

BAB III

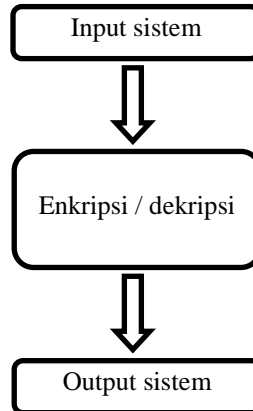
PERENCANAAN SISTEM

Pada bab ini pembahasan akan difokuskan pada pembuatan dan perancangan sistem yang merupakan pokok pembahasan dari proyek akhir.

3.1 Perancangan Sistem

Perangkat lunak untuk enkripsi dan dekripsi gambar menggunakan algoritma secure image protection dirancang dengan menggunakan bahasa pemrograman Java dengan beberapa komponen standar seperti Frame, Text Area, Button, Label,, dan sebagainya.

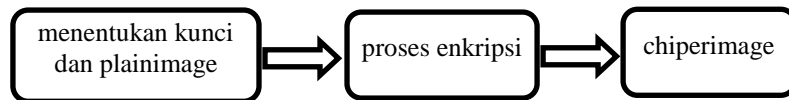
Pada subbab ini akan dibahas mengenai perancangan dan pembuatan aplikasi. Keseluruhan blok diagram Secure Image Protection dapat dilihat pada gambar 3.1



Gambar 3.1. Blok diagram sistem aplikasi

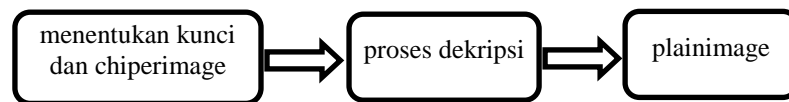
Secure Image Protection (SIP) secara umum terdiri dari input sistem yang terdiri dari gambar yang akan dienkripsi atau didekripsi dan kunci kemudian dilakukan proses enkripsi atau dekripsi kemudian menghasilkan output gambar yang sudah disembunyikan informasinya. Untuk proses dekripsi dilakukan dengan

modus kebalikan dari proses enkripsi. Secara garis besar sistem untuk mendapatkan gambar yang telah di enkripsi ditunjukkan pada gambar 3.2. Untuk mengembalikan gambar yang telah di enkripsi ke gambar aslinya atau proses dekripsi ditunjukkan pada gambar 3.3 .



Gambar 3.2. Blok diagram enkripsi gambar

Sedangkan secara garis besar proses dekripsi digambarkan pada blok diagram pada gambar 3.3..



Gambar 3.3. Blok diagram dekripsi gambar

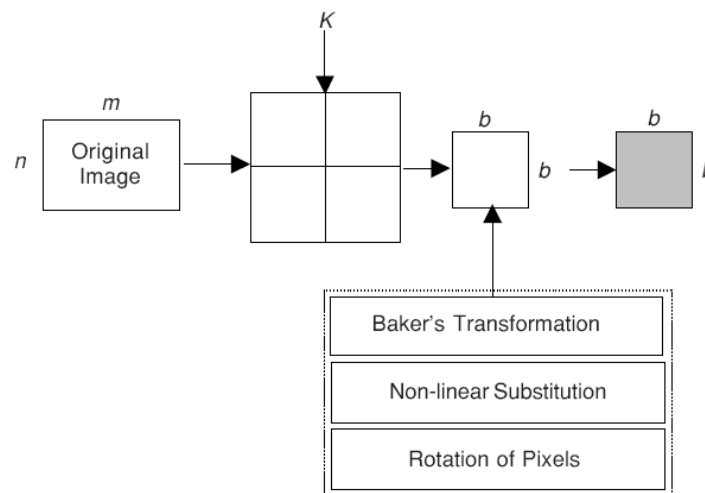
1. Input Sistem

Input untuk sistem ini adalah gambar yang akan dienkripsi dan nilai kunci. Dalam tugas akhir ini terdapat fungsi f dimana fungsi f ini digunakan untuk menunjukkan gambar ukuran $m \times n$ dimana m dan n mewakili baris dan kolom gambar masing-masing. $f(x, y)$ adalah nilai skala keabuan piksel pada posisi x dan y di mana $0 < x \leq m - 1$ dan $0 < y \leq n - 1$. Sebelum melanjutkan ke proses enkripsi, maka gambar akan menjalani konfigurasi awal.

Padding piksel yang ditambahkan pada gambar yang tidak memenuhi syarat metode SIP. Syarat metode ini adalah gambar harus dapat dipartisi menjadi blok-blok persegi ukuran $b \times b$. Kondisi padding piksel sebagai berikut

- $\text{Padding piksel} = n - (m \% n)$
- jika ukuran gambar $m < n$, gambar dibagi menjadi a blok dimana $a = (n + \text{padding piksel}) / m$ dan $b = m$,
- jika ukuran gambar $m > n$, gambar dibagi menjadi a blok dimana $a = (m + \text{padding piksel}) / n$ dan $b = n$.

