

BAB IV

PEMBUATAN DAN PENGUJIAN SISTEM

4.1 Pembuatan Sistem

Setelah sistem dianalisa dan dilakukan perancangan secara rinci, maka tahap selanjutnya adalah implementasi. Implementasi merupakan tahap membuat sistem sehingga menjadi aplikasi yang siap untuk dioperasikan. Pada tahapan implementasi terdapat dua cakupan yaitu proses yang meliputi struktur *frame*, *layer*, *coding*, dan hal-hal yang berhubungan dengan aplikasi. Bahasa pemrograman yang digunakan dalam pembuatan aplikasi adalah Action script 3.0.

4.1.1 Implementasi Proses

Setelah pengumpulan bahan materi, gambar, dan suara telah selesai dilakukan, maka akan dilanjutkan pada proses desain ke dalam Adobe Flash CS6. Resolusi layar yang digunakan di sini yaitu 933 x 700 *pixel* dengan *frame rate* 8 dan 12 fps, dan menggunakan bahasa pemrograman *Action Script* 3.0

Tabel 4.1 Tombol Navigasi

No	Gambar Tombol	Keterangan
1		Tombol ini berfungsi untuk masuk ke menu materi
2		Tombol ini berfungsi untuk masuk ke menu simulasi
3		Tombol ini berfungsi untuk masuk ke menu evaluasi

Tabel 4.1 Tombol Navigasi (Lanjutan)

No	Gambar Tombol	Keterangan
4		Tombol ini berfungsi untuk keluar aplikasi
5		Tombol ini berfungsi untuk <i>previous frame</i> dan <i>next frame</i>
6		Tombol ini berfungsi untuk memulai kompresi data dengan menggunakan algoritma run-length encoding
7		Tombol ini berfungsi untuk memulai kembali proses kompresi dari awal.

Aplikasi ini terdiri dari 4 *scene* . Pembagian *movie* ke dalam beberapa *scene* ini berguna untuk memudahkan dalam mengatur tampilan beberapa adegan yang berbeda.

4.1.1.1 *Scene* Menu Utama

Pada *scene* menu utama ini terdapat tombol-tombol yang akan menuju ke *scene* berikutnya. Tombol tersebut yaitu menu materi, menu simulasi, menu evaluasi, dan tombol keluar dari aplikasi. Di halaman ini, terdapat animasi teks judul yang bergerak.



Gambar 4.1 Scene Menu Utama

Dalam *scene* menu utama yang ditunjukkan pada gambar 4.1 di atas, terdapat 5 layer yang memiliki fungsi masing-masing. Layer itu sendiri berguna untuk meletakkan sebuah objek. Selain itu, layer juga dapat digunakan untuk meletakkan barisan-barisan kode *Action Script*. Deskripsi fungsi-fungsi layer tersebut dapat dilihat pada tabel 4.2.

Tabel 4.2 Fungsi Layer pada *Scene* Menu Utama

Layer	Fungsi
<i>Action</i>	Untuk meletakkan <i>script</i> dari masing-masing fungsi tombol yang ada.
Tombol	Untuk meletakkan tombol-tombol pilihan menu yang disajikan.
<i>Movie Clip</i>	Untuk meletakkan objek berupa movie clip.
<i>Paper</i>	Sebagai isi content pada menu-menu.
<i>background</i>	Sebagai gambar latar.

Penulisan *Action Script* 3.0 untuk memberikan perintah pada tombol, tidak dapat dituliskan langsung pada action tombolnya, melainkan dituliskan pada layernya. Perintah *script* berikut ini merupakan aksi yang diberikan kepada tombol-tombol yang ada untuk menuju scene yang dimaksudkan.

Fungsi dari penamaan variabel adalah agar penamaan tersebut dapat dipanggil pada function sesudahnya tentunya penamaan ini tidak boleh sama dengan penamaan lainnya, pada variabel ini berfungsi jika dipanggil nama variabelnya maka perintah ini akan meload file baru, file baru tersebut diterapkan pada *scene* menu materi, *scene* menu simulasi, dan *scene* menu simulasi. Perintah script yang digunakan untuk penamaan variable pada *scene* menu utama adalah sebagai berikut:

```
//Variabel
var loadMateri = new Loader();
var loadSimulasi = new Loader();
var loadEvaluasi = new Loader();
```

