BAB III

METODOLOGI PENELITIAN

Pada penelitian ini dilakukan studi literatur untuk mengidentifikasi masalah, pengembangan model dasar, kemudian implementasi model ke dalam suatu program dengan menggunakan *GUI Python*. Berikut merupakan langkah-langkah yang dilakukan dalam menyelesaikan penelitian:

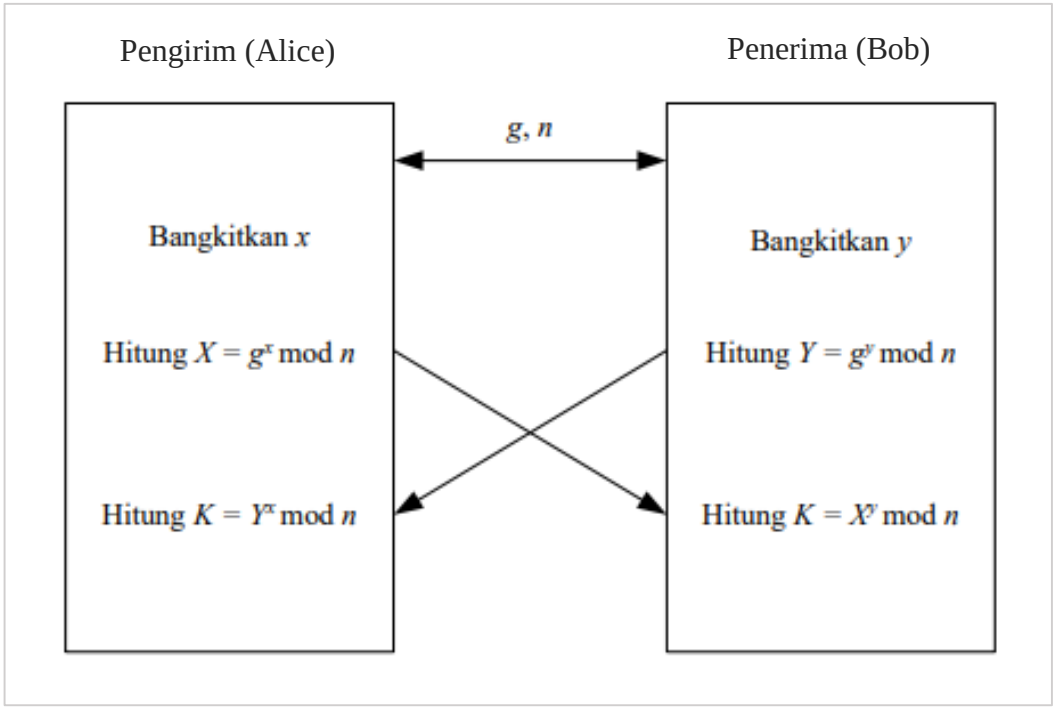
3. 1 Identifikasi Masalah

Penyimpanan data dalam komputer sudah sering dilakukan karena kemudahan untuk mengakses kembali dan mentransfer data. Salah satu program yang sering digunakan untuk menyimpan data adalah *Microsoft Excel*, yaitu program aplikasi menjadi andalan bagi kalangan professional. Di antara begitu banyaknya data yang tersimpan, terdapat berbagai data rahasia yang tidak boleh sampai diretas oleh pihak yang tidak diinginkan. Akan tetapi, kebocoran data berbentuk *file excel* ternyata masih terjadi di Indonesia yaitu pada awal September 2022 lalu. Saat itu, *Hacker* bernama *Bjorka* memberikan sampel 1.048.576 data pemilih KPU dari berbagai provinsi dalam bentuk *file excel*.

File excel yang telah digunakan banyak orang ternyata tidak luput dari ancaman kebocoran data, maka dapat dilakukan usaha meningkatkan keamanan data dengan bantuan kriptografi. Untuk mengurangi peluang bocornya data, penelitian ini mengaplikasikan dua algoritma kriptografi, yaitu pertukaran kunci menggunakan algoritma *Diffie-Hellman* dan proses enkripsi serta dekripsi *file excel* menggunakan algoritma *RSA*. Algoritma *Diffie-Hellman* membutuhkan perhitungan logaritma diskrit yang sulit untuk dipecahkan. Sedangkan, letak kekuatan algoritma RSA berada pada sulitnya memfaktorkan bilangan menjadi faktor-faktor prima yang digunakan pada proses mengubah plainteks menjadi cipherteks dan sebaliknya.