Sedangkan kunci K (K_r) terdiri dari parameter jumlah iterasi dilambangkan dengan K_r . Pada kunci (K) memiliki fungsi untuk menentukan jumlah iterasi dan jumlah pergeseraan piksel. Jumlah iterasi pada dasarnya menentukan tingkat keamanan. Jelas jumlah iterasi yang lebih tinggi meningkatkan waktu komputasi meningkatkan keamanan citra cipher, karena hal ini akan meningkatkan kerja untuk brute force attack. Untuk sistem keseluruhan dari algoritma SIP ditunjukkan pada Gambar 3.4 dibawah ini.



Gambar 3.4 SIP menunjukkan sistem secara keseluruhan.

2. Proses Enkripsi

Proses enkripsi terdiri dari tiga fungsi utama.

Fungsi 1: Melakukan permutasi piksel dengan asumsi fungsi telah dimodifikasi.

1. untuk iterasi sebanyak Kr
2. untuk $k = 0$ sampai $k < \text{jumlah blok (a)}$, lakukan :
 - untuk $i = 0$ sampai $i < b[k]$, lakukan :
 1. $\text{tempNode} = \text{null}$
 2. $\text{nodeAwal} = b[k]/2 + i + k$
 3. masuk ke fungsi $\text{cek}(\text{nodeAwal})$
 4. untuk $j = 0$ sampai $j < b[k]/2$
 - $\text{nodeAwal} = \text{nodeAwal} + j$
 - masuk ke fungsi $\text{cek}(\text{nodeAwal})$
 - masuk ke fungsi $\text{cekNode}(\text{nodeAwal})$
 - $\text{nodeTujuan} = \text{nodeAwal} + 2 + j + i + k$
 - masuk ke fungsi $\text{cek}(\text{nodeTujuan})$
 - masuk ke fungsi $\text{cekNode}(\text{nodeTujuan})$
 - $\text{swap}(\text{nodeAwal}, \text{nodeTujuan})$

dimana fungsi :

- $\text{cek}(\text{node})$
 1. selama $\text{node} > b[k]$ maka
 2. $\text{node} = \text{node} - b[k]$
- $\text{cekNode}(\text{node})$
 1. selama node ada pada array tempNode , maka
 2. $\text{node}++$
 3. masuk ke fungsi $\text{cek}(\text{node})$
- $\text{swap}(\text{nodeAwal}, \text{nodeTujuan})$
 1. masukkan nilai nodeAwal ke dalam temp
 2. masukkan nilai nodeTujuan ke dalam nodeAwal

3. masukkan nilai temp ke nodeTujuan

Fungsi 2: Nonlinear substitusi umpan balik. Fungsi ini akan mengubah level skala keabuan dari piksel dengan melakukan bitwise sederhana operasi umpan balik nonlinier, yaitu $f'(x_{l+1}, y_k) = f(x_l, y_k) \text{ XOR } f(x_{l+1}, y_k)$ untuk $k = 0 - (b-1)$ dan $l = 0 - (b-1)$.

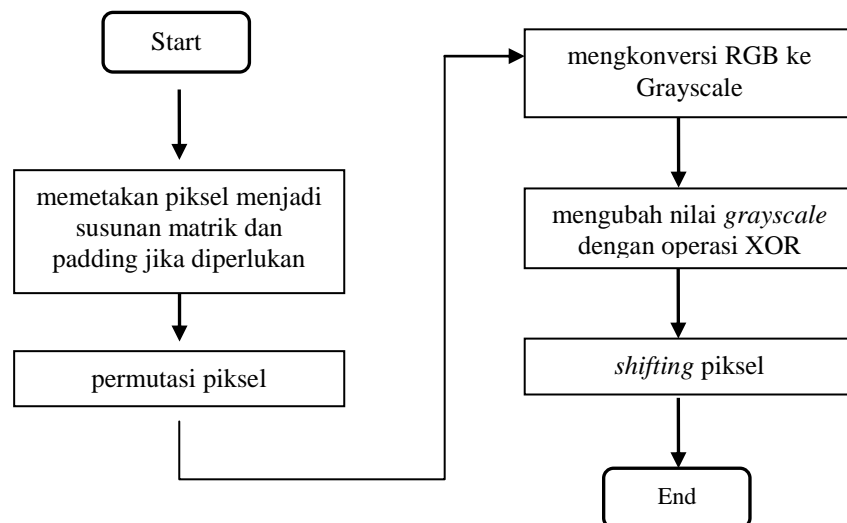
Fungsi 3: Pergeseran piksel dalam baris. Untuk lebih mengacak transposisi dari piksel, piksel pada tiap baris akan diputar ke kiri dengan 0, 1, 2, 3 atau 5 shift tergantung pada nilai modulus (nomor baris).