Fungsi dari *action script* untuk tombol materi adalah perintah jika tombol materi di tekan, maka perintah itu dapat meload atau mengambil file berupa materi.swf, setelah itu penulis mengatur letak koordinat agar menu materi ini tepat dengan isi *content* pada menu utama. Saat menu materi ini di pilih menu yang lainnya, seperti menu simulasi dan evaluasi akan berhenti dan menutup filenya. Perintah script yang digunakan pada tombol materi pada *scene* menu utama adalah sebagai berikut:

```
//Materi
btn_materi.addEventListener(MouseEvent.CLICK, materi);
function materi (e:MouseEvent)
{
    addChild(loadMateri);
    loadMateri.load(new URLRequest("materi.swf"));
    loadMateri.x = 0;
    loadMateri.y = 0
    loadSimulasi.unloadAndStop();
    loadEvaluasi.unloadAndStop();
}
```

Fungsi dari *action script* untuk tombol simulasi adalah perintah jika tombol simulasi di tekan, maka perintah itu dapat meload atau mengambil file berupa simulasi.swf, setelah itu penulis mengatur letak koordinat agar menu simulasi ini tepat dengan isi *content* pada menu utama. Saat menu simulasi ini di pilih menu yang lainnya, seperti menu materi dan evaluasi akan berhenti dan

menutup filenya. Perintah script yang digunakan pada tombol simulasi pada *scene* menu utama adalah sebagai berikut:

```
//Simulasi
btn_simulasi.addEventListener(MouseEvent.CLICK, simulasi);
function simulasi (e:MouseEvent)
{
addChild(loadSimulasi);
loadSimulasi.load(new URLRequest("simulasi.swf"));
loadSimulasi.x = 0;
loadSimulasi.y = 0
loadMateri.unloadAndStop();
loadEvaluasi.unloadAndStop();
}
```

Fungsi dari *action script* untuk tombol evaluasi adalah perintah jika tombol evaluasi di tekan, maka perintah itu dapat meload atau mengambil file berupa evaluasi.swf, setelah itu penulis mengatur letak koordinat agar menu evaluasi ini tepat dengan isi *content* pada menu utama. Saat menu evaluasi ini di pilih menu yang lainnya, seperti menu materi dan simulasi akan berhenti dan menutup filenya. Perintah script yang digunakan pada tombol simulasi pada *scene* menu utama adalah sebagai berikut:

```
//Evaluasi
btn_evaluasi.addEventListener(MouseEvent.CLICK, evaluasi);
function evaluasi (e:MouseEvent)
{
addChild(loadEvaluasi);
loadEvaluasi.load(new URLRequest("evaluasi.swf"));
loadEvaluasi.x= 0;
loadEvaluasi.y= 0
loadMateri.unloadAndStop();
loadSimulasi.unloadAndStop();
}
```

Fungsi dari *action script* untuk tombol keluar adalah perintah jika tombol keluar di tekan, maka perintah itu dapat menutup semua aplikasi. Perintah script yang digunakan pada tombol keluar pada *scene* menu utama adalah sebagai berikut:

```
//keluar aplikasi
btn_keluar.addEventListener(MouseEvent.CLICK, CloseApp);
function CloseApp(e:MouseEvent)
{
fscommand("quit");
}
```

4.1.1.2 Scene Menu Materi

Pada menu materi, terdapat tombol-tombol yang akan menuju ke halaman materi selanjutnya atau sebelumnya. Pada halaman ini disertai dengan musik latar dan animasi.



Gambar 4.2 Scene Menu Materi

Scene materi pada gambar 4.2, mempunyai 7 layer yang akan dijelaskan fungsi-fungsinya pada tabel 4.3.

Tabel 4.3 Fungsi Layer pada Scene Menu Materi

Layer	Fungsi
Action	Untuk meletakkan <i>script</i> dari masing-masing fungsi tombol yang ada.
Tombol	Untuk meletakkan tombol-tombol pilihan menu yang disajikan.
Text Transition	Untuk menampilkan animasi text.
Isi Materi	Untuk menampilkan isi materi.
Garis	Untuk menampilkan garis berupa tabel.
Judul Materi	Untuk menampilkan judul materi.
Paper	Sebagai isi content pada isi materi

Perintah script yang digunakan pada musik latar di menu materi ini adalah berfungsi sebagai pemutar musik latar jika menu materi dipilih.. Perintah script pemutar music pada menu materi adalah sebagai berikut:

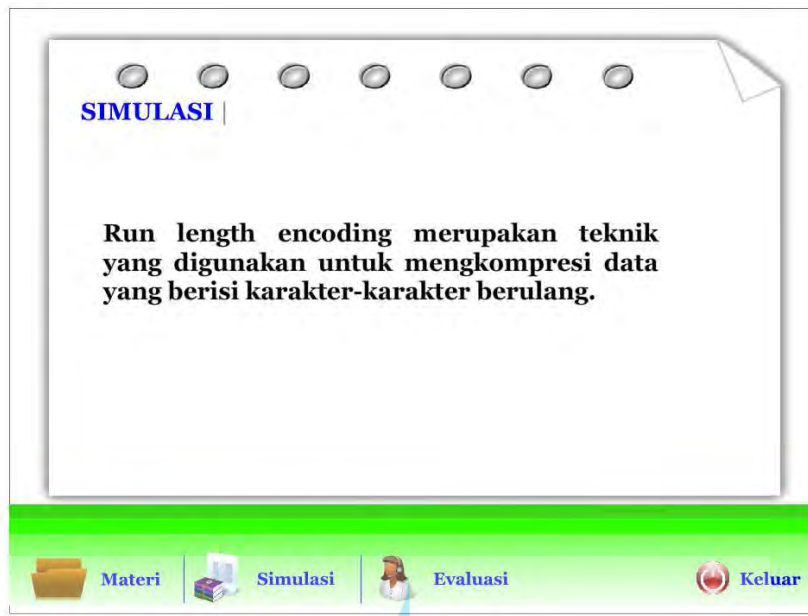
```
var snd:Sound = new Sound();
snd.load(new URLRequest("sound/Kenny.mp3"));
snd.play();
```

Fungsi dari *action script* untuk tombol *next* dan *previous* adalah perintah jika tombol *next* di pilih maka perintah ini akan melakukan ke proses *frame* selanjutnya sedangkan jika tombol *previous* dipilih maka perintah ini akan melakukan ke proses frame sebelumnya. Perintah script yang digunakan pada tombol *next* dan *previous* di menu materi ini adalah sebagai berikut:

```
stop();
nex.addEventListener(MouseEvent.CLICK,onNext);
function onNext(event:MouseEvent)
{
nextFrame()
}
prev.addEventListener(MouseEvent.CLICK,onPrev);
function onPrev(event:MouseEvent)
{
prevFrame()
}
```

4.1.1.3 Scene Menu Simulasi

Pada menu simulasi ini terdapat beberapa contoh simulasi dari materi yang disampaikan seperti proses kompresi dan dekompresi data yang menggunakan algoritma run-length encoding. Pada menu simulasi ini disertai dengan musik latar dan animasi.



Gambar 4.3 Scene Menu Simulasi

Scene simulasi pada gambar 4.3, mempunyai 7 layer yang akan dijelaskan fungsi-fungsinya pada tabel 4.4.

Tabel 4.4 Fungsi Layer pada *Scene* Menu Simulasi

Layer	Fungsi
Action	Untuk meletakkan script
Angka	Untuk menampilkan angka pada bendera
Bendera	Untuk meletakkan gambar bendera
Byte	Sebagai animasi angka yang menyatakan jumlah byte pada setiap warna
Transisi	Sebagai animasi text.
Judul	Untuk menampilkan judul simulasi

Perintah *action script* yang digunakan pada musik latar di menu simulasi ini adalah berfungsi sebagai pemutar musik latar jika menu simulasi dipilih.. Perintah script pemutar musik pada menu simulasi adalah sebagai berikut:

```
var snd:Sound = new Sound();
snd.load(new URLRequest("sound/Depapepe - Start.mp3"));
snd.play();
```

4.1.1.4 Scene Menu Evaluasi

Pada menu evaluasi ini terdapat gambaran tentang proses kompresi data dengan menggunakan algoritma run-length encoding. Dimana gambaran tersebut terdapat beberapa kotak yang dapat diberikan warnadan bagi user sendiri dapat memilih warna dari pallet yang tersedia serta dapat melakukan proses kompresi. Pada menu evaluasi ini disertai dengan musik latar dan animasi.



Gambar 4.4 Scene Menu Evaluasi

Scene evaluasi pada gambar 4.4, mempunyai 5 layer yang akan dijelaskan fungsi-fungsinya pada tabel 4.5.

