

BAB III

ANALISIS DAN PERANCANGAN

III.1. Analisis

Dalam tugas akhir ini, yang dibahas adalah permasalahan mengenai penciptaan piranti lunak yang mampu memanfaatkan sumber daya kamera tanpa menggunakan *driver*.

III.1.1. Analisis Konsep Kerja JMF

Java Media Framework (JMF) sebagai paket API yang memudahkan dalam pengelolaan media dengan Java, salah satunya yaitu sebagai *adapter* dalam mengakses sumber daya perangkat multimedia kamera. Berdasarkan literatur yang diperoleh dari *website* resmi Oracle, secara umum cara kerja JMF adalah sebagai berikut:

1. Cari `CaptureDeviceInfo` yang berada pada protokol "vfw:" atau "vdm:"
2. Jika terdapat device maka simpan lokasi tersebut ke dalam sebuah variabel `MediaLocator`.
3. Panggil *method* `createRealizedPlayer()` dari *class* `Manager` kemudian simpan nilai kembalian kedalam variabel dari *class* `Player`.
4. Tambahkan `Player` kedalam komponen `SwingContainer`.
5. Untuk menghentikan penggunaan *registry camera device* panggil *method* `stop()` dan `deallocate` dari *class* `Player`.

III.1.2. Analisis Kebutuhan Sistem

Pembuatan aplikasi pemampatan citra dengan metode Huffman ini membutuhkan serangkaian peralatan yang dapat mendukung kelancaran proses pembuatan dan pengujian aplikasi. Berikut ini aspek-aspek yang di butuhkan dalam pembuatan aplikasi :

III.1.2.1. Perangkat Keras (*Hardware*)

Hardware merupakan komponen yang terlihat secara fisik, yang saling bekerjasama dalam pengolahan data. Perangkat keras (hardware) yang digunakan meliputi :

1. Monitor 15 ”
2. Processor Pentium IV
3. Harddisk 80GB
4. Memori 1 GB
5. VGA Card ATI 256 Mb
6. *Keyboard* dan *mouse*

III.1.2.2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan penulis untuk aplikasi perancangan aplikasi ini adalah:

1. Sistem Operasi Windows 7
2. Netbeans 7.3 Beta
3. Java SDK dan JRE.

III.1.2.3. Unsur Manusia (*Brainware*)

Brainware adalah manusia yang terlibat dalam mengoperasikan serta mengatur sistem di dalam komputer. Diartikan juga sebagai perangkat intelektual yang mengoperasikan dan mengeksplorasi kemampuan dari *Hardware* maupun *Software*.

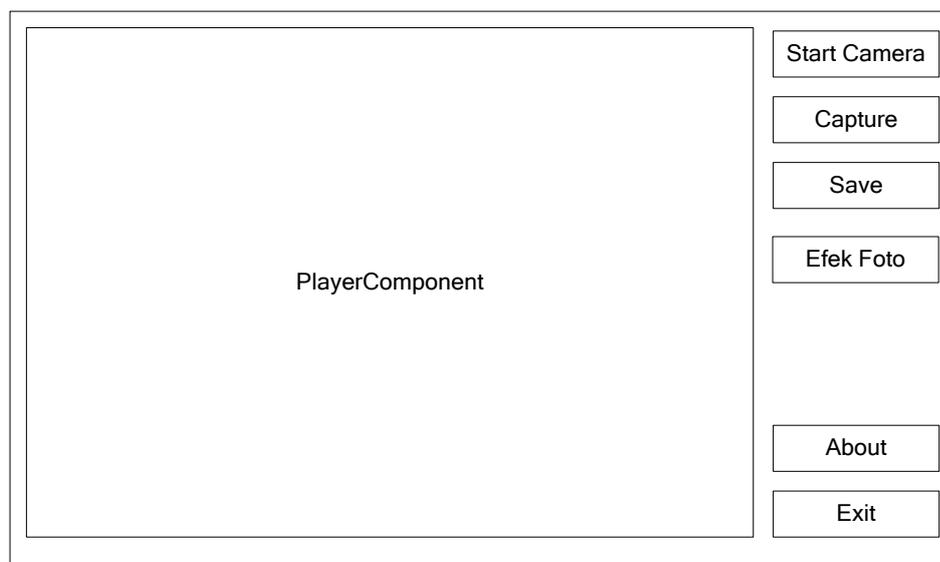
III.2. Perancangan

Perancangan yang dilakukan dalam pembangunan aplikasi webcam ini meliputi perancangan *user interface* dan logika program yang direpresentasikan dengan Flowchart.