1. untuk iterasi sebanyak Kr
2. untuk $l = 0$ sampai $b-1$
 - $\text{shift_no} = l \text{ mod } 25$;
 - untuk $k = 0$ sampai $b-1$
 - Rotate-left row(l) sebanyak shift_no

Semua fungsi di atas diulang untuk nomor Kr putaran.

3.1.1 Flowchart Enkripsi

Enkripsi pada sistem keseluruhan aplikasi dapat dilihat pada gambar 3.5



Gambar 3.5 Flowchart enkripsi

Dari gambar 3.5 tahap awal pada saat proses enkripsi adalah memetakan piksel, maksudnya adalah susunan piksel-piksel suatu citra dirubah dalam susunan matrik sehingga matrik array inilah yang diproses dari tahap awal sampai tahap akhir dihasilkannya citra enkripsi. Setelah itu permutasi piksel, pada tahap ini susunan matrik tadi di lakukan proses permutasi piksel atau dengan kata lain dilakukan pengacakan sesuai keteraturan pada program yang dibuat untuk melakukan permutasi piksel tersebut.

Kemudian tahap selanjutnya adalah konversi nilai RGB ke nilai grayscale. Artinya setiap susunan matrik tadi memiliki nilai RGB disetiap index matriknya. Nilai-nilai RGB tersebut dirubah menjadi nilai grayscale sesuai dengan rumus $f = \frac{R+G+B}{3}$ Setelah matrik ini memiliki nilai grayscale maka citra yang diproses pun juga akan menjadi citra grayscale.

Selanjutnya adalah merubah nilai grayscale yang ada dengan cara operasi XOR, perubahan nilai ini dilakukan dengan cara nilai grayscale pada index tertentu di XORkan dengan nilai grayscale index sebelumnya. Dan begitu seterusnya sampai dan selesai.

Tahapan yang terakhir adalah shifting piksel, arti dari shifting piksel ini sendiri adalah menggeser susunan matrik seuai dengan program yang dibuat. Pada proses enkripsi pergeseran dilakukan ke kiri maka jika pada proses dekripsi pergeseran akan dilakukan pada arah sebaliknya.

Pada proses konversi RGB ke *Grayscale* ada rumus tersendiri yang sebelumnya sudah didefinisikan pada awal. Untuk mengubah citra RGB menjadi citra grayscale adalah dengan menghitung rata-rata nilai intensitas RGB dari setiap piksel penyusun citra tersebut. Rumus matematis yang digunakan adalah sebagai berikut:

$$f_0(x, y) = \frac{f_i^R(x, y) + f_i^G(x, y) + f_i^B(x, y)}{3}$$

Contoh penggunaan rumus di atas bisa dilihat pada potongan program berikut:

```

for(int x = 0; x <size.width; x ++) {
    for(int y = 0; y < size.height; y++) {
        int RGB = prosesimage.getRGB(x,y); // memperoleh nilai RGB setiap
        pexel citra
        int alpha = (RGB << 24) & 0xFF; // memperoleh nilai alpha
        int red = (RGB >> 16) & 0xFF; // memperoleh nilai red
        int green = (RGB >> 8 ) & 0xFF; // memperoleh nilai green
        int blue = (RGB >> 0) & 0xFF; // memperoleh nilai blue
        int avg = (red + green + blue) / 3; // menghting nilai rata-rata RGB
        int gray = alpha | avg << 16 | avg << 8 | avg; // memperoleh nilai
        grayscale
        prosesimage.setRGB(x, y, gray); // nilai grayscale untuk setiap piksel
    }
}

```

Sedangkan pada saat proses dekripsi chiper image, tahap untuk kembali merubah grayscale ke RGB tidak dapat didefinisikan. Dengan demikian hasil dari gambar dekripsi adalah gambar grayscale atau berupa gambar hitam putih.