3. 2 Model Dasar

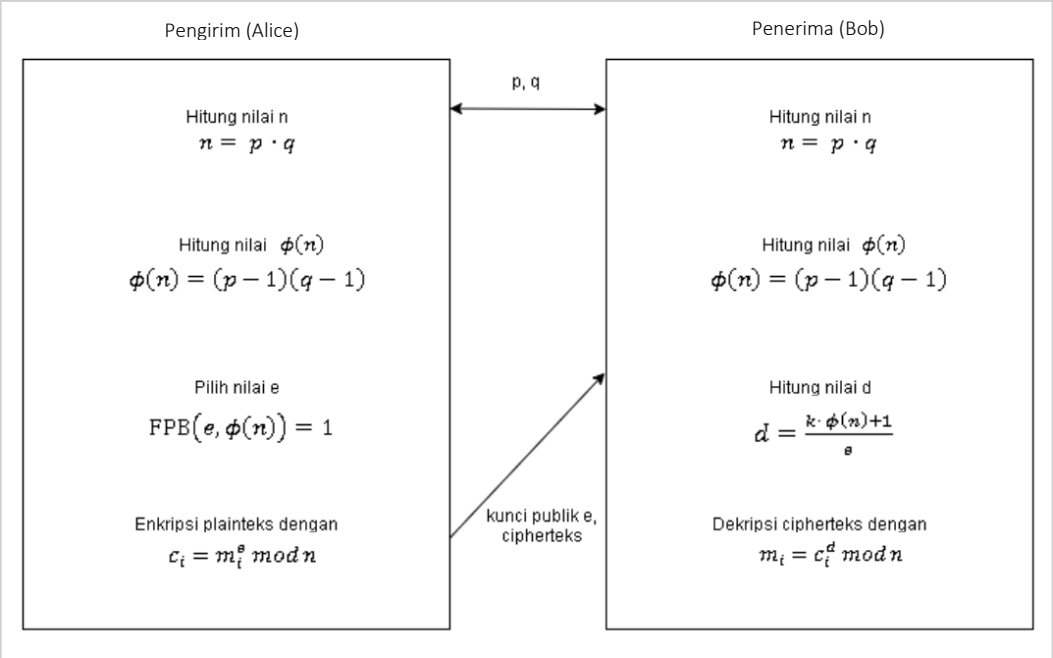
Skema pertukaran kunci *Diffie-Hellman* berdasarkan algoritma yang telah dipaparkan pada BAB II bagian 2.4 ditunjukkan pada Gambar 3.1 berikut:



Gambar 3. 1 Skema Pertukaran Kunci Diffie-Hellman

Skema pada Gambar 3.1 tersebut memperlihatkan bahwa Alice dan Bob menyepakati bilangan g dan n. Alice kemudian membangkitkan x, menghitung nilai X dan mengirimkan nilai X itu ke Bob, lalu menghitung kunci K dengan nilai Y yang diterimanya dari Bob. Sedangkan Bob membangkitkan y, menghitung nilai Y kemudian mengirimkan nilai Y itu ke Alice, dan menghitung kunci K dengan nilai X yang diterimanya dari Alice. Jika perhitungan benar, nilai kunci K milik Alice akan sama dengan nilai kunci K milik Bob.

Adapun untuk proses enkripsi dan dekripsi pesan menggunakan algoritma RSA yang telah dijelaskan pada BAB II bagian 2.5 akan ditunjukkan dengan skema pada Gambar 3.2 berikut:

