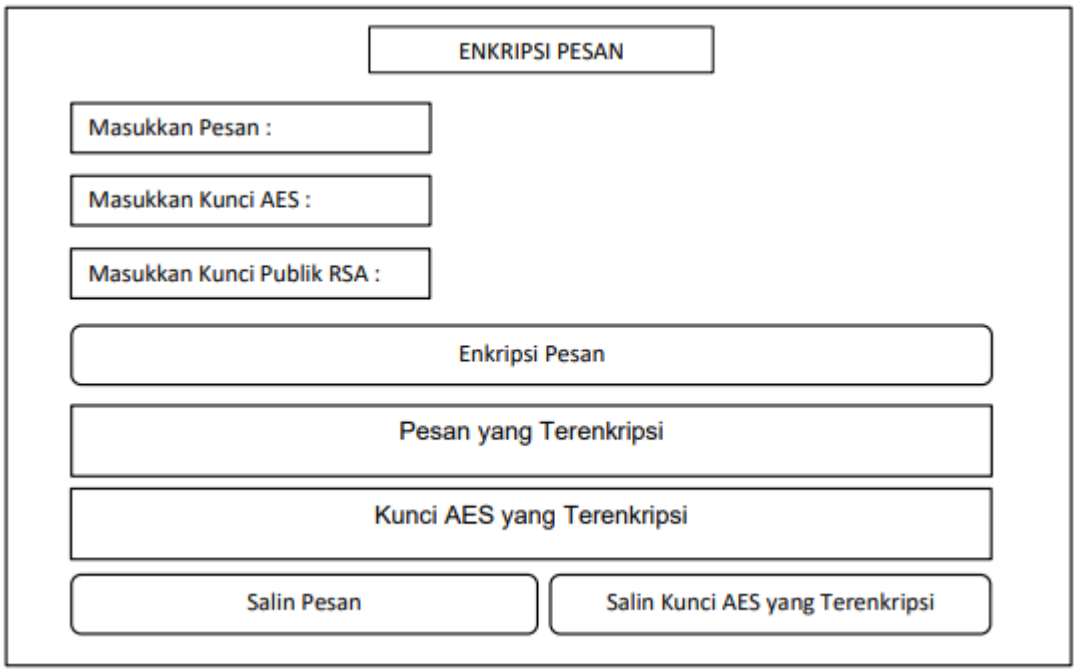
Gambar 3.4 Rancangan Tampilan Menu Utama

b. Pembangkitan Kunci



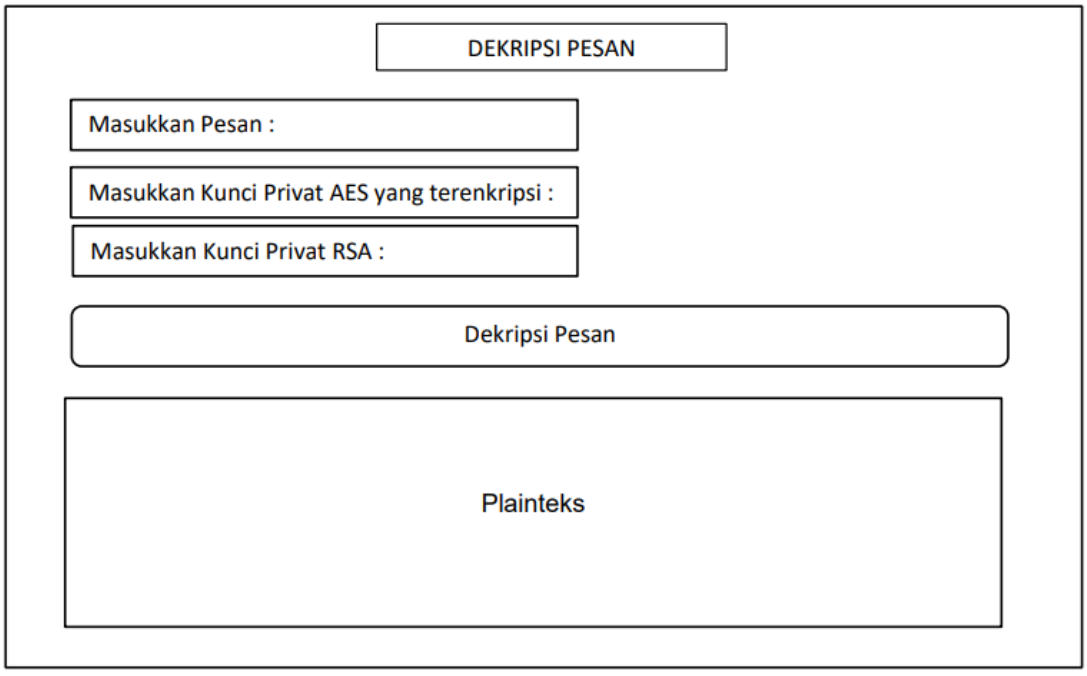
Gambar 3.5 Rancangan Tampilan Pembangkitan Kunci

