BAB III

METODOLOGI PENELITIAN

3.1 Identifikasi Masalah

Dalam proses pertukaran pesan dengan mesin Enigma, *cipher text* dikirim bersamaan dengan *key* yang berupa pengaturan awal mesin Enigma yang digunakan antara lain jenis rotor, pengaturan rotor, pengaturan huruf awal dari rotor yang digunakan, dan pengaturan *plugboard*. Untuk mengamankan *key* Enigma ini dari pihak ketiga, *key* dapat dienkripsi dengan algoritma kriptografi modern. Dalam penelitian ini kriptografi modern yang akan digunakan adalah algoritma ElGamal. Algoritma ElGamal dipilih karena sulitnya menyelesaikan masalah algoritma diskrit akibat bilangan acak k yang dipilih pada saat proses enkripsi dirahasiakan. Oleh karena itu, luaran dari proses enkripsi keseluruhan berupa *cipher text* Enigma yang berisi pesan rahasia dan *cipher text* ElGamal yang berisi *key* untuk *cipher text* Enigma.

3.2 Model Dasar

3.2.1 Algoritma Enigma

Dalam algoritma Enigma, sebelum melakukan proses enkripsi, hal yang perlu dilakukan adalah menentukan jenis rotor yang akan digunakan dan pengaturan rotornya. Misalkan jenis rotor yang akan digunakan adalah Rotor I, Rotor II, dan Rotor III dengan pengaturan berurutan, serta reflektor yang digunakan adalah reflektor B. Setelah itu dilakukan inisialisasi huruf-huruf pada rotor ke bentuk angka sebagai berikut:

Tabel 3.1

Inisialisasi Rotor dan Reflektor

Reflektor B			Rotor I			Rotor II			Rotor III			Asal	
Y	0	24	E	0	4	A	0	0	B	0	1	A	0
R	1	17	K	1	10	J	1	9	D	1	3	B	1
U	2	20	M	2	12	D	2	3	F	2	5	C	2
H	3	7	F	3	5	K	3	10	H	3	7	D	3
Q	4	16	L	4	11	S	4	18	J	4	9	E	4
S	5	18	G	5	6	I	5	8	L	5	11	F	5
L	6	11	D	6	3	R	6	17	C	6	2	G	6
D	7	3	Q	7	16	U	7	20	P	7	15	H	7
P	8	15	V	8	21	X	8	23	R	8	17	I	8
X	9	23	Z	9	25	B	9	1	T	9	19	J	9
N	10	13	N	10	13	L	10	11	X	10	23	K	10
G	11	6	T	11	19	H	11	7	V	11	21	L	11
O	12	14	O	12	14	W	12	22	Z	12	25	M	12
K	13	10	W	13	22	T	13	19	N	13	13	N	13
M	14	12	Y	14	24	M	14	12	Y	14	24	O	14
I	15	8	H	15	7	C	15	2	E	15	4	P	15
E	16	4	X	16	23	Q	16	16	I	16	8	Q	16
B	17	1	U	17	20	G	17	6	W	17	22	R	17
F	18	5	S	18	18	Z	18	25	G	18	6	S	18
Z	19	25	P	19	15	N	19	13	A	19	0	T	19
C	20	2	A	20	0	P	20	15	K	20	10	U	20
W	21	22	I	21	8	Y	21	24	M	21	12	V	21
V	22	21	B	22	1	F	22	5	U	22	20	W	22
J	23	9	R	23	17	V	23	21	S	23	18	X	23
A	24	0	C	24	2	O	24	14	Q	24	16	Y	24
T	25	19	J	25	9	E	25	4	O	25	14	Z	25

Prasetio, Triandi, & Hardianto (2018) menuliskan operasi pada proses enkripsi adalah:

$$c_i = (m_{1,i} + k_i) mod\ 26 \rightarrow (m_{2,i} - k_i) mod\ 26$$

dengan:

c_i : *cipher text* ke-i

$m_{1,i}$: karakter *plain text* ke-i

$m_{2,i}$: pasangan karakter *plain text* ke-i pada rotor selanjutnya

k_i : kunci rotor (3→2→1→reflektor(0) −→1−→2−→3)

→ : transformasi tabel rotor

Proses dekripsi dilakukan dengan langkah-langkah yang sama pada proses enkripsi. Setelah mengatur rotor dan melakukan inisialisasi, *cipher text* didekripsi dengan operasi:

$$m_i = (c_{1,i} + k_i) mod\ 26\ \rightarrow (c_{2,i} - k_i) mod\ 26$$

dengan:

m_i : *plain text* ke-i

$c_{1,i}$: karakter *cipher text* ke-i

$c_{2,i}$: pasangan karakter *cipher text* ke-i pada rotor selanjutnya

k_i : kunci rotor (3→2→1→reflektor(0) −→1−→2−→3)