III.2.1 Perancangan Antarmuka

III.2.1.1 Antarmuka Utama

Antarmuka utama menyediakan peralatan dan komponen yang dibutuhkan untuk mengontrol aplikasi, rancangan tersebut dilihat pada gambar III.1 dibawah ini:



Gambar III.1 Antarmuka Utama

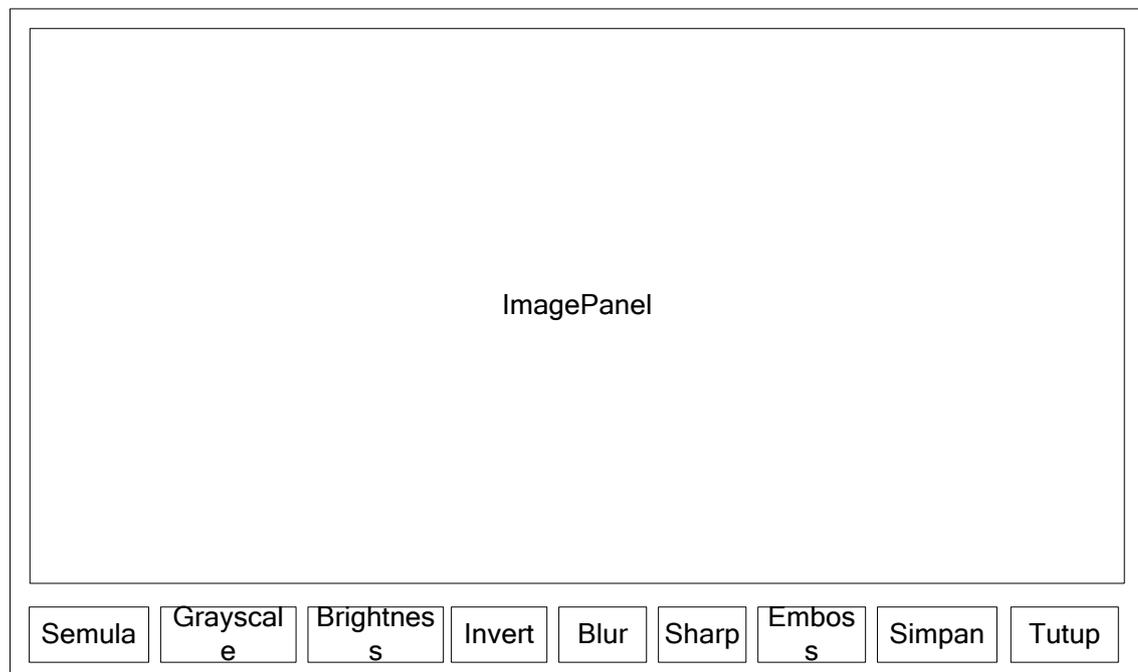
Komponen-komponen yang digunakan untuk antarmuka diatas yaitu sebagai berikut:

Tabel III.1 Komponen Antarmuka Utama

No	Nama Komponen	Kegunaan
1.	JFrame	Untuk menampung semua komponen diatasnya
2.	CameraPlayer	Untuk menampilkan grafik yang di- <i>stream</i>
3.	JButton(Start Camera)	Untuk memulai CameraPlayer
4.	JButton(Capture)	Untuk meng- <i>grab</i> citra CameraPlayer
5.	JButton(Save)	Untuk menyimpan hasil <i>grabbing</i>
6.	JButton(About)	Untuk menampilkan dialog About
7.	JButton(Exit)	Untuk menghentikan CameraPlayer dan keluar