c. Enkripsi Pesan



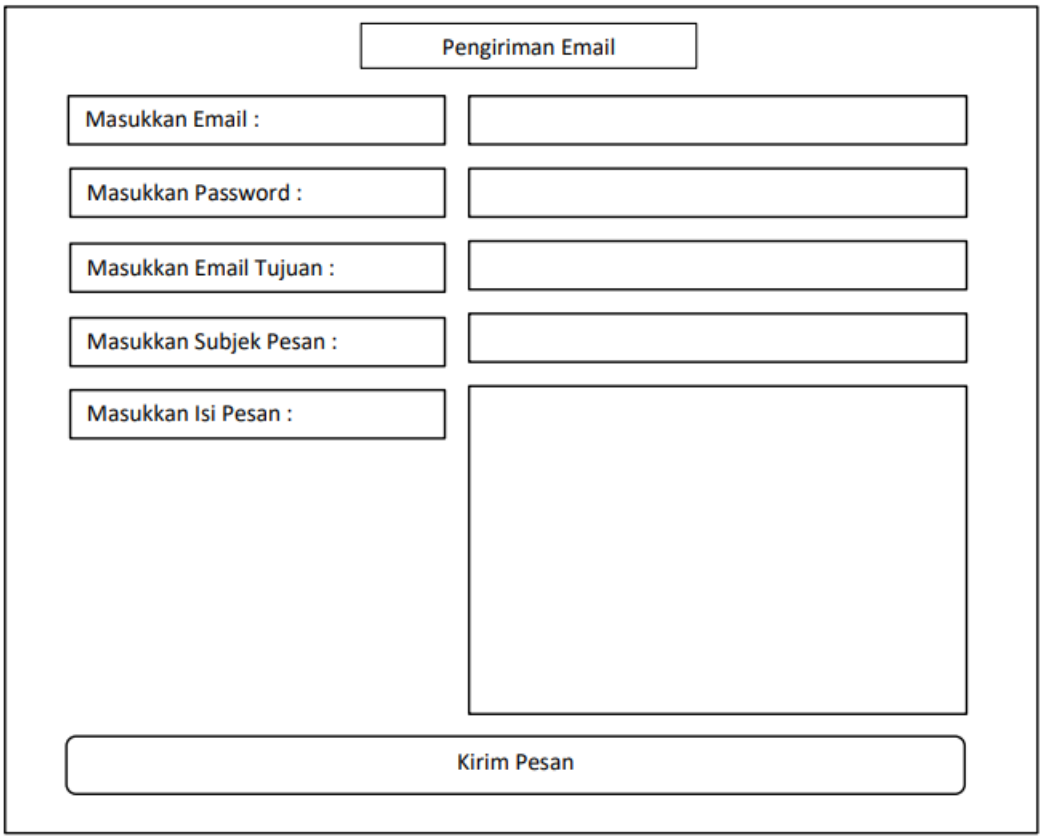
Gambar 3.6 Rancangan Tampilan Enkripsi Pesan

d. Dekripsi Pesan



Gambar 3.7 Rancangan Tampilan Dekripsi Pesan

e. Pengiriman *Email*



Gambar 3.8 Rancangan Tampilan Pengiriman *Email*

3.5 Penggunaan Program *Python*

Dalam penggunaannya, *library python* yang akan digunakan untuk menunjang pembuatan aplikasi adalah sebagai berikut:

1. *Smtplib*

   Modul *smtplib* digunakan untuk mengirim *email* melalui Simple *Mail* Transfer Protocol (SMTP). Modul ini akan membuat koneksi ke server SMTP dan mengirim *email* menggunakan akun *email* yang terautentikasi.

2. *Pycryptodome*

   Pycryptodome (juga dikenal sebagai *pycryptodomex*) adalah suatu *library python* yang menyediakan berbagai fungsi kriptografi.

3. *Tkinter*

   Tkinter digunakan untuk membuat aplikasi dengan grafis antarmuka (GUI). *Library* ini menyediakan berbagai alat untuk membangun GUI seperti label, tombol, kotak teks, dan lainnya

3.6 Proses Validasi

Tahap validasi dilakukan untuk memastikan program hasil implementasi berjalan dengan benar. Proses validasi dilakukan dengan cara melakukan perhitungan manual pada proses pembangkitan kunci, enkripsi, dan dekripsi.

Proses validasi pembangkitan kunci terdiri dari perhitungan manual kunci publik dan kunci privat, kemudian pada validasi enkripsi pesan dilakukan perhitungan manual untuk mendapatkan pesan dan kunci yang terenkripsi, sedangkan pada validasi dekripsi pesan dilakukan perhitungan manual untuk mendapatkan kunci dan pesan yang terdekripsi.

3.7 Penarikan Kesimpulan

Setelah program aplikasi tervalidasi, maka program sudah dipastikan berjalan dengan benar, algoritma kombinasi ini kemudian dapat digunakan dalam penyandian pesan dan program yang telah dibuat dapat digunakan sebagai implementasi.

Dengan ditingkatkannya algoritma RSA dan digabungkan dengan algoritma AES, diharapkan dapat mempermudah dalam menyandikan pesan serta menambah kesulitan penyadap dalam memecahkan pesan rahasia. Implementasi penggabungan algoritma RSA yang ditingkatkan dan AES pada *email* dapat menjadi salah satu cara agar penggunaan *email* sebagai sarana komunikasi lebih aman.