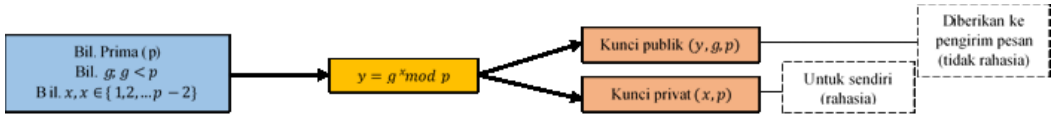
→ : transformasi tabel rotor

Arah pergerakan rotor adalah Rotor 3 – Rotor 2 – Rotor 1 – Reflektor – Rotor 1 – Rotor 2 – Rotor 3, berlaku untuk kedua proses enkripsi dan dekripsi. Skema proses enkripsi dan dekripsi dengan algoritma Enigma dapat dilihat pada Gambar 3.1.

Gambar 3.1 Skema proses enkripsi dan dekripsi algoritma Enigma

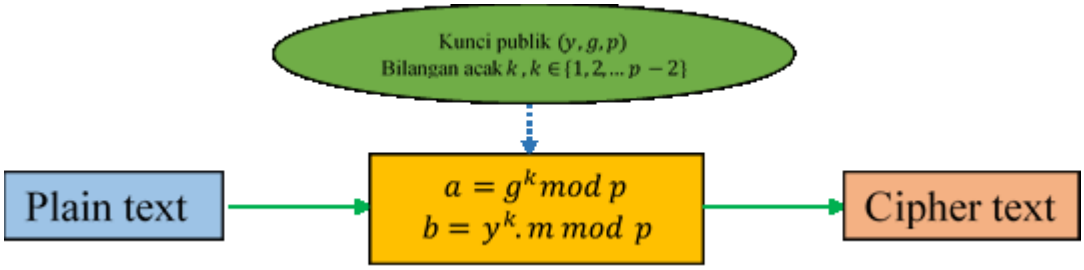
3.2.2 Algoritma ElGamal

Pada perancangan aplikasi ini, algoritma ElGamal akan diterapkan untuk mengenkripsi *key* Enigma menjadi sebuah *cipher key*. Sebelum melakukan proses enkripsi dan dekripsi *cipher key*, terlebih dahulu kedua pihak yang akan saling bertukar pesan (Alice dan Bob) melakukan pembangkitan kunci ElGamal (kunci publik dan kunci privat). Kemudian Alice dan Bob saling bertukar kunci publik yang tidak bersifat rahasia. Skema proses pembangkitan kunci ElGamal dapat disajikan pada Gambar 3.2.



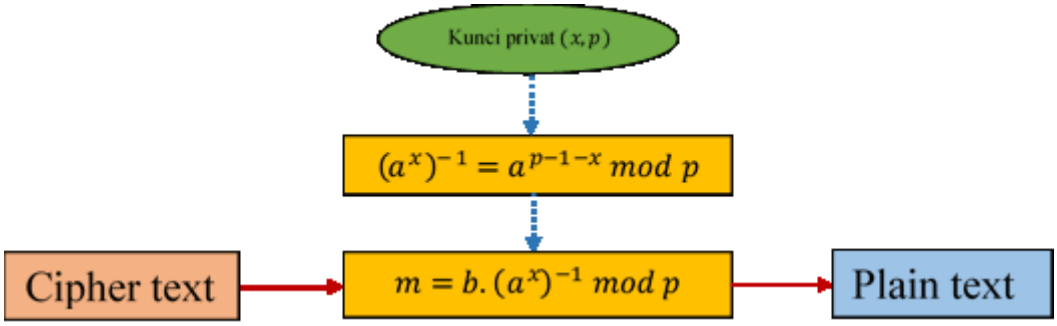
Gambar 3.2 Skema proses pembangkitan kunci dengan algoritma ElGamal

Misalkan Bob melakukan pembangkitan kunci, kemudian kunci publik (y, g, p) yang telah diperoleh diberikan kepada Alice untuk melakukan enkripsi pesan (*key* Enigma) dengan algoritma ElGamal. Saat melakukan enkripsi, selain menggunakan kunci publik yang diberikan Bob, Alice juga menggunakan suatu bilangan acak k yang bersifat rahasia dalam perhitungannya. Skema proses enkripsi yang dilakukan Alice disajikan pada Gambar 3.3.



