

BAB III

ANALISIS DAN PERANCANGAN

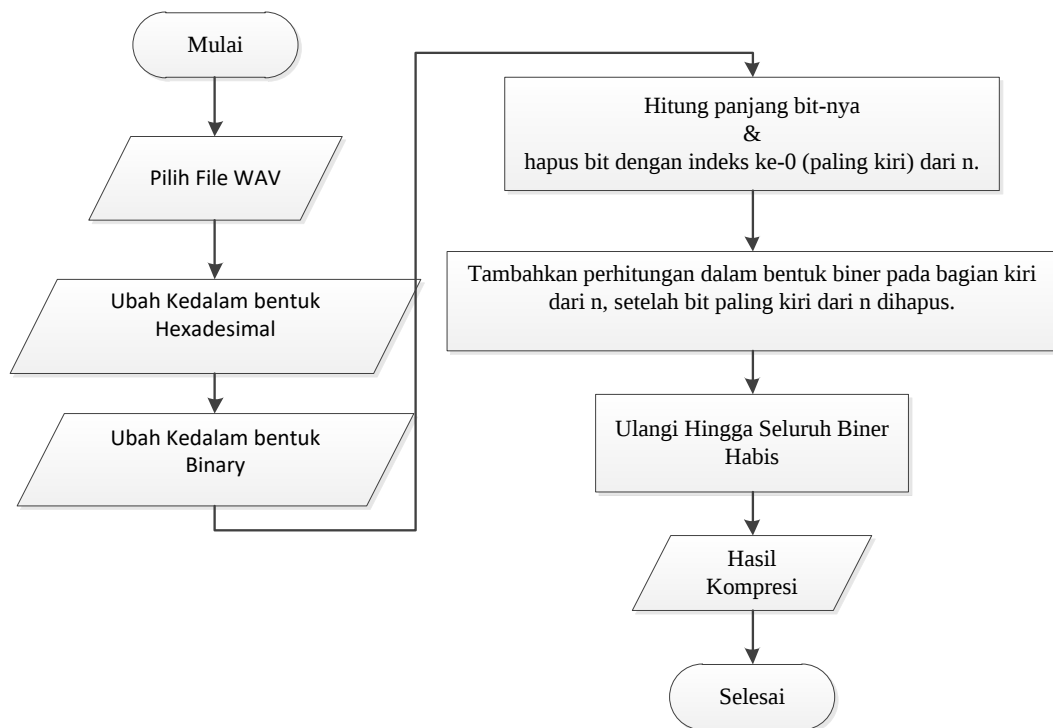
III.1. Analisis Masalah

Salah satu file audio yang banyak dipergunakan dalam sistem operasi *Windows* adalah dalam bentuk format WAV, format ini banyak dipergunakan untuk keperluan game dan multimedia. Pada dasarnya file WAV adalah merupakan format RAW (format kasar), dimana sinyal suara langsung direkam dikuantisasi menjadi data digital. Format dasar dari file ini secara *default* tidak mendukung kompresi dan dikenal dengan nama PCM (*Pulse Code Modulation*).

Jika dilakukan rekaman sebuah lagu dengan kualitas CD Audio dengan menggunakan *sampling rate* 44.1 kHz, 16 bit per *sample* serta menggunakan 2 kanal audio (*stereo*), maka total media penyimpanan yang dibutuhkan untuk menyimpan data audio ini per detik adalah 176.400 *byte* sehingga untuk durasi 1 menit akan diperlukan 10,584 Mb. Jika rata-rata durasi satu lagu selama adalah 5 menit maka akan dibutuhkan ruang penyimpanan sebesar lebih dari 50 Mb, ini hanya contoh kecil dari gambaran ukuran file WAV, bagaimana jika durasi dari sebuah file WAV itu memiliki durasi yang panjang, tentunya akan membutuhkan ruang penyimpanan yang jauh lebih besar. Oleh karenanya dibutuhkan sebuah aplikasi kompresi khusus file WAV yang dapat melakukan proses kompresi yang dapat memperkecil ukuran file tanpa mengubah isi dari file WAV tersebut.

III.1.1. Flowchart Metode

Adapun flowchart dari penerapan metode Elias Delta Code adalah sebagai berikut :



Gambar III.1. Flowchart Metode *Elias Delta Code*

III.1.2 Penerapan Algoritma Elias Delta Code

Berikut ini adalah penerapan algoritma Elias Delta Code pada aplikasi yang akan dirancang.

1. Menentukan file WAV yang akan dikompresi.
2. Mengubah bentuk file kedalam bentuk *hexadecimal*, pada saat melakukan konver file wav menjadi *hexadecimal* dipergunakan aplikasi bantu yaitu *binary viewer* dan diperoleh hasil :

Addr...	Text (ASCII)	Hexadecimal
00000000	... f t y p	00 00 00 18 66 74 79 70
00000008	m p 4 2 ...	6D 70 34 32 00 00 00 00
00000010	i s o m m p 4 2	69 73 6F 6D 6D 70 34 32
00000018	... \$ 1 I m o o v	00 24 31 49 6D 6F 6F 76
00000020	... l m v h d	00 00 00 6C 6D 76 68 64
00000028	... 9 d ... 9 d ...	00 00 00 00 D8 39 64 14
00000030	9 d ... 2 ...	D8 39 64 14 00 00 32 00
00000038	... 4 > ...	04 AE 34 3E 00 01 00 00
00000040		01 00 00 00 00 00 00 00
00000048		00 00 00 00 00 01 00 00
00000050		00 00 00 00 00 00 00 00
00000058		00 00 00 00 00 01 00 00