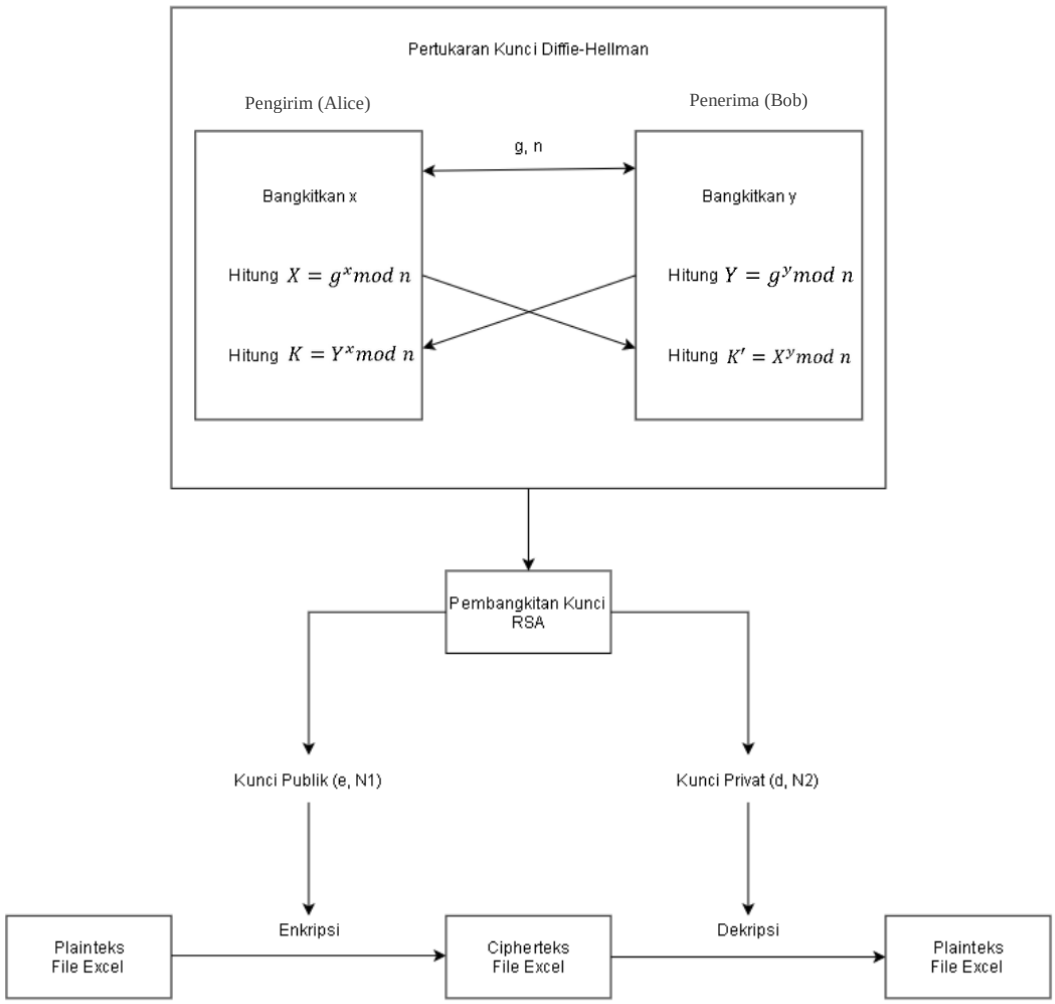


Gambar 3. 2 Skema Algoritma RSA

Skema pada Gambar 3.2 tersebut memperlihatkan bahwa Alice dan Bob menyepakati bilangan p dan q. Alice dan Bob masing-masing menghitung nilai n dan $\phi(n)$ berdasarkan bilangan p dan q. Alice kemudian menghitung nilai kunci publik e dan mengenkripsi plainteks menjadi cipherteks, dilanjutkan dengan Alice mengirim kunci publik e dan cipherteks ke Bob. Bob menghitung kunci privat d dengan kunci publik e yang diterimanya dari Alice, serta melakukan dekripsi cipherteks dengan kunci privat yang didapatnya. Jika perhitungan benar maka plainteks hasil perhitungan Bob akan sesuai dengan pesan asli yang ingin disampaikan oleh Alice.