Gambar 3.3 Skema proses enkripsi dengan algoritma ElGamal

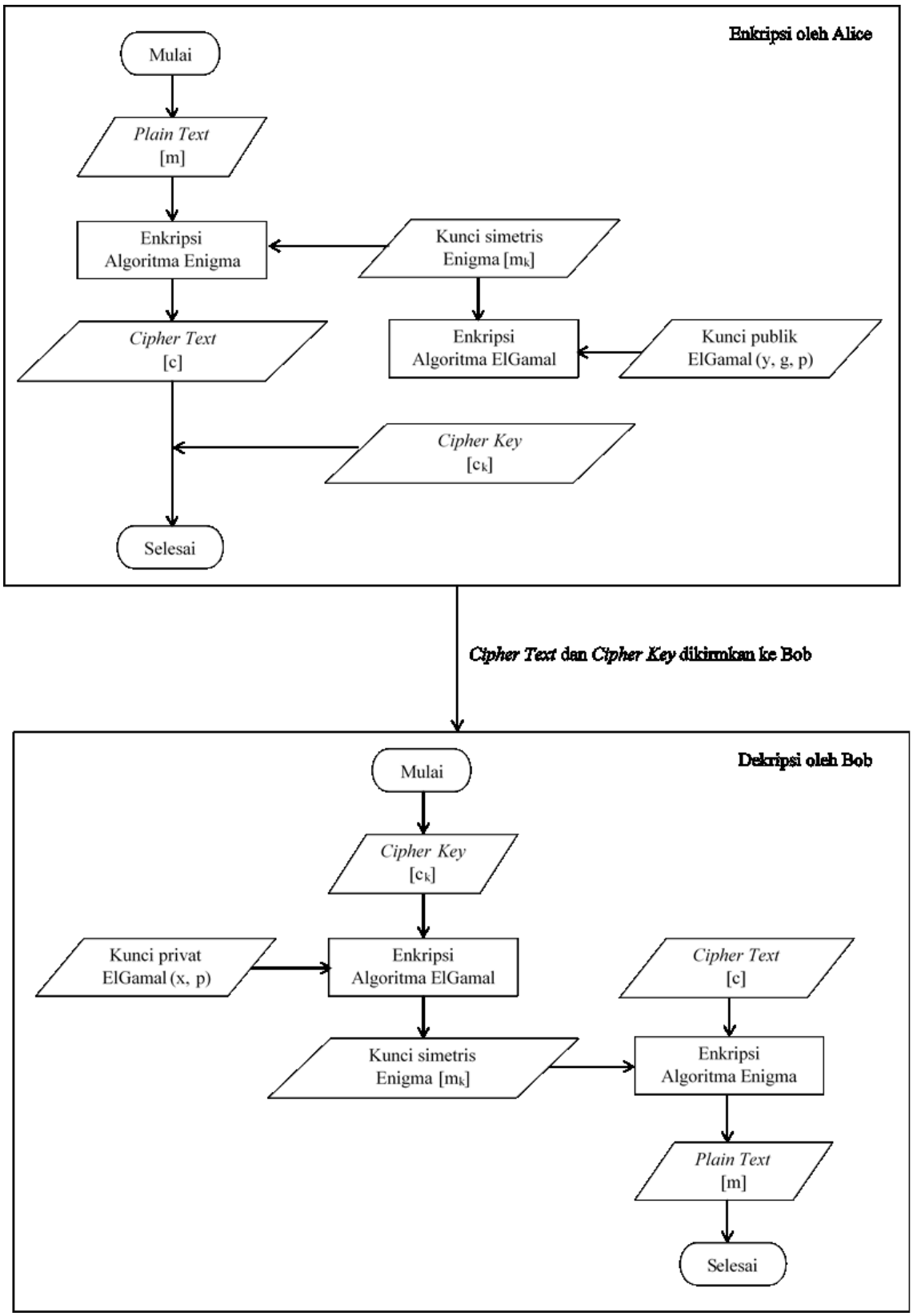
Setelah *cipher key* diperoleh, Alice mengirimkan *cipher key* dan *cipher text* kepada Bob. Kemudian sebelum Bob melakukan dekripsi untuk *cipher text* dengan algoritma Enigma, terlebih dahulu Bob melakukan dekripsi untuk *cipher key* dengan algoritma ElGamal. Saat melakukan dekripsi, Bob menggunakan kunci privat (x, p) miliknya yang diperoleh dari proses pembangkitan kunci. Skema proses dekripsi *cipher key* yang dilakukan Bob disajikan pada Gambar 3.4.