Tabel 4.5 Fungsi Layer pada *Scene* Menu Evaluasi

Layer	Fungsi
<i>Action</i>	Untuk meletakkan <i>script</i> dari masing-masing fungsi tombol yang ada.
<i>Movie Clip</i>	Untuk meletakkan objek berupa movie clip.
<i>Button</i>	Untuk meletakkan tombol-tombol pilihan menu yang disajikan.
<i>Text</i>	Untuk meletakkan text pada isi content
<i>Paper</i>	Sebagai isi content pada isi evaluasi

Perintah *action script* untuk memberikan warna pada kotak di menu evaluasi ini adalah dengan cara memberikan penamaan variabel warna terpilih sebagai warna yang nantinya di *transform* atau di rubah ketika kotak1 di pilih. Kotak1 merupakan penamaan *instance* di aplikasi flash berupa sebuah objek *movie clip*. Perintah *action script* untuk memberikan beberapa warna pada kotak pada menu evaluasi adalah sebagai berikut:

```
var warna_terpilih:ColorTransform = transform.colorTransform;

//Warna list button
kotak1.addEventListener(MouseEvent.CLICK, warnai);

function warnai(mm:MouseEvent):void
{
mm.target.transform.colorTransform = warna_terpilih;
}
```

Perintah *action script* yang digunakan pada tombol *reset* di menu evaluasi ini berfungsi jika tombol *reset* di tekan, maka isi yang sudah diisi atau di *input* akan dikembalikan seperti semula. Perintah *action script* untuk tombol *reset* pada menu evaluasi adalah sebagai berikut:

```
function hapus(event:MouseEvent):void
{
kotak1.transform.colorTransform = warna;
hasil.text='';
}
reset.addEventListener(MouseEvent.CLICK, hapus);
```

Perintah *action script* yang digunakan pada musik latar di menu evaluasi ini adalah berfungsi sebagai pemutar musik latar jika menu evaluasi dipilih.. Perintah *script* pemutar musik pada menu evaluasi adalah sebagai berikut:

```
var snd:Sound = new Sound();
snd.load(new URLRequest("sound/Relaxing.mp3"));
snd.play();
```

4.1.2 Implementasi Proses Menampilkan Hasil Kompresi

Pada proses ini akan dijelaskan bagaimana proses kompresi dan menampilkan hasil kompresi algoritma *run length encoding* berjalan jika user atau pengguna melakukan evaluasi.

1. *Start*, Algoritma Atau menu evaluasi dijalankan.
2. *Read var warna1:ColorTransform = transform.colorTransform;*
 Pada baris ini adalah untuk mengetahui penamaan warna yang akan di rubah pada proses perubahan warna yang akan dijalankan atau dipanggil sebagai nama variabel.
3. *Read var warna_terpilih:ColorTransform = transform.colorTransform;*
 Pada baris ini menjelaskan bahwa warna_terpilih ini merupakan nama variable yang telah diwarnai.
4. *Read var counter1:int = 0;*
 pada baris ini menjelaskan nama variable untuk perhitungan counter warna.
5. *Read var txtTemp:int = 0;*
6. *Read var lblTemp:String ;*
 pada baris ini menjelaskan nama variable untuk penamaan label atau angka.
7. *Read var colortemp:String ;*
 pada baris ini menjelaskan nama variable untuk penamaan warna
8. *Read var arrayColor:Array = new Array();*
 pada baris ini menjelaskan nama variable yang akan dibentuk menjadi array.
9. *warna1.color = 0xff0000;*
 pada baris ini merupakan penamaan warna1 berbentuk bilangan hexadecimal sebagai warna merah.
10. *merah.transform.colorTransform = warna1;*
 pada baris ini menjelaskan bahwa nama merah merupakan penamaan movie clip untuk warna objek sebagai pallet berwarna merah.
11. *function warnai(mm:MouseEvent):void*
 {
 mm.target.transform.colorTransform = warna_terpilih;
 }

Pada baris ini menjelaskan bahwa warnai merupakan penamaan function yang nantinya akan melakukan perintah mewarnai objek, objek disini berupa movie clip berbentuk kotak.

12. *kotak1.addEventListener(MouseEvent.CLICK, warnai);*

Pada baris ini menjelaskan objek bernama kotak1 sebagai movie clip merupakan target untuk diwarnai.