3. 3 Pengembangan Model

Penggabungan dua algoritma yaitu *Diffie-Hellman* dan RSA bertujuan agar data di dalam *file excel* tidak mudah dipecahkan. Kombinasi Algoritma *Diffie-Hellman-RSA* telah dijelaskan pada Bab 3.4.3 dengan skema pengembangan model yang ditunjukkan pada Gambar 3.3 sebagai berikut:



Gambar 3. 3 Skema Pengamanan Data Pada File Excel

Pada mulanya, dilakukan pembangkitan kunci mengikuti algoritma *Diffie-Hellman* seperti yang sudah ditunjukkan pada skema Gambar 3.1, kemudian dilanjutkan dengan pembuatan kunci publik dan kunci privat berdasarkan modifikasi algoritma RSA. Dihasilkan suatu kunci publik $(e, N1)$ untuk mengenkripsi plainteks *file excel* menjadi cipherteks *file excel* dan dihasilkan juga kunci privat $(d, N2)$ untuk mendekripsi cipherteks *file excel* menjadi plainteks *file excel*.

3. 4 Perancangan Program

Program aplikasi menggunakan bahasa pemrograman *Python* dengan tampilan yang *user-friendly*. Adapun *library Python* yang digunakan di antaranya

pandas untuk membaca *file excel*, dan *pyxslb* untuk menulis kembali hasil enkripsi dan dekripsi file. Program memiliki empat bagian, bagian pertama merupakan halaman utama yang akan muncul saat program dijalankan. Ada juga halaman pembangkitan kunci yang berisi dua tab, yaitu tab pembangkitan kunci publik dan tab pembangkitan kunci privat. Bagian selanjutnya merupakan bagian enkripsi yang akan digunakan oleh Alice untuk mengenkripsi *file excel*, dan bagian dekripsi untuk Bob mendekripsi *file excel*.

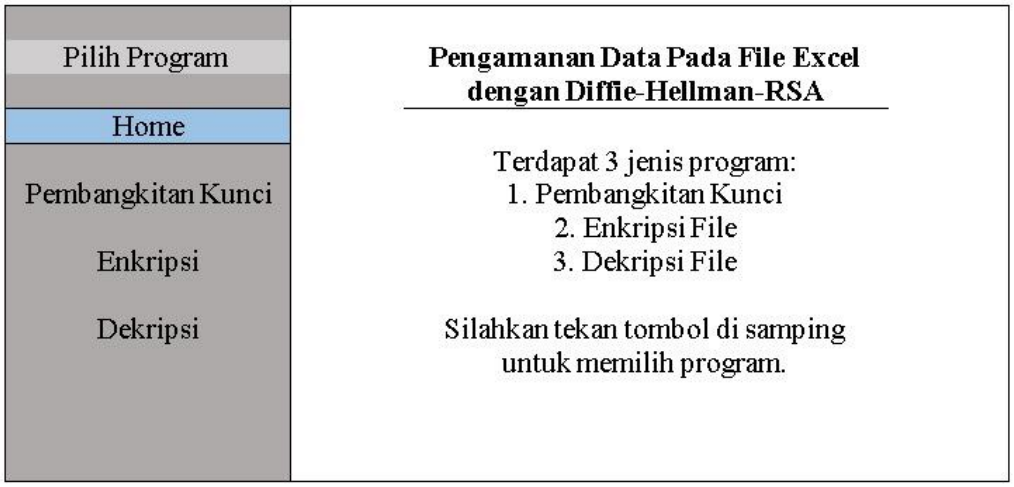
3. 4. 1 Input dan *Output*

Program melindungi data pada *file excel*, sehingga dua fungsi utamanya adalah mengenkripsi *file excel* dan juga mendekripsinya. Namun, sebelum melakukan enkripsi ataupun dekripsi, pengguna harus melakukan pembangkitan kunci di halaman pembangkitan kunci. Pada tab pembangkitan kunci publik, pengirim pesan atau Alice melakukan input empat parameter perhitungan *Diffie-Hellman-RSA*, yaitu dua bilangan prima *g* dan *n*, bilangan rahasia *x* dan nilai Y yang didapat dari Bob. Terdapat tombol menghitung nilai X dan juga tombol pembangkitan kunci publik, di mana nilai X dan kunci publik nantinya akan dikirim ke Bob untuk membangkitkan kunci privat. Kunci publik juga harus disimpan oleh Alice untuk digunakan pada proses enkripsi *file*. Pada tab pembangkitan kunci privat, penerima pesan atau Bob juga melakukan input untuk dua bilangan prima *g* dan *n*, bilangan rahasia *y*, nilai X yang didapat dari Alice dan ditambahkan dengan kunci publik yang juga didapat dari Alice. Terdapat tombol untuk menghitung nilai Y dan juga tombol pembangkitan kunci privat, kunci privat harus dijaga kerahasiaannya karena akan digunakan untuk proses dekripsi *file*.