Gambar 3.4 Skema proses dekripsi dengan algoritma ElGamal

3.3 Pengembangan Model

Pada penelitian ini, Alice (berperan sebagai pengirim pesan rahasia) dan Bob (berperan sebagai penerima pesan rahasia) sebelumnya telah saling bertukar suatu kunci publik untuk digunakan pada algoritma ElGamal. Alice mengenkripsi plain text (m) dengan algoritma Enigma menggunakan suatu key Enigma (m_k) berupa pengaturan awal (jenis rotor yang digunakan, pengaturan rotor, dan huruf awal pada rotor). Dari proses pengenkripsian plain text (m) ini dihasilkan suatu cipher text (c). Kemudian Alice juga mengenkripsi key Enigma (m_k) dengan algoritma ElGamal menggunakan kunci publik Bob. Proses pengenkripsian ini dihasilkan cipher lainnya, misalkan disebut cipher key (c_k). Alice kemudian mengirimkan cipher text (c) dan cipher key (c_k) kepada Bob. Untuk mengetahui isi pesan, pertama-tama Bob menggunakan kunci privat miliknya yang bersifat rahasia untuk mendekripsi cipher key (c_k) dengan algoritma ElGamal. Hasil pendekripsian cipher key (c_k) akan dihasilkan suatu key, dalam hal ini key Enigma (m_k). Setelah key Enigma (m_k) berhasil didapatkan, Bob mendekripsi cipher text (c) dari Alice dengan algoritma Enigma menggunakan key Enigma (m_k) yang telah didekripsi sebelumnya, sehingga diperoleh plain text (m) atau pesan asli yang dikirim oleh Alice. Skema keseluruhhan kriptografi *hybrid* dengan algoritma Enigma dan algoritma ElGamal dapat dilihat pada Gambar 3.5.



Gambar 3.5 Skema Kriptografi *Hybrid* dengan algoritma Enigma dan ElGamal

3.4 Konstruksi Program

3.4.1 Input dan Output

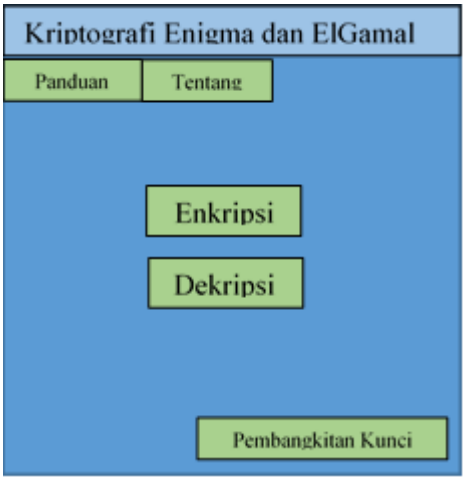
Pembuatan program ini terbagi menjadi tiga bagian, yaitu pembangkitan kunci, proses enkripsi (dengan algoritma Enigma dan ElGamal), serta proses dekripsi (dengan algoritma Enigma dan ElGamal). Proses pembangkitan kunci, inputnya berupa bilangan integer yang memenuhi properti algoritma ElGamal. Setelah diproses, outputnya berupa kunci publik dan kunci privat.

Pada proses enkripsi dengan algoritma Enigma, inputnya berupa string dari isi pesan asli atau *plain text* dan outputnya juga berupa string dari pesan rahasia atau *cipher text*. Pada proses enkripsi dengan algoritma ElGamal, inputnya berupa bilangan integer dari kunci publik dan string dari kunci Enigma atau *plainkey*, kemudian outputnya adalah *cipher key* berupa pasangan bilangan integer dari setiap karakter *plainkey* yang diinputkan.

Proses dekripsi dengan algoritma Enigma inputnya berupa string dari pesan rahasia atau *cipher text* dan outputnya berupa string dari pesan asli atau *plain text*. Sedangkan pada proses dekripsi dengan algoritma ElGamal, inputnya berupa pasangan bilangan *cipher key* dan kunci privat, kemudian outputnya adalah *plain key* atau kunci Enigma berupa string.