Kemudian untuk selanjutnya adalah mengubah nilai *grayscale* dengan proses XOR, dimana Secara singkat, operasi XOR akan mengembalikan nilai 1 jika jumlah operand bernilai satu ganjil, jika tidak maka akan mengembalikan hasil 0 seperti yang ditunjukkan pada table dibawah ini . Berikut adalah table XOR : *Tabel 3.1. Proses XOR*

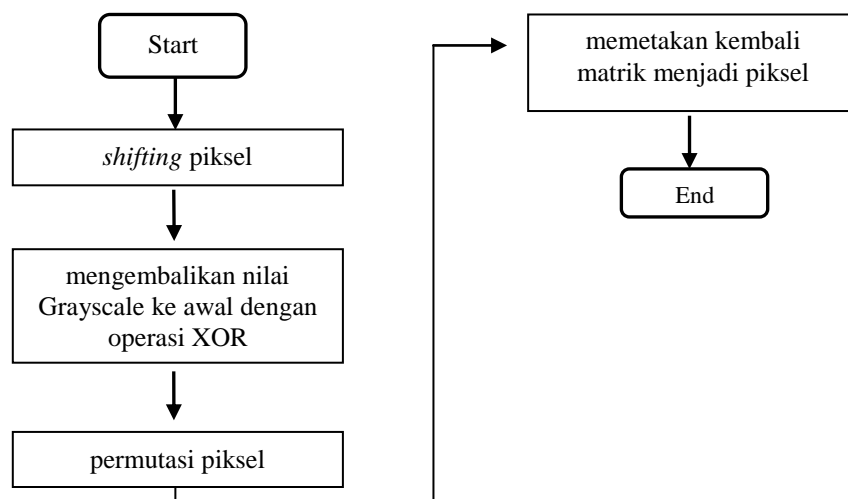
Tabel 3.1. Proses XOR

A	B	$A \oplus B$
1	1	0
1	0	1
0	1	1
0	0	0

Nilai A jika di-XOR-kan dengan nilai B sebanyak dua kali maka akan didapatkan nilai A kembali. Karena sifat istimewa yang dimiliki operasi XOR tersebut sehingga operasi XOR cenderung dipakai dalam proses enkripsi dan dekripsi yang memiliki algoritma yang sama.

3.1.2 Flowchart Dekripsi

Dekripsi pada sistem keseluruhan aplikasi ini dapat dilihat pada gambar 3.7.

**Gambar 3.7** Flowchart dekripsi

Pada proses dekripsi tahapan yang pertama adalah shifting piksel, seperti yang telah dijelaskan di proses enkripsi tadi bahwa ketika proses dekripsi dilakukan pergeseran susunan matrik piksel yang memiliki nilai

grayscale tadi digeser ke kanan sebanyak jumlah pergeseran yang telah dibuat pada program.

Setelah itu tahap selanjutnya adalah mengembalikan nilai grayscale tadi yang telah di XOR dengan cara XOR kembali. Karena untuk mengembalikan nilai XOR itu hanya perlu diXORkan lagi apapun nilainya, sehingga nilai grayscale yang awal dapat diperoleh kembali.

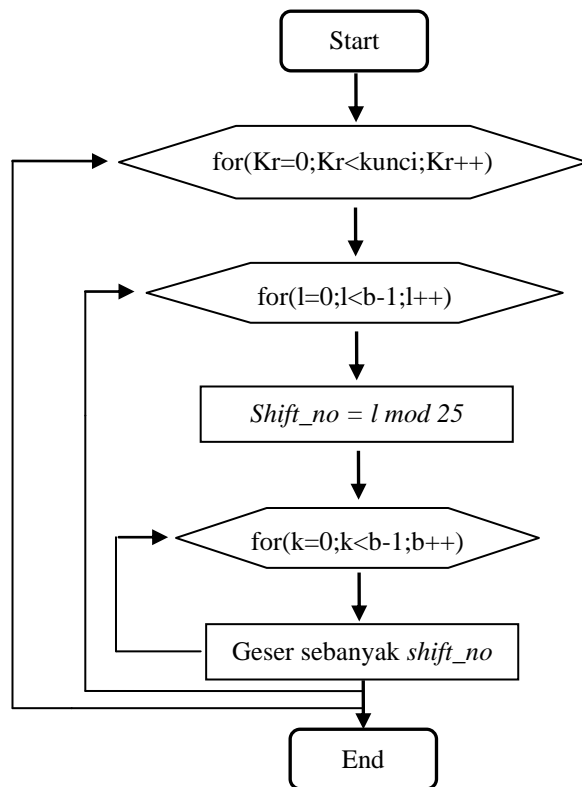
Kemudian permutasi piksel, sama dengan apa yang telah dijelaskan pada proses enkripsi tadi yaitu mengembalikan acakan piksel yang dilakukan pada proses enkripsi sebelumnya. Sehingga akan diperoleh susunan matrik sesuai dengan kondisi semula sebelum terjadi proses permutasi piksel.

Kemudian yang terakhir adalah memetakan kembali susunan matrik tadi menjadi citra asal.