Proses pengamanan data dilanjutkan dengan mengenkripsi data, pengguna dapat mengunggah *file excel* yang akan dienkripsi dan melakukan input nilai kunci publik. *Output* yang dihasilkan pada tab ini adalah *file excel* yang telah terenkripsi dan dapat diunduh agar kemudian dapat dikirim pada orang yang dituju dengan aman. Selanjutnya pada bagian dekripsi data, pengguna mengunggah *file excel* yang telah terenkripsi, kemudian memasukkan nilai kunci privat. Adapun *output* yang didapat adalah *file excel* dengan data yang sebenarnya.

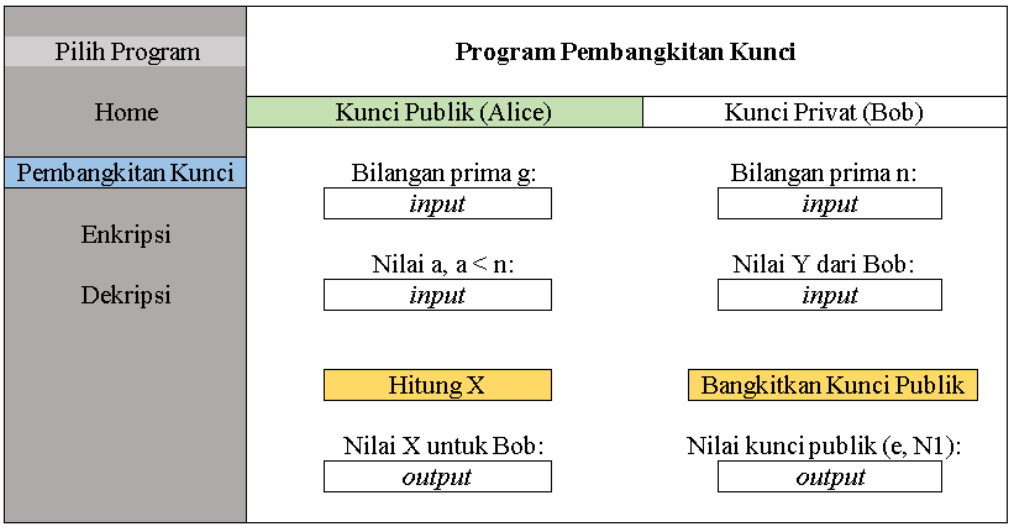
3. 4. 2 Rancangan Tampilan Program

a. Halaman Utama



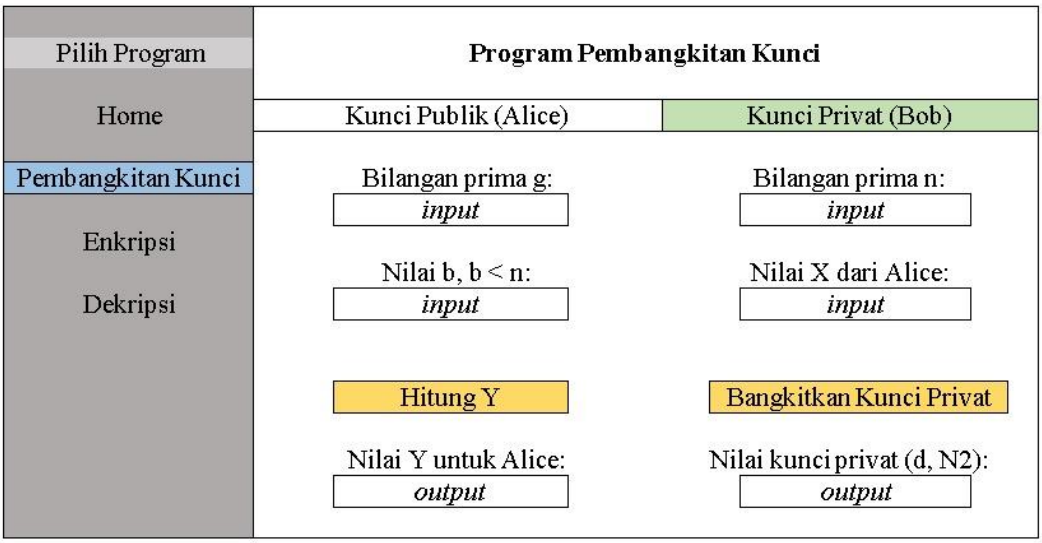
Gambar 3. 4 Rancangan Tampilan Halaman Utama