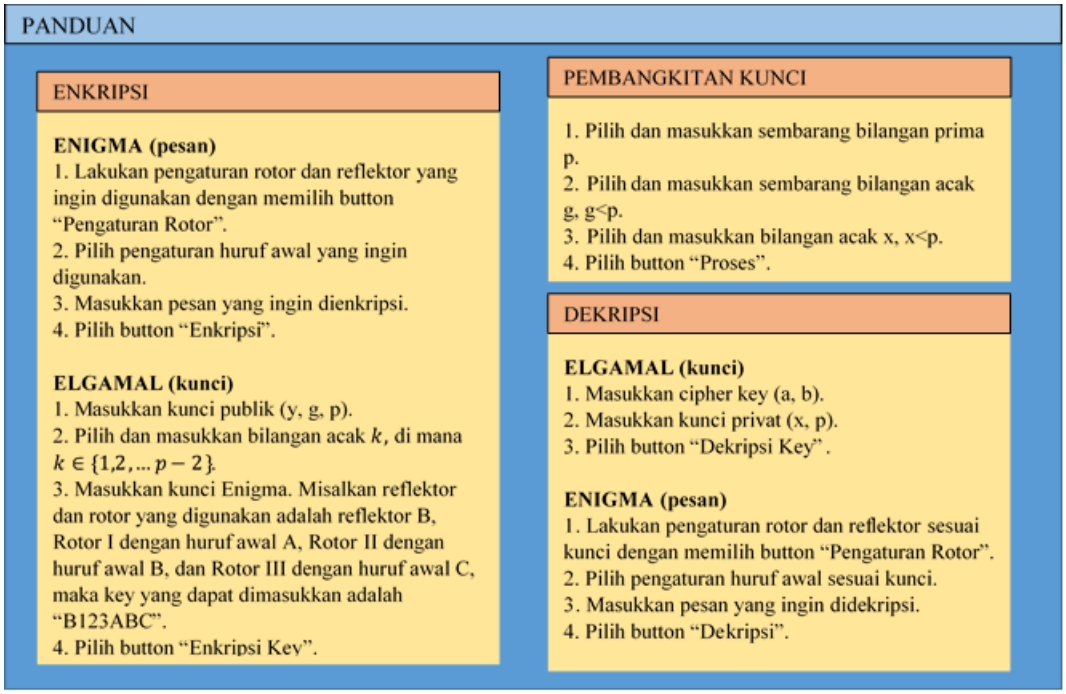
3.4.2 Perancangan Tampilan

Saat program dijalankan, *window* utama yang muncul adalah berupa *main menu* yang berisi *button* “Panduan”, “Pembangkitan Kunci”, “Enkripsi”, “Dekripsi”, dan “Tentang” yang jika dipilih akan memunculkan *window* baru dengan fungsi yang berbeda-beda sesuai kebutuhan pengguna.



Gambar 3.6 Rancangan Tampilan *Window* Main Menu

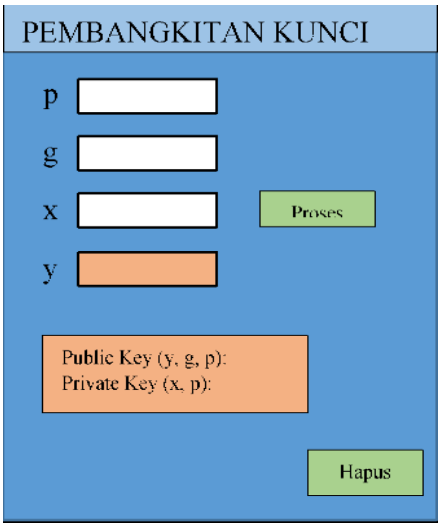
Pada *window* Panduan, akan ditampilkan panduan menggunakan program mulai dari pembangkitan kunci, enkrips Enigma, enkripsi ElGamal, dekripsi Enigma, dan dekripsi ElGamal.



Gambar 3.7 Rancangan Tampilan *Window* Panduan

Window Pembangkitan Kunci berfungsi untuk membangkitkan kunci publik dan kunci privat yang digunakan pada kriptografi ElGamal. Pada *window* ini pengguna diminta untuk menginput properti-properti bilangan yang dibutuhkan