13. *function ambil(ev:MouseEvent):void*

```
{
    warna_terpilih.color = ev.target.transform.colorTransform.color;
    trace(warna_terpilih);
}
```

Pada baris ini merupakan function pengambilan warna pada pallet yang telah terpilih dan mengeluarkan warna sesuai yang dipilihnya.

14. *merah.addEventListener(MouseEvent.CLICK, ambil);*

pada baris ini merupakan action list jika mouse di klik warnanya maka akan terpilih warna merah.

15. *function hapus(event:MouseEvent):void*

```
{
    kotak1.transform.colorTransform = warna;
    hasil.text="";
}
```

Pada baris ini menjelaskan fungtion reset agar dapat mereset ulang program yang sebelumnya telah dijalankan.

16. *reset.addEventListener(MouseEvent.CLICK, hapus);*

Pada baris ini menjeelaskan action list button reset, jika button reset ini ditekan maka akan menjalankan function hapus. Maka proram akan mereset ulang .

17. *function findbyName(value: *):String*

```
{ if(value == warna1.color.toString() )
    {return "M";}
    else {return "P";}
}
```

Pada baris ini merupakan penamaan nama function output yang nantinya ketika function ini di jalankan akan menghasilkan warna1 berupa M, M merupakan warna merah dan P merupakan warna putih.

```

18. function submitName(evt:MouseEvent){
    hasil.text = "";
    arrayColor.splice(0);
    arrayColor.push(kotak1.transform.colorTransform.color);
    lblTemp = "0";
    counter1=0;
    txtTemp = arrayColor[0]
    for(var i:int = 0; i<arrayColor.length+1; i++){
        //lblTemp = lblTemp + arrayColor[i].toString();
        colortemp = findbyName(arrayColor[i])
        if(arrayColor[i] != txtTemp){
            if(lblTemp == "0"){
                if(counter1 != 1){
                    lblTemp = counter1.toString() + findbyName(txtTemp)
                }else{
                    lblTemp = findbyName(txtTemp)
                }
            }
        }
        Else
        {
            if(counter1 != 1){
                lblTemp = lblTemp + "+"counter1.toString() + findbyName(txtTemp)
            }else{
                lblTemp = lblTemp + "+" findbyName(txtTemp)
            }
        }
        txtTemp = arrayColor[i];
        counter1 = 1;
    }else{

```

```

    counter1 = counter1 + 1;
    txtTemp = arrayColor[i];
}
}
hasil.text = lblTemp;
}

```

Pada baris ini merupakan function submitname , yaitu dimana proses kompresi berjalan.

19. `kompres.addEventListener(MouseEvent.CLICK,submitName);`

Pada baris ini merupakan action list button submit untuk menjalankan function submitname.

20. *End*, Algoritma Atau menu evaluasi berakhir.

4.2 Pengujian Sistem

Pengujian aplikasi simulasi kompresi *run length encoding* dilakukan untuk mendapatkan hasil bahwa aplikasi yang telah dibuat ini dapat berjalan dengan baik dan sesuai dengan yang direncanakan sebelumnya. Pengujiannya dilakukan dengan metode pengujian *Black Box*, yaitu pengujian terhadap fungsi-fungsi antarmuka yang terdapat dalam aplikasi.

Dalam pengujian ini, dilakukan pengujian tombol-tombol yang ada pada aplikasi yaitu dapat berjalan sesuai dengan fungsinya (navigasi, suara). Selanjutnya pengujian dilakukan ketika aplikasi tersebut sedang berlangsung.

4.2.1 Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut:

1. Microsoft Windows 7
2. Adobe Flash CS6
3. Adobe Photoshop CS6
4. Adobe Flash Player 13.0

4.2.2 Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut:

1. Prosesor : Intel® Core (TM)2 Duo CPU T6500 @2.10GHz
2. RAM : 1.00 GB
3. Hardisk : 150 GB
4. LCD : LCD 14" TFT Color LCD
5. Optical Drive : DVD RW 16x

4.2.3 Skenario Pengujian

Skenario pengujian ini digunakan untuk menentukan langkah-langkah dalam melakukan pengujian. Dilakukan dengan menjalankan aplikasi simulasi kompresi *run length encoding*, kemudian dilakukan pengujian dengan mengklik tombol-tombol yang terdapat pada setiap *scene*.