b. Pembangkitan Kunci Alice



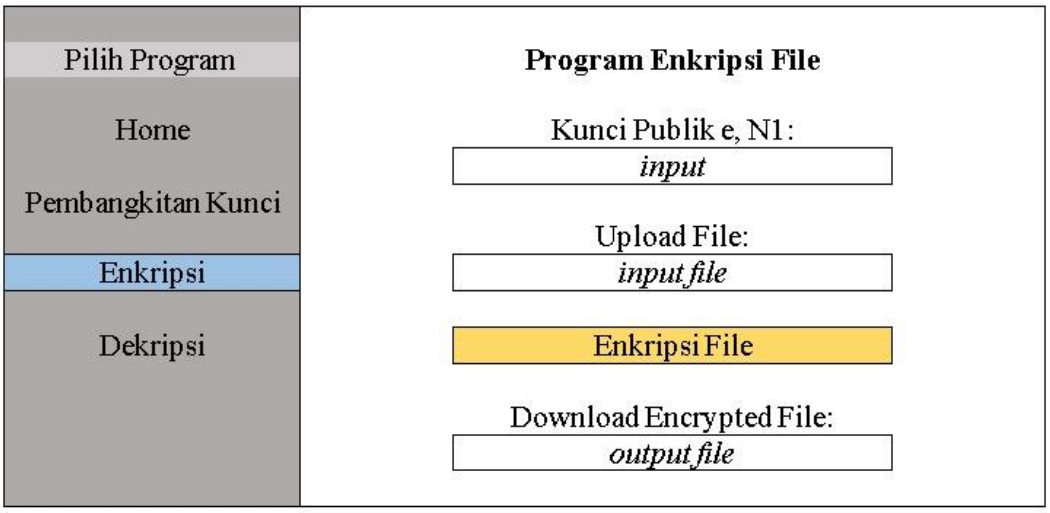
Gambar 3. 5 Rancangan Tampilan Tab Pembangkitan Kunci Publik

c. Pembangkitan Kunci Bob



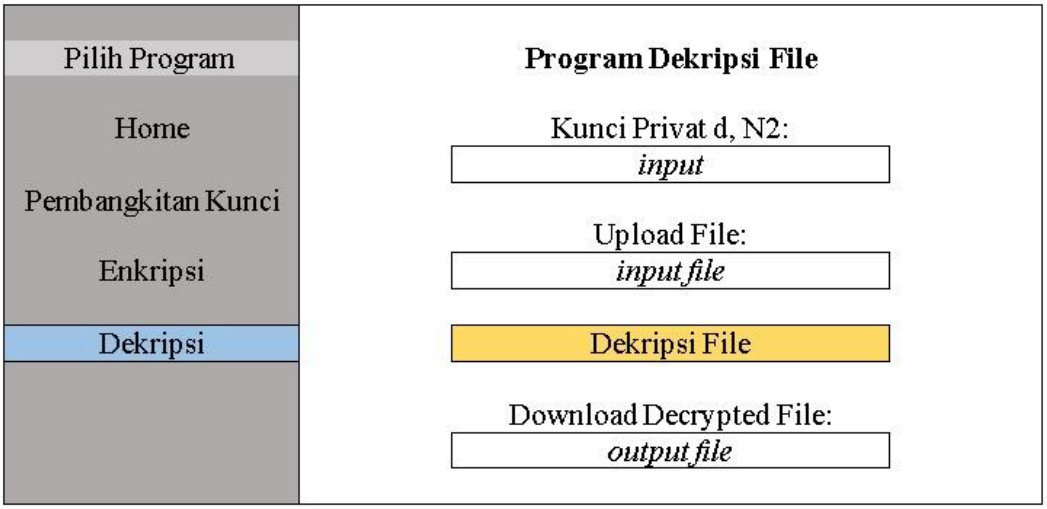
Gambar 3. 6 Rancangan Tampilan Tab Pembangkitan Kunci Privat