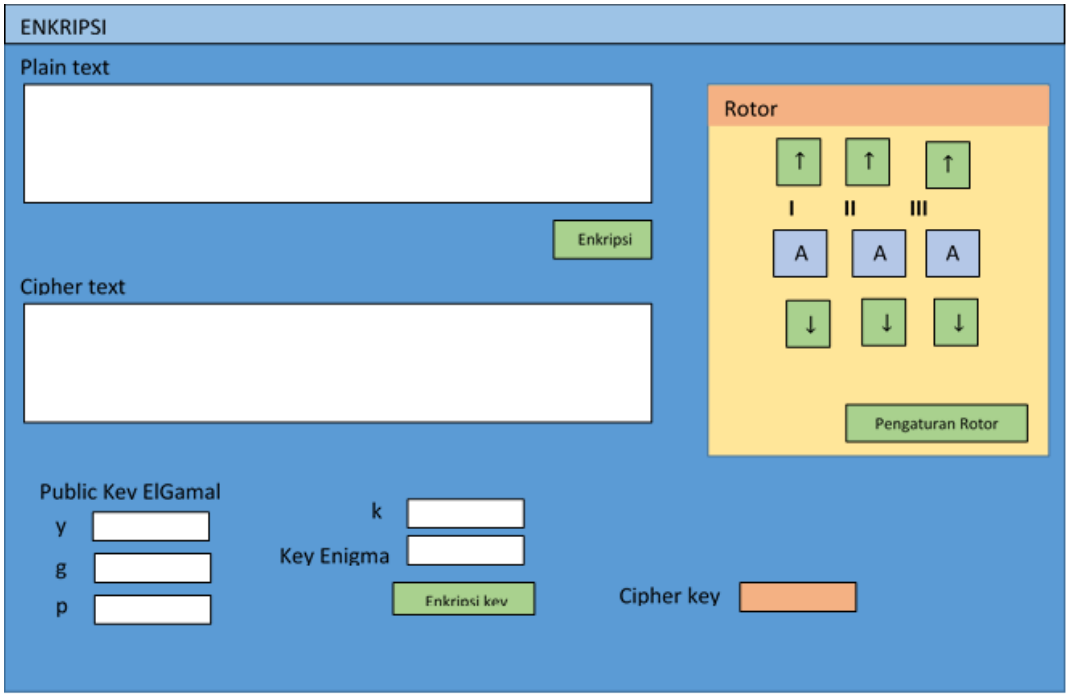
yaitu p, g, dan x. *Button* “Proses” pada *window* ini akan menghitung nilai y dan ditampilkan pada *window*. Kemudian akan ditampilkan juga hasil akhir pada *window* Pembangkitan Kunci yaitu kunci publik dan kunci privat.



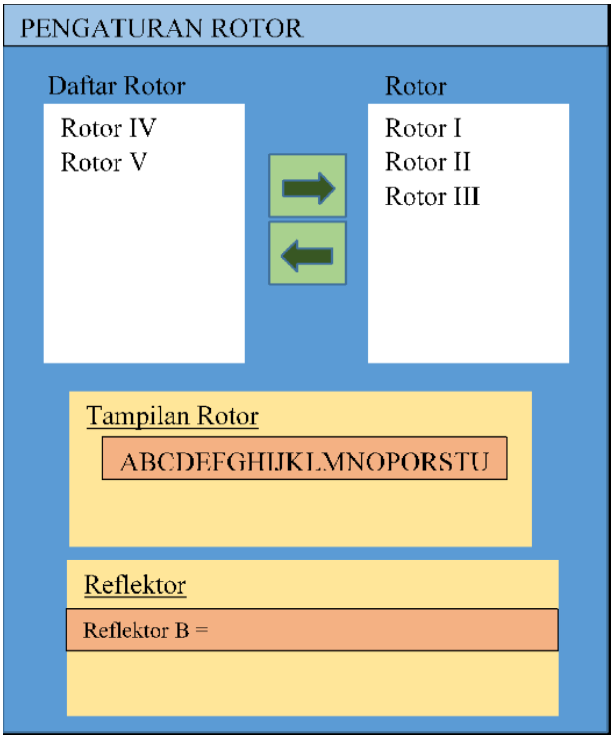
Gambar 3.8 Rancangan Tampilan *Window* Pembangkitan Kunci

Window Enkripsi berfungsi untuk mengenkripsi pesan atau *plaintext* dengan algoritma Enigma. Setelah menginputkan isi pesan, dengan *button* “Enkripsi” program akan memproses pesan dan menampilkan *ciphertext.* Pengguna juga dapat memilih jenis rotor yang ingin digunakan dengan memilih *button* “Pengaturan Rotor” yang akan memunculkan *window* baru.

Selain itu, pada *window* ini pengguna dapat melakukan enkripsi *plainkey* dengan algoritma ElGamal. Dengan menginputkan kunci publik yang telah dibangkitkan sebelumnya dan pengaturan rotor yang digunakan pada proses enkripsi pesan, program akan menampilkan *cipher key* setelah pengguna memilih *button* “Enkripsi Key”.