3.1.3 Flowchart Fungsi Pada Enkripsi dan Dekripsi

3.1.3.1 Fungsi Shifting

Fungsi shifting adalah fungsi yang digunakan untuk melakukan pergeseran piksel. Pada fungsi ini telah didefinisikan jumlah pergeserannya. Dalam program jumlah pergeseran didefinisikan dan di inisialisasi dengan nilai modulo 25, jumlah pergeseran yang besar akan lebih tampak hal ini dikarenakan apabila menggeser piksel dengan jumlah yang sedikit maka kemungkinan kecil tidak terlihat perubahan sama sekali pada citra tersebut. Seperti dapat dilihat pada gambar 3.8



Gambar 3.8 Flowchart Fungsi *Shifting*

3.2 Perancangan Aplikasi

Perangkat lunak enkripsi gambar ini dirancang dengan menggunakan beberapa *user interface*, antara lain :

1. *UI* Menu Utama
2. *UI* *Browse File*
2. *UI* Membuat Kunci.
3. *UI* Enkripsi.
4. *UI* Dekripsi.
5. *UI* *Step*
6. *UI* *Help*
7. *UI* *Exit*

Menu yang terdapat pada menu utama adalah dengan rincian sebagai berikut :

1. *Browse File*, digunakan untuk mencari dan memilih file gambar yang akan dienkripsi atau didekripsi. Tipe file yang didukung oleh aplikasi ini hanya file dengan extension .jpg
2. Membuat Kunci, mengisi textbox untuk membuat kunci yang terdiri 4 karakter.
3. Enkripsi, enkripsi gambar berfungsi untuk memproses enkripsi gambar yang nantinya akan menghasilkan gambar chipper.
4. Dekripsi, dekripsi gambar berfungsi untuk memproses dekripsi gambar. Gambar chipper ini nanti yang akan dikembalikan menjadi plain image atau gambar asal.
5. Step, sebenarnya menu ini hanya tambahan yang berfungsi menampilkan tahapan-tahapan yang dilalui baik dalam proses enkripsi atau proses dekripsi. Jadi selain ada menu enkripsi dan dekripsi juga ada menu step untuk membantu menampilkan hasil gambar setiap proses secara visual.
6. Help, menu ini berfungsi untuk membantu pengguna dalam menjalankan aplikasi ini. Prosedur penggunaan aplikasi ini, mulai dari browse file, kemudian mendefinisikan kunci, kemudian memproses baik enkripsi dan dekripsi.
7. Exit, fungsi ini digunakan untuk keluar dari aplikasi.

3.2.1 Desain Menu Utama

Di dalam aplikasi diperlukan menu agar semua fungsi dari aplikasi tersebut dapat diakses dengan cepat oleh pengguna. Berikut tampilan menu seperti pada gambar 3.9

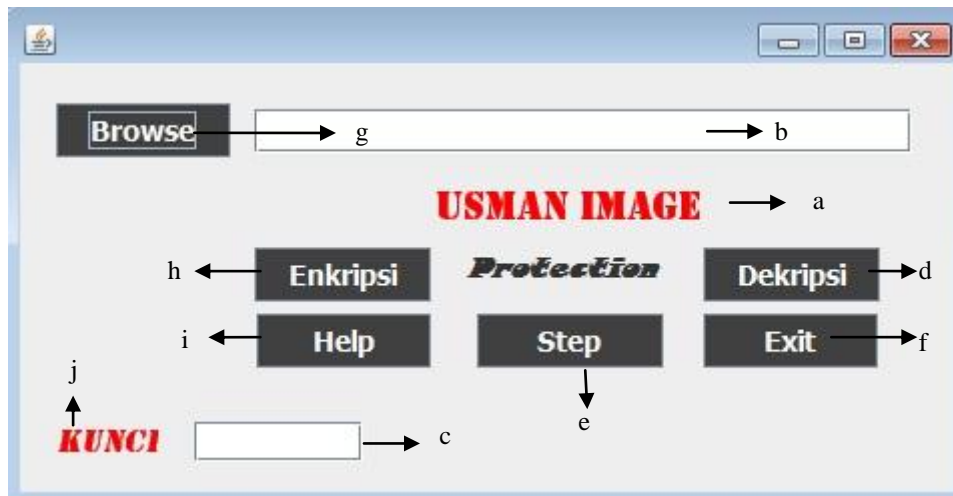


Gambar 3.9 Menu Utama

Keterangan :

1. Browse file, untuk untuk mencari file gambar yang akan dipilih
2. Kunci, untuk memberikan kunci yang digunakan proses enkripsi dan dekripsi.
3. Enkripsi, untuk masuk melakukan enkripsi gambar.
4. Dekripsi, untuk masuk dekripsi gambar
5. Step, untuk menampilkan tahapan setiap proses dalam enkripsi secara visual.
6. Help, untuk menampilkan panduan penggunaan aplikasi
7. Exit, untuk keluar dari aplikasi