III.2.1.2 Antarmuka Efek Foto

Antarmuka efek foto menyediakan peralatan dan komponen yang dibutuhkan untuk memanipulasi foto yang telah di-*capture*, rancangan tersebut dilihat pada gambar III.2 dibawah ini:



Gambar III.2 Antarmuka Efek Foto

Komponen-komponen yang digunakan untuk antarmuka diatas yaitu sebagai berikut:

Tabel III.2 Komponen Antarmuka Efek Foto

No	Nama Komponen	Kegunaan
1.	JDialog	Untuk menampung semua komponen diatasnya
2.	ImagePanel	Untuk menampilkan foto
3.	JButton(Semula)	Untuk mengembalikan ke foto semula
4.	JButton(Grayscale)	Untuk menfilter hitam putih
5.	JButton(Brightness)	Untuk menambah kecerahan
6.	JButton(Invert)	Untuk membalikan warna
7.	JButton(Blur)	Untuk melembutkan permukaan foto
8.	JButton(Sharp)	Untuk menajamkan foto
9.	JButton(Emboss)	Untuk menambah cekungan tepi foto
10.	JButton(Simpan)	Untuk menyimpan foto
11.	JButton(Tutup)	Untuk menutup jendela

III.2.1.3 Antarmuka Kotak Dialog Tentang Program

Antarmuka kotak dialog tentang program menyediakan antarmuka untuk menampilkan informasi tentang program yang dibuat, rancangan tersebut dilihat pada gambar III.3 dibawah ini:

Tentang Program <Title>

Deskripsi Program

Gambar III.3 Antarmuka Kotak Dialog Tentang Program

Komponen-komponen yang digunakan untuk antarmuka diatas yaitu sebagai berikut:

Tabel III.3 Komponen Antarmuka Utama

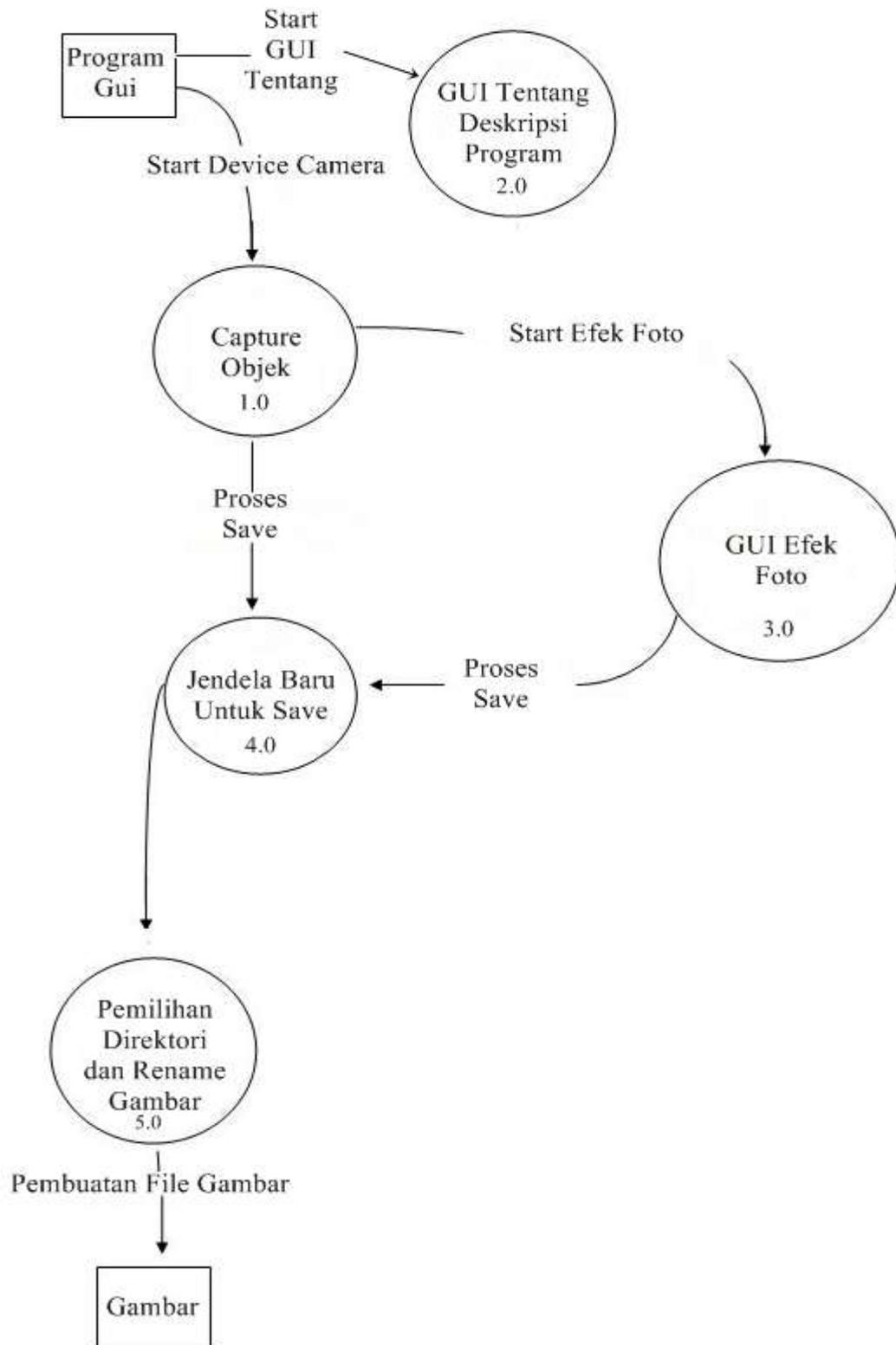
No	Nama Komponen	Kegunaan
1.	JLabel	Untuk menampilkan <i>title</i> About
2.	JLabel	Untuk menampilkan deskripsi program
3.	JSeparator	Untuk memisahkan <i>title</i> dengan deskripsi

III.2.2 Data Flow Diagram

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. DFD merupakan alat bantu dalam menggambarkan atau menjelaskan sistem yang sedang berjalan logis.

III.2.2.1 Latar Belakang Data Flow Diagram

Suatu yang lazim bahwa ketika menggambarkan sebuah sistem kontekstual data flow diagram yang akan pertama kali muncul adalah interaksi antara sistem dan entitas luar. DFD didisain untuk menunjukkan sebuah sistem yang terbagi-bagi menjadi suatu bagian sub-sistem yang lebih kecil dan untuk menggaris bawahi arus data antara kedua hal yang tersebut diatas. Diagram ini lalu "dikembangkan" untuk melihat lebih rinci sehingga dapat terlihat model-model yang terdapat di dalamnya. DFD dari program aplikasi webcam ini dapat dilihat pada gambar III.4 dibawah ini:

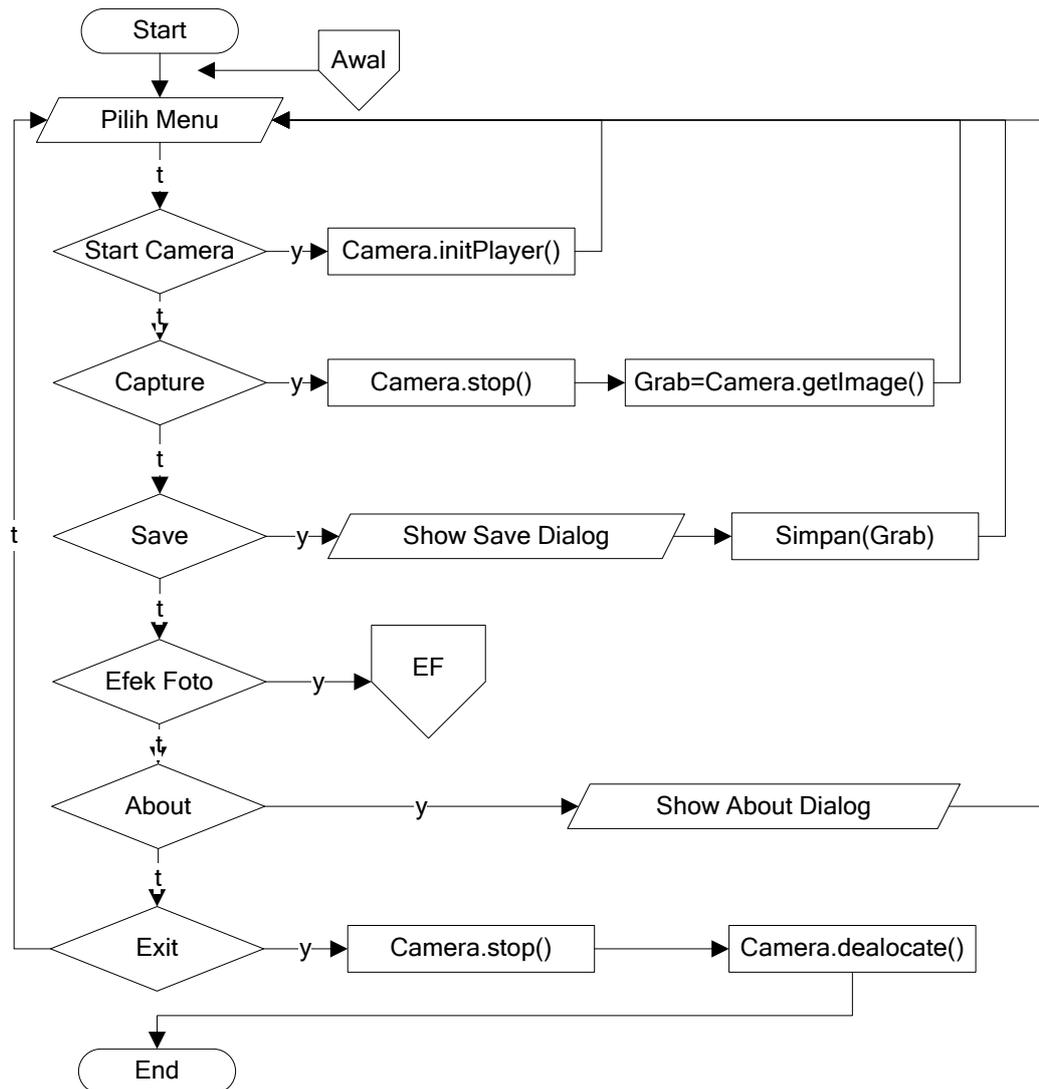


Gambar III.4 Data Flow Diagram

III.2.3 Flowchart

III.2.3.1 Flowchart Utama

Untuk mempermudah pengimplementasian program, diperlukan sebuah logika kerja aplikasi yang dapat dilihat pada Flowchart gambar III.5 dibawah ini:



Gambar III.5 Flowchart Sistem

Pada Flowchart diatas terdapat beberapa *decision* yang membentuk percabangan logika program sebagai berikut:

1. Saat aplikasi dijalankan, pengguna akan disajikan beberapa pilihan aksi.

2. Jika pengguna mengklik tombol Start Camera maka sistem akan mencari CameraDevice yang dapat digunakan kemudian memanggil *method* start() dari variabel CameraPlayer.
3. Jika pengguna mengklik tombol Capture maka sistem akan menghentikan streaming citra dengan memanggil *method* stop() dari CameraPlayer dan melakukan *grabbing* grafik yang selanjutnya disimpan dalam sebuah variabel.
4. Jika pengguna mengklik tombol Save maka sistem akan menampilkan SaveDialog dari *class* JfileChooser kemudian menyimpan citra hasil *grabbing* sesuai dengan lokasi *output stream* yang dipilih pengguna.
5. Jika pengguna mengklik tombol Efek Foto maka sistem akan menampilkan kotak dialog Efek Foto.
6. Jika pengguna mengklik tombol About maka sistem akan menampilkan kotak dialog tentang program.
7. Jika pengguna mengklik tombol Exit maka CameraPlayer akan dihentikan dengan memanggil *method* stop() dan mendealokasikan penggunaan memori dengan memanggil *method* dealocate(), selanjutnya aplikasi dihentikan.