d. Enkripsi



Gambar 3. 7 Rancangan Tampilan Halaman Enkripsi

e. Dekripsi



Gambar 3. 8 Rancangan Tampilan Halaman Dekripsi

3. 4. 3 Algoritma Pengamanan Data

Algoritma pengamanan data menjadi acuan dalam pengembangan model yang telah dipaparkan sebelumnya. Program dimulai dari pembangkitan kunci dengan algoritma *hybrid Diffie-Hellman* dan RSA, kemudian akan dilakukan enkrispi data pada *file excel* dan dekripsi data pada *file excel*. Adapun elemen-elemen pada algoritma *Diffie-Hellman-RSA* ditunjukkan pada Tabel 3.1 berikut:

Tabel 3. 1 Elemen-elemen dalam *Diffie-Hellman-RSA*

Elemen	Keterangan	Kerahasiaan
g	bilangan prima yang dipilih, $126 < g$	**rahasia**
n	bilangan prima yang dipilih, $126 < g < n$	**rahasia**
x	kunci privat Alice, $x < n$	**rahasia**
y	kunci privat Bob, $y < n$	**rahasia**
X	kunci publik Alice	tidak rahasia
Y	kunci publik Bob	tidak rahasia
$K, K1\ dan\ K2$	kunci simetri, $K1 = K2 = K$	**rahasia**
$N1$	kunci publik, $N1 = g \cdot n \cdot K$	tidak rahasia
$N2$	kunci privat, $N2 = \frac{N1}{K}$	**rahasia**

Elemen	Keterangan	Kerahasiaan
$\phi(n)$	$\phi(n) = (p-1)(q-1)$,	**rahasia**
e	kunci enkripsi, harus memenuhi $\text{FPB}(e, \phi(n)) = 1$	tidak rahasia
d	kunci dekripsi, di mana $d \equiv e^{-1} \, mod \, \phi(n)$	**rahasia**
m	plainteks	**rahasia**
c	cipherteks	tidak rahasia

a. Pembangkitan Kunci

Pembangkitan kunci dengan algoritma *hybrid Diffie-Hellman* dan RSA sebelumnya pernah diteliti oleh Bhattacharjee, dkk (2016) dan ditemukan algoritma yang tepat untuk menyatukan *Diffie-Hellman* dan RSA. Mulanya akan dilakukan proses pembangkitan kunci dengan algoritma *Diffie-Hellman*, kemudian kunci yang didapatkan dari langkah tersebut akan disimpan sebagai salah satu elemen guna membangkitkan kunci publik dan privat dengan algoritma RSA. Adapun proses algoritma tersebut adalah sebagai berikut:

1. Alice dan Bob menyepakati dua bilangan prima (g, n) yang **bersifat rahasia** dengan $g < n$. Selain itu, nilai g dan n haruslah lebih dari 126 atau $126 < g < n$.

   Nilai g dan n harus lebih dari 126 dimaksudkan agar perhitungan enkripsi dengan kode ASCII nantinya memiliki hasil yang tepat, yaitu dengan syarat nilai g dan n lebih dari nilai kode ASCII yang digunakan. Karakter yang umum digunakan dalam penulisan pesan berada pada urutan 32 sampai dengan 126 pada tabel ASCII maka diambil nilai 126 sebagai batas bawah nilai g dan n.
2. Alice memilih suatu nilai x dengan $x < n$ dan Bob memilih suatu nilai y dengan $y < n$. Nilai x dan y bersifat rahasia.
3. Alice menghitung nilai X dan Bob menghitung nilai Y. Nilai X kemudian dikirim ke Bob oleh Alice dan sebaliknya, Bob mengirim nilai Y pada Alice.