Dimana desain dari tampilan seperti diatas adalah rinciannya sebagai berikut seperti yang telah di desain pada program Netbeans pada gambar 3.10



Gambar 3.10 Desain Menu Utama

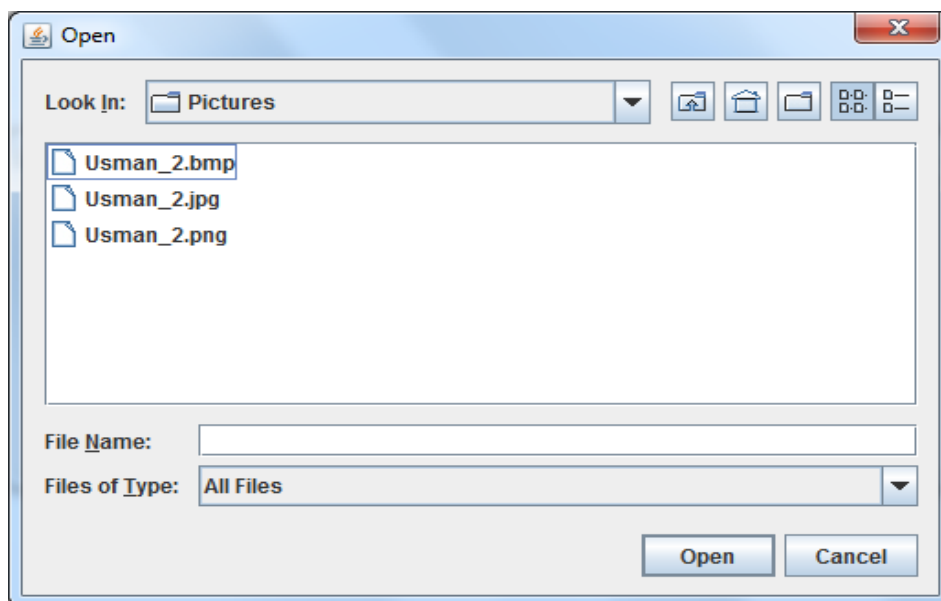
Keterangan :

- a. Label, dengan nama variabel label2. Dimana berisi caption text “Usman Image Protection”
- b. TextField, dengan nama variabel txtPath. Variabel ini digunakan untuk menyimpan path atau menentukan lokasi file yang akan dipilih.
- c. TextField, dengan nama variabel txtKunci. Variabel ini digunakan untuk menyimpan nilai kunci yang dimasukkan untuk enkripsi dan dekripsi.
- d. Button, dengan nama variabel dekrif. Tombol yang digunakan untuk menjalankan proses dekripsi.
- e. Button, dengan nama variabel step. Tombol yang digunakan untuk mengetahui tahapan enkripsi pada aplikasi ini.
- f. Button, dengan nama variabel exit. Tombol yang digunakan untuk keluar dari aplikasi ini.
- g. Button, dengan nama variabel browse. Tombol yang digunakan untuk membuka dan memilih file gambar yang akan di enkripsi dan dekripsi..
- h. JLabel, dengan nama variabel jLabel1. Merupakan caption text yang bertuliskan “Kunci”.

- i. Button, dengan nama variabel enkrip. Tombol yang digunakan untuk menjalankan proses enkripsi.
- j. Button, dengan nama variabel butHelp. Tombol ini digunakan untuk menampilkan panduan yang membantu pengguna dalam menjalankan aplikasi ini. Prosedur penggunaan aplikasi ini, mulai dari browse file, kemudian mendefinisikan kunci, kemudian memproses baik enkripsi dan dekripsi

3.2.3 Desain Browse File

Ketika menekan button browse maka setelah itu akan tampak tampilan seperti pada gambar 3.11