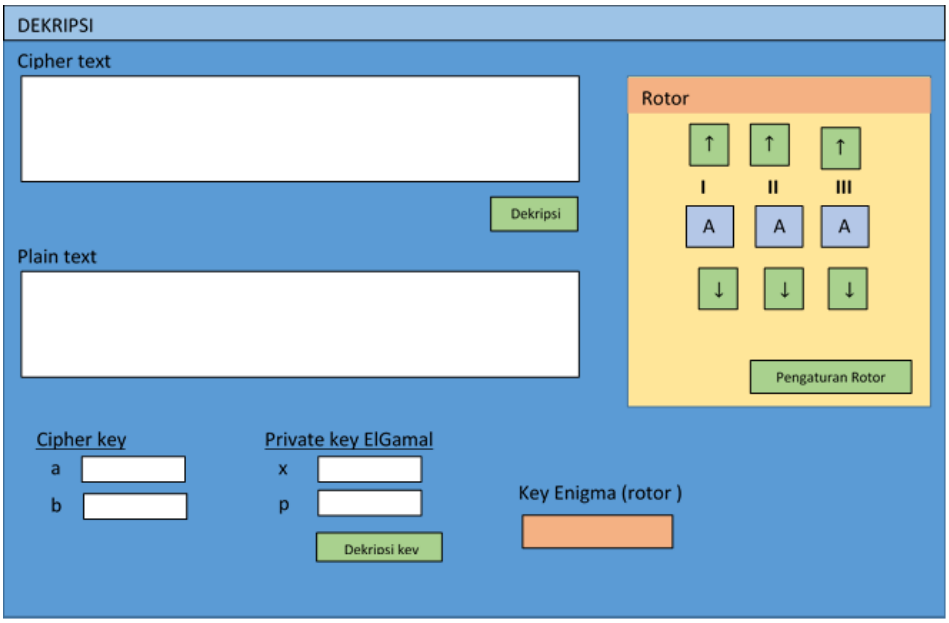
Gambar 3.9 Rancangan Tampilan *Window* Enkripsi



Gambar 3.10 Rancangan *Window* Pengaturan Rotor.

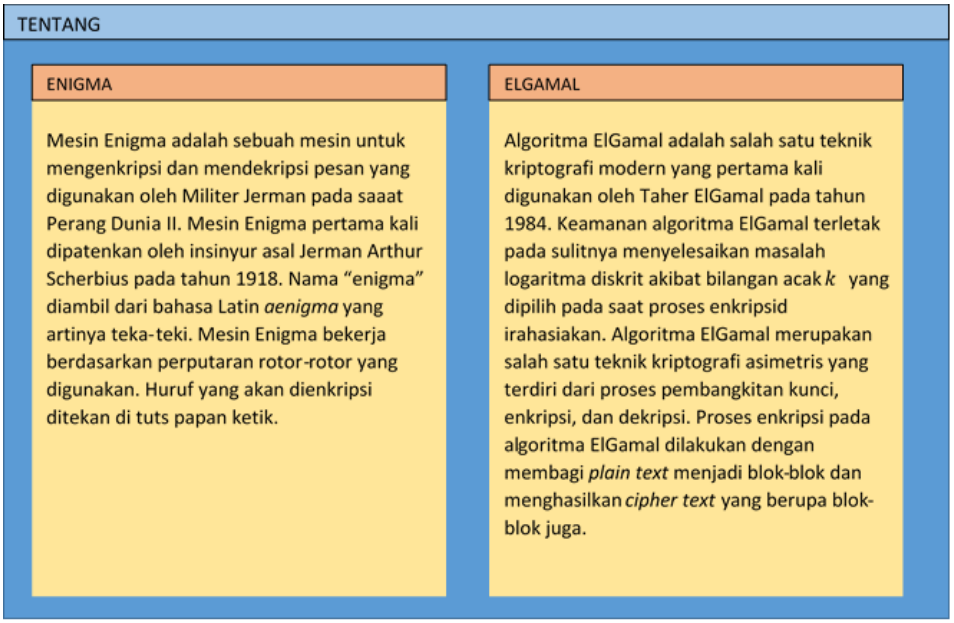
Window Dekripsi berfungsi untuk mengenkripsi pesan rahasia atau *ciphertext* dengan algoritma Enigma. Sebelum mendekripsi pesan rahasia,

pengguna dapat melakukan dekripsi *cipherkey* dengan algoritma ElGamal untuk memperoleh kunci *plainkey* yang berupa pengaturan rotor. Pengguna dapat mengatur jenis rotor yang digunakan sesuai kunci dengan memilih *button* "Pengaturan Rotor" yang akan memunculkan *window* baru. Setelah menginputkan pesan rahasia, dengan *button* "Dekripsi" program akan memproses pesan dan menampilkan *plaintext.*



Gambar 3.11 Rancangan Tampilan *Window* Dekripsi

Window Tentang berisi informasi umum mengenai Enigma dan ElGamal.



Gambar 3.12 Rancangan Tampilan *Window* Tentang

3.4.3 Algoritma

3.4.3.1 Algoritma Pembangkitan Kunci ElGamal

Pada subbab ini akan dituliskan algoritma yang akan digunakan dalam proses pembangkitan kunci publik dan kunci privat dengan algoritma ElGamal sebelum kedua pihak saling bertukar pesan. Algoritma yang akan digunakan adalah:

1) Input sembarang p, dengan p adalah bilangan prima.
2) Input g dengan $g < p$.
3) Input x dengan $x \in \{1,2, \dots p-2\}$.
4) Menghitung y dengan $y = g^x mod\ p$.
5) Menampilkan bilangan y.
6) Menampilkan kunci publik (y, g, p).
7) Menampilkan kunci privat (x, p).