$$X = g^x mod \, n$$

$$Y = g^y mod\ n$$

4. Alice menghitung nilai $K1$ dan Bob menghitung nilai $K2$, sedemikian sehingga didapat kunci $K = K1 = K2$.

$$K1 = Y^x mod\ n$$

$$K2 = X^y mod\ n$$

$$K = K1 = K2$$

5. Periksa apakah nilai K merupakan bilangan prima. Jika iya, maka kunci K tetap dan jika tidak maka dipilih nilai K baru, yaitu bilangan selanjutnya yang merupakan bilangan prima.
6. Perhitungan dilanjutkan dengan algoritma RSA, hitung nilai $N1$ dan $\phi(n)$

$$N1 = n \cdot g \cdot K$$

$$\phi(n) = (n-1)(g-1)$$

7. Sistem akan membangkitkan suatu bilangan acak e yang memenuhi $\text{FPB}(e, \phi(n)) = 1$. Maka didapatkan kunci publik $(e, N1)$ untuk mengenkripsi pesan dan membangkitkan kunci privat.
8. Alice mengirimkan kunci publik $(e, N1)$. Bob membangkitkan kunci privat d dan $N2$, dengan

$$d \equiv e^{-1} mod\ \phi(n)$$

$$N2 = \frac{N1}{K}$$

9. Diperoleh kunci privat $(d, N2)$ untuk mendekripsi pesan.

b. Proses Enkripsi dan Dekripsi *File Excel*

Proses enkripsi dan dekripsi *file excel* mengikuti alur algoritma RSA. Pesan plainteks akan dibagi menjadi blok-blok plainteks kemudian diubah menjadi blok-blok cipherteks yang nantinya menjadi satu bagian cipherteks. Lebih lanjut akan dipaparkan proses enkripsi *file excel*:

1. Alice mengunggah *file excel* yang akan dienkripsi serta memasukkan kunci publik $(e, N1)$
2. Sistem akan membaca data yang terdapat dalam *file excel* mulai dari baris dan kolom pertama yang berisi data hingga baris dan kolom terakhir

3. Pesan atau data pada tiap *cell* akan dipartisi menjadi blok-blok plainteks m, yang kemudian dikonversi menjadi kode ASCII. Setelahnya, blok-blok plainteks diubah menjadi blok-blok cipherteks c dengan rumus berikut:

$$c_i = m_i^e \ mod \ N1$$

4. Cipherteks kemudian menggantikan nilai plainteks pada *cell* yang sama sehingga diperoleh *file excel* dengan data yang telah dienkripsi
5. Alice dapat mengunduh *file excel* yang telah terenkripsi dan mengirimnya pada Bob.

Sedangkan untuk proses dekripsi *file excel* dijelaskan sebagai berikut:

1. Bob mengunggah *file excel* yang didapatnya dari Alice serta memasukkan kunci privat $(d, N2)$
2. Sistem akan membaca data yang terdapat dalam *file excel* mulai dari baris dan kolom pertama yang berisi data hingga baris dan kolom terakhir
3. Pesan atau data pada tiap *cell* akan dipartisi menjadi blok-blok cipherteks, kemudian diubah menjadi blok-blok plainteks dengan rumus berikut:

$$m_i = c_i^d \ mod \ N2$$

4. Plainteks yang masih berupa kode ASCII kemudian dikonversi kembali menjadi karakter yang sesuai dan plainteks tersebut menggantikan nilai cipherteks pada *cell* yang sama sehingga diperoleh *file excel* dengan data yang telah didekripsi
5. Bob dapat mengunduh *file excel* yang telah didekripsi, sehingga didapatkan data pada *file excel* yang sama dengan data asli yang dimiliki Alice.

3. 5 Rancangan Validasi

Program akan divalidasi untuk memastikan program berjalan dengan benar. Validasi dilakukan dengan cara mencocokkan hasil enkripsi dan dekripsi dari perhitungan manual *Diffie-Hellman-RSA* dengan data yang dienkripsi dan didekripsi oleh program. Jika perhitungan manual dan perhitungan yang dilakukan oleh program menghasilkan nilai yang sama, maka program tervalidasi dan dapat digunakan untuk melindungi data pada *file excel*.