Tabel 4.6 Skenario Pengujian

No	Hal yang diuji	Cara menguji	Hasil yang diharapkan
1	Halaman menu utama	Menekan tombol-pilihan menu	Menampilkan halaman menu yang dituju
		Menekan tombol keluar aplikasi	Aplikasi tertutup
2	Halaman menu materi	Menekan tombol gambar materi	Masuk ke halaman materi
		Menekan tombol <i>next</i> dan <i>prev</i>	Masuk ke halaman materi kompresi, dan user dapat melihat materi selanjutnya atau sebelumnya
3	Halaman simulasi	Menekan tombol gambar simulasi	Masuk ke halaman simulasi
4	Halaman evaluasi	Menekan tombol gambar evaluasi	Masuk ke halaman evaluasi
		Mengklik pallet warna	Pallet warna terpilih oleh <i>cursor mouse</i>
		Mengklik/memberi warna kotak-kotak kompresi	Kotak- kotak kompresi dapat diberi warna
		Mengklik tombol kompresi	Menampilkan hasil kompresi <i>run length encoding</i>
		Mengklik tombol reset	Menghapus data sebelumnya dan memulai proses kompresi dari awal.

4.2.4 Hasil Pengujian

Dari skenario pengujian sebelumnya, didapatkan hasil pengujian sebagai berikut :

Tabel 4.7 Hasil Pengujian

No	Hal yang diuji	Cara menguji	Hasil yang diharapkan	Hasil pengujian	Penjelasan
1	Halaman menu utama	Menekan tombol-tombol pilihan menu	Menampilkan halaman menu yang dituju	Sesuai	Masuk ke halaman yang dituju, dengan suara latar yang baru
		Menekan tombol keluar aplikasi	Aplikasi tertutup	Sesuai	Aplikasi sudah tidak tampil lagi
2	Halaman menu materi	Menekan tombol gambar materi	Masuk ke halaman materi	Sesuai	Terdapat materi berupa kompresi, dekompresi, dan <i>run length encoding</i> , serta suara latar yang baru
		Menekan tombol <i>next</i> dan <i>prev</i>	Masuk ke halaman materi kompresi, dan user dapat melihat materi selanjutnya atau sebelumnya	Sesuai	Terdapat materi yang berbeda dan dapat melihat materi sebelumnya
3	Halaman simulasi	Menekan tombol gambar simulasi	Masuk ke halaman simulasi	Sesuai	Tampil simulasi kompresi berupa gambar bendera merah putih dan suara latar yang berbeda
4	Halaman evaluasi	Menekan tombol gambar evaluasi	Masuk ke halaman evaluasi	Sesuai	Tampil user interface proses kompresi
		Mengklik pallet warna	Pallet warna terpilih oleh <i>cursor mouse</i>	Sesuai	Warna yang dipilih dapat berfungsi dengan baik
		Mengklik/memberi warna kotak-kotak kompresi	Kotak-kotak kompresi dapat diberi warna	Sesuai	Kotak-kotak kompresi akan berisi warna yang dipilih sesuai dengan pallet warna yang dipilih.

Tabel 4.7 Hasil Pengujian (Lanjutan)

No	Hal yang diuji	Cara menguji	Hasil yang diharapkan	Hasil pengujian	Penjelasan
		Mengklik tombol kompresi	Menampilkan hasil kompresi <i>run length encoding</i>	Sesuai	Hasil output kompresi berupa algoritma run-length encoding
		Mengklik tombol reset	Menghapus data sebelumnya dan memulai proses kompresi dari awal.	Sesuai	Memulai kembali proses kompresi dari awal

Pada tabel 4.7 memperlihatkan hasil pengujian atas fungsi aplikasi. Dimana hasil yang diharapkan pada tiap-tiap tombol aplikasi berfungsi dengan baik dan sesuai dengan hasil pengujian.

4.2.5 Analisis Hasil Pengujian

Setelah dilakukan pengujian pada aplikasi simulasi kompresi algoritma *run length encoding* ini, maka dapat dilakukan beberapa analisa sebagai berikut :

1. Pada pengujian halaman materi menampilkan beberapa materi yang terdiri dari kompresi, dekompresi dan algoritma *run length encoding*.
2. Pada halaman simulasi, user dapat melihat proses simulasi kompresi dan dekompresi algoritma *run length encoding* berupa gambar bendera merah putih.
3. Pada halaman evaluasi, user harus memilih pallet warna dan memberi warna pada kotak-kotak kompresi yang tersedia sehingga user dapat melakukan proses kompresi.