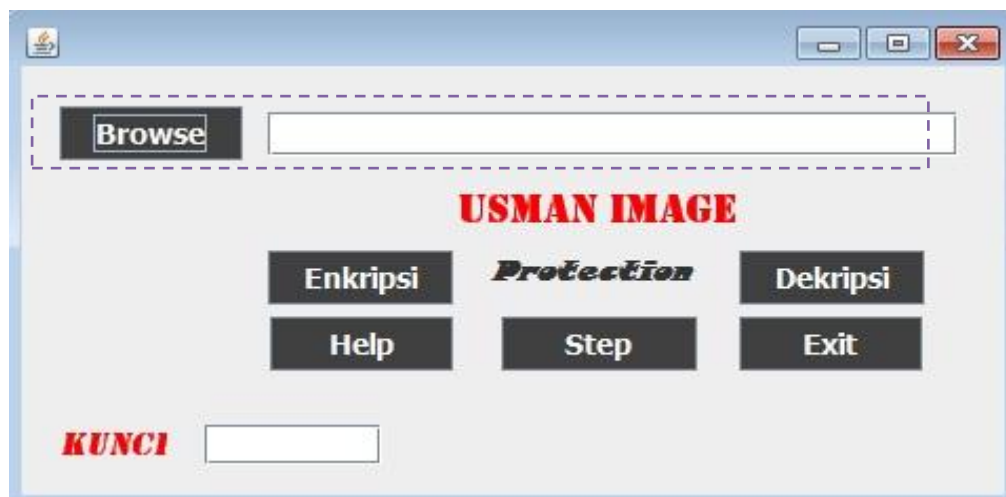
Gambar 3.11 *Open File*

Setelah menekan tombol browse maka akan muncul tampilan open file dialog seperti tampak pada gambar 3.11. Kemudian pada menu utama aplikasi akan tampak path atau lokasi file yang dipilih pada textfield txtPath. Citra dengan nama Usman_2.jpg adalah citraseperti tampak pada gambar 3.12



Gambar 3.12 Gambar asal / plain image Usman_2.jpg

Diatas adalah tampak gambar awal sebelum diproses enkripsi, gambar asli.



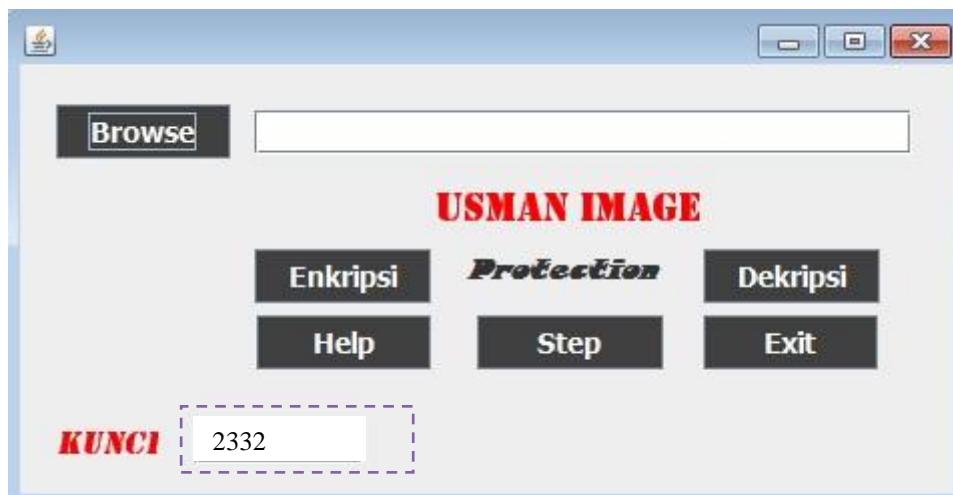
Gambar 3.13 Menu Browse File

Yang ada pada kotak diatas pada gambar 3.13 merupakan path atau lokasi file yang sudah dipilih. Dan nantinya file ini yang akan dienkripsi atau didekripsi. Lokasi file yang kita pilih ada sesuai dengan keinginan kita, mana file gambar yang akan kita enkripsi atau dekripsi.

3.2.3 Desain Pembuatan Kunci

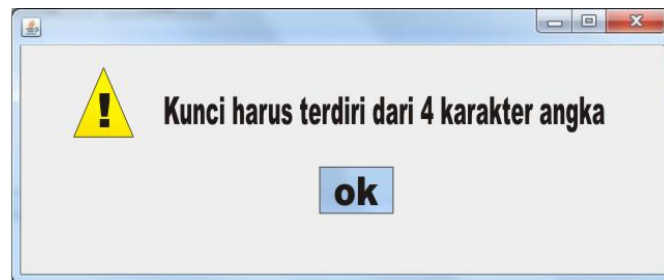
Setelah file sudah dipilih maka tahap selanjutnya adalah membuat kunci untuk enkripsi ataupun dekripsi. Karena kunci dalam aplikasi ini adalah bersifat simetrik maka kunci yang didefinisikan untuk proses enkripsi dan dekripsi adalah harus sama.