III.2.3.2 Flowchart Efek Foto

Berikut ini adalah rangkaian program untuk menambahkan efek pada foto yang telah di-*capture* dari aplikasi. Flowchart Efek Foto dapat dilihat pada gambar III.6 di bawah ini:



Gambar III.6 Flowchart Efek Foto

EF merupakan nama pengganti notasi referensi flowchart eksternal untuk flowchart Efek Foto, berikut ini adalah penjelasan dari gambar III.5 diatas:

1. Pada kota dialog EF terdapat beberapa tombol yang dapat diklik untuk memanipulasi foto sesuai keinginan, tombol tersebut yaitu tombol Semula, Grayscale, Brightness, Invert, Blur, Sharp, Emboss, Simpan dan Tutup.
2. Jika tombol Semula ditekan maka ImagePanel akan menampilkan citra dari gambar original sebelum dilakukan manipulasi.
3. Jika tombol Grayscale ditekan maka sistem akan membuat sebuah variabel CS yang diinisialisasikan dari kelas ColorSpace. Kemudian sistem membuat variabel operator OP dengan parameter CS, OP memiliki *behavior* yaitu Filter dengan *behavior* tersebut akan menghasilkan citra dalam bentuk *grayscale*.
4. Jika tombol Brightness ditekan maka sistem akan membuat sebuah variabel B yang diinisialisasikan dari kelas Kernel dengan nilai {1,1,1.1}. Kemudian sistem membuat variabel operator OP dengan parameter B, OP memiliki *behavior* yaitu Filter dengan *behavior* tersebut akan menghasilkan citra dengan kecerahan 10% dari citra semula.
5. Jika tombol Invert ditekan maka sistem akan membuat sebuah variabel CS yang diinisialisasikan dari kelas ColorSpace Inverter. Kemudian sistem membuat variabel operator OP dengan parameter CS, OP memiliki *behavior* yaitu Filter dengan *behavior* tersebut akan menghasilkan citra dalam bentuk warna terbalik.

6. Jika tombol Blur ditekan maka sistem akan membuat sebuah variabel B yang diinisialisasikan dari kelas Kernel 3x3 dengan nilai $\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$. Kemudian sistem membuat variabel operator OP dengan parameter B, OP memiliki *behavior* yaitu Filter dengan *behavior* tersebut akan menghasilkan citra yang telah dilembutkan.
7. Jika tombol Sharp ditekan maka sistem akan membuat sebuah variabel S yang diinisialisasikan dari kelas Kernel 3x3 dengan nilai $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$. Kemudian sistem membuat variabel operator OP dengan parameter S, OP memiliki *behavior* yaitu Filter dengan *behavior* tersebut akan menghasilkan citra yang telah dilembutkan.
8. Jika tombol Emboss ditekan maka sistem akan membuat sebuah variabel E yang diinisialisasikan dari kelas Kernel 3x3 dengan nilai $\begin{bmatrix} -2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$. Kemudian sistem membuat variabel operator OP dengan parameter E, OP memiliki *behavior* yaitu Filter dengan *behavior* tersebut akan menghasilkan citra yang telah ditajamkan.
9. Jika tombol Simpan ditekan maka sistem akan menampilkan sebuah kotak dialog untuk memilih lokasi penyimpanan citra. Setelah lokasi terpilih sistem akan membuat file citra dari kelas ImageIO.
10. Jika tombol Tutup ditekan maka sistem akan menutup kotak dialog untuk menambah Efek Foto dan keadaan sistem akan dikembalikan pada keadaan semula seperti saat pertama kali dijalankan.