3.4.3.2 Algoritma Enkripsi *Plain Text* dengan Algoritma Enigma

Pada subbab ini akan dituliskan algoritma yang akan digunakan dalam proses pengenkripsian pesan asli atau *plain text* dengan algoritma Enigma. Algoritma yang akan digunakan adalah:

1) Pilih 3 buah rotor yang akan digunakan.
2) Atur huruf awal yang akan digunakan pada setiap rotor.
3) Input *plain text* m.

4) Mengelompokkan karakter-karakter *plain text* m_i menjadi *array of string.*
5) Mengubah karakter dalam *plain text* m menjadi angka dalam modulo 26.
6) Dengan pengulangan *for*, untuk setiap indeks di array akan dihitung *cipher text* c_i dengan rumus $c_i = (m_{1,i} + k_i) mod\ 26 \rightarrow (m_{2,i} - k_i) mod\ 26$.
7) Mengubah c_i menjadi karakter.

3.4.3.3 Algoritma Enkripsi *Key* Enigma dengan Algoritma ElGamal

Pada subbab ini akan dituliskan algoritma yang akan digunakan dalam proses pengenkripsian kunci Enigma (rotor) atau *key Enigma* dengan algoritma ElGamal. Algoritma yang akan digunakan adalah:

1) Input key Enigma (m_k) berupa string.
2) Input bilangan acak k dengan $k \in \{1,2, \dots p-2\}$.
3) Input kunci public (y, g, p).
4) Mengelompokkan string key Enigma menjadi array of character.
5) Mengubah array of character menjadi kode ASCII.
6) Dengan pengulangan for, untuk setiap indeks di array dihitung
$$b = y^k m_i \ mod\ p$$
7) Menghitung
$$a = g^k \ mod\ p$$
8) Menampilkan *cipher key* c_k a dan array of b.

3.4.3.4 Algoritma Dekripsi *Key* Enigma dengan Algoritma ElGamal

Pada subbab ini akan dituliskan algoritma yang akan digunakan dalam proses pendekripsian kunci Enigma (rotor) atau *key Enigma* dengan algoritma ElGamal. Algoritma yang akan digunakan adalah:

1) Input *cipher key* c_k a dan b berupa array of integer yang dipisahkan oleh spasi " ".
2) Input kunci privat (x, p).
3) Menghitung $(a^x)^{-1} = a^{p-1-x} \ mod\ p$.
4) Dengan pengulangan for, untuk setiap indeks di array dari b dihitung
$$m_i = \left(\frac{b}{a^x}\right) mod\ p = b.(a^x)^{-1} \ mod\ p$$
5) Mengubah setiap indeks m_i ke bentuk karakter.

3.4.3.5 Algoritma Dekripsi *Cipher Text* dengan Algoritma Enigma

Pada subbab ini akan dituliskan algoritma yang akan digunakan dalam proses pendekripsian pesan rahasia dengan algoritma Enigma. Algoritma yang akan digunakan adalah:

1) Pilih 3 buah rotor yang akan digunakan.
2) Atur huruf awal yang akan digunakan pada setiap rotor.
3) Input *cipher text* c.
4) Mengelompokkan karakter-karakter *plain text* c_i menjadi *array of string.*
5) Mengubah karakter dalam *cipher text* m menjadi angka dalam modulo 26.
6) Dengan pengulangan *for*, untuk setiap indeks di array akan dihitung *plain text* m_i dengan rumus

$$m_i = (c_{1,i} + k_i) mod\ 26 \rightarrow (c_{2,i} - k_i) mod\ 26$$

7) Mengubah m_i menjadi karakter.

3.4.4 Pembuatan Program

Program aplikasi ini akan dibuat menggunakan bahasa pemrograman Python 3.9.1. Program dibuat menggunakan *Graphical User Interface* (GUI). *Library* standar pada Python untuk membuat GUI adalah Tkinter. Tkinter cenderung mudah dan cepat digunakan untuk membuat aplikasi GUI. Selain itu, Tkinter menyediakan objek-objek *interface* yang mudah digunakan agar program yang dibuat *user-friendly*.

3.4.5 Validasi

Tahap validasi program yang dibuat akan dilakukan dengan membandingkan hasil perhitungan program dengan hasil perhitungan manual menggunakan bantuan kalkulator atau Microsoft Excel.