Kunci adalah hal yang penting dalam aplikasi ini, hal ini dikarenakan kunci tidak hanya digunakan untuk memulai proses enkripsi dan dekripsi saja tetapi juga berpengaruh pada proses yang terjadi dalam sistem keseluruhan enkripsi atau dekripsi. Kunci terdiri dari karakter angka saja, defini kunci dapat dilihat pada gambar 3.14



Gambar 3.14 Pemberian kunci

Pemberian kunci diatas sangat penting karena apabila kunci tidak didefinisikan maka tidak akan bisa melakukan proses enkripsi dan dekripsi. Apabila ada kesalahan dalam pemberian kunci misal hanya memasukkan 2 karakter angka atau sudah memberikan 4 karakter angka namun dengan kombinasi huruf maka akan muncul peringatan seperti tampak gambar 3.15



Gambar 3.15 Peringatan kunci tidak sesuai

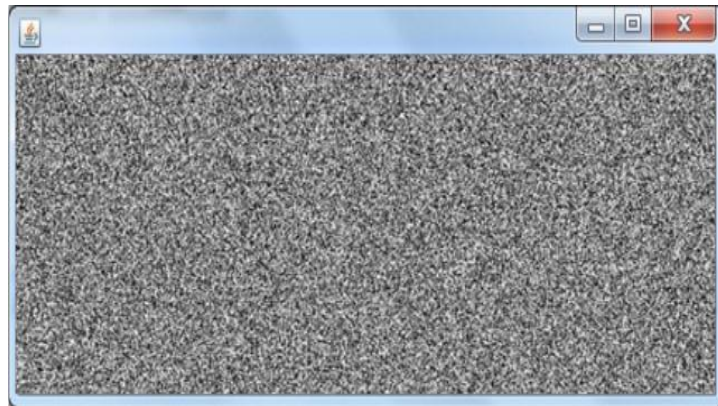
3.2.4 Desain Menu Enkripsi

Setelah menentukan file dan kunci maka selanjutnya dapat melakukan proses enkripsi atau dekripsi, untuk melakukan proses enkripsi langsung saja tekan tombol enkripsi seperti yang ada dalam kotak pada gambar 3.16.



Gambar 3.16 Menu Enkripsi

Setelah itu hasil gambar enkripsi atau chipper image akan tampil seperti pada gambar 3.17



Gambar 3.17 Gambar hasil Enkripsi

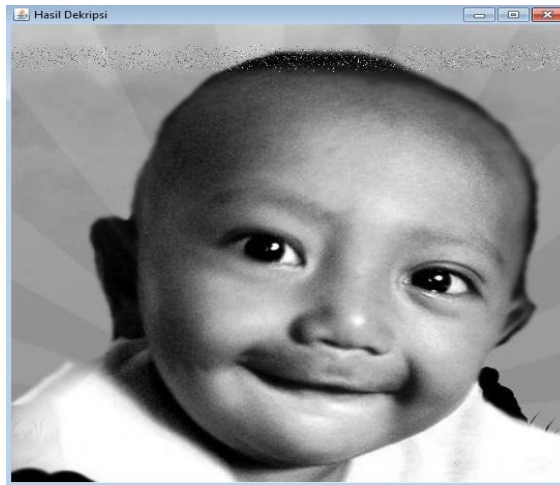
3.2.5 Desain Menu Dekripsi

Setelah menentukan file dan kunci maka selanjutnya dapat melakukan proses enkripsi atau dekripsi, untuk melakukan proses dekripsi langsung saja tekan tombol dekripsi seperti yang ada dalam kotak pada gambar 3.18



Gambar 3.18 Menu Dekripsi

Setelah itu hasil gambar dekripsi atau plain image akan tampil seperti pada gambar 3.19



Gambar 3.19 Gambar hasil dekripsi

Gambar yang dihasilkan dari proses dekripsi adalah gambar hitam putih atau gayscale.

3.2.6 Desain Menu Step

Pada menu ini merupakan tombol yang hanya berfungsi untuk membantu pengguna mengetahui tahapan proses yang terjadi pada proses enkripsi. Namun pada aplikasi yang sudah fix nantinya menu tombol ini tidak akan ada.



Gambar 3.20 Menu step

Tekan button seperti yang ada pada kotak pada gambar 3.20 maka akan tampil setiap step citra yang dihasilkan step pada enkripsi.

3.2.7 Desain Menu Help

Pada menu ini merupakan tombol yang berfungsi untuk memberikan panduan dalam penggunaan aplikasi ini, button help ada dalam kotak seperti pada gambar 3.21



Gambar 3.21 Menu help

Dan untuk seterusnya akan seperti itu, ketika button Help ditekan maka akan muncul panduan untuk pengguna dalam menggunakan aplikasi ini. Panduan prosedural mulai dari cara browse file sampai dan proses enkripsi maupun dekripsinya.

3.2.8 Desain Menu Exit

Pada menu ini merupakan tombol yang berfungsi untuk keluar dari aplikasi yang sedang dijalankan, posisi button exit seperti pada gambar 3.22



Gambar 3.22 Menu exit

Ketika button exit ditekan artinya anda sudah keluar dari aplikasi enkripsi dan dekripsi gambar ini.