Gambar III.2 Hasil konversi dengan menggunakan Binary Viewer

Untuk kebutuhan perhitungan maka nilai *hexadecimal* diambil dari sisi kiri baris ke-3 bilangan ke-25 sampai dengan baris ke-7 bilangan ke-49 , yaitu 00, 24, 31, 49, 6D, 6F, 6F, 76, 00, 00, 00, 6C, 6D,76, 68, 64, 00, 00, 00, 00, D8, 39, 64, 14, D8

- Selanjutnya dilakukan proses konversi hexadesimal menjadi binary, sehingga diperoleh :

Tabel 3.1 Konversi Hexadesimal Ke Binary

Nilai	
Hexadecimal	Binary
00	00000000
24	00100100
31	00110001
49	01001001
6D	01101101
6F	01101111
6F	01101111
76	01101110
00	00000000
00	00000000
00	00000000
6C	01101100
6D	01101101

76	01110110
68	01101000
64	01100100
00	00000000
00	00000000
00	00000000
00	00000000
D8	11011000
39	01101000
64	01100100
14	00010100
D8	11011000

4. Tahap selanjutnya adalah pembentukan set karakter. Karakter yang diulang akan dihilangkan sehingga hanya satu karakter yang hidup.

Hasil dari proses set karakter dapat dilihat pada tabel berikut ini:

Tabel 3.2 Pembentukan Set Karakter

Nilai			
Hexadecimal	Binary	Freq	Bits x Freq
00	00000000	8	64
24	00100100	1	8
31	00110001	1	8
49	01001001	1	8
6D	01101101	2	16
6F	01101111	2	16
76	01110110	2	16
6C	01101100	1	8
68	01101000	18	8
64	01100100	2	16
D8	11011000	2	16
39	01101000	1	8
14	00010100	1	8
Total			200

Berdasarkan tabel 3.2 di atas, satu nilai pixel hexadesimal (karakter) bernilai 8 bit bilangan biner. Sehingga 13 bilangan hexadesimal mempunyai nilai biner sebanyak 200 bit. Untuk mengubah satuan menjadi byte maka jumlah keseluruhan bit dibagi 8. Maka dihasilkan $200/8 = 25$ byte.

5. Pembentuk kode Elias Delta Code

Aturan dalam pembentukan kode bilangan dengan menggunakan Elias Delta Code dapat dilihat pada landasan teori pada bab sebelumnya. Adapun nilai dari Elias Delta Code dapat dilihat pada tabel berikut:

Tabel 3.3 Proses pengkodean algoritma Elias Delta Code.

Hexa	Keterangan	Kode Delta
00	1	1
64	2 = 10 ₂ (2 bit), Hapus angka 1 paling kiri dan tambahkan 2 = 10 ₂ (2 bit), sehingga menjadi 10 0. 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 0100.	0100
6D	3 = 11 ₂ (2 bit), Hapus angka 1 paling kiri dan tambahkan 2 = 10 ₂ (2 bit), sehingga menjadi 10 1. 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 0101	01110
6F	4 = 100 ₂ (3 bit), Hapus angka 1 paling kiri dan tambahkan 3 = 11 ₂ (2 bit), sehingga menjadi 11 00. 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 01100.	01101
76	5 = 101 ₂ (3 bit), Hapus angka 1 paling kiri dan tambahkan 3 = 11 ₂ (2 bit), sehingga menjadi 11	01110

	01. 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 01101.	
D8	6 = 1102 (3 bit), Hapus angka 1 paling kiri dan tambahkan 3 = 11 (2 bit), sehingga menjadi 11 10. 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 01110.	01111
14	7 = 1112 (3 bit), Hapus angka 1 paling kiri dan tambahkan 3 = 11 (2 bit), sehingga menjadi 11 11. 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 01111.	01111
24	8 = 10003 (4 bit), hapus angka 1 paling kiri dan tambahkan 4 = 100 (3 bit), sehingga menjadi 100 000. 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100000.	00100001
31	9 = 10013 (4 bit), hapus angka 1 paling kiri dan tambahkan 4 = 100 (3 bit), sehingga menjadi 100 001. 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100001.	00100010
39	10 = 10103 (4 bit), hapus angka 1 paling kiri dan tambahkan 4 = 100 (3 bit), sehingga menjadi 100 010. 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100010.	00100011
49	11 = 10113 (4 bit), hapus angka 1 paling kiri dan tambahkan 4 = 100 (3 bit), sehingga menjadi 100 011. 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100011.	00100011
68	12 = 11003 (4 bit), hapus angka 1 paling kiri dan tambahkan 4 = 100 (3 bit), sehingga menjadi 100 100.	00100100

	3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100100.	
6C	13 = 11013 (4 bit), hapus angka 1 paling kiri dan tambahkan 4 = 100 (3 bit), sehingga menjadi 100 101. 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100101	00100101

6. Proses kompresi untuk algoritma Elias Delta Code

Tabel 3.4 Proses pengkodean algoritma Elias Delta Code.

Nilai					
Hexadecimal	ASCII	Frequensi	Elias Delta Code	Length	Bits
00	0	8	1	1	8
64	100	2	0100	4	8
6D	109	2	0101	4	8
6F	111	2	01100	5	10
76	118	2	01101	5	10
D8	216	2	01110	8	10
14	20	1	01111	5	5
24	36	1	00100000	8	8
31	49	1	00100001	8	8
39	57	1	00100010	8	8
49	73	1	00100011	8	8
68	104	1	00100100	8	8
6C	108	1	00100101	8	8
Total					107

III.2. Perancangan

Berikut ini perencanaan dari aplikasi yang akan dirancang. Desain aplikasi pada penelitian ini dibagi menjadi dua desain, yaitu desain sistem secara global untuk penggambaran model sistem secara garis besar dan desain sistem secara detail untuk membantu dalam pembuatan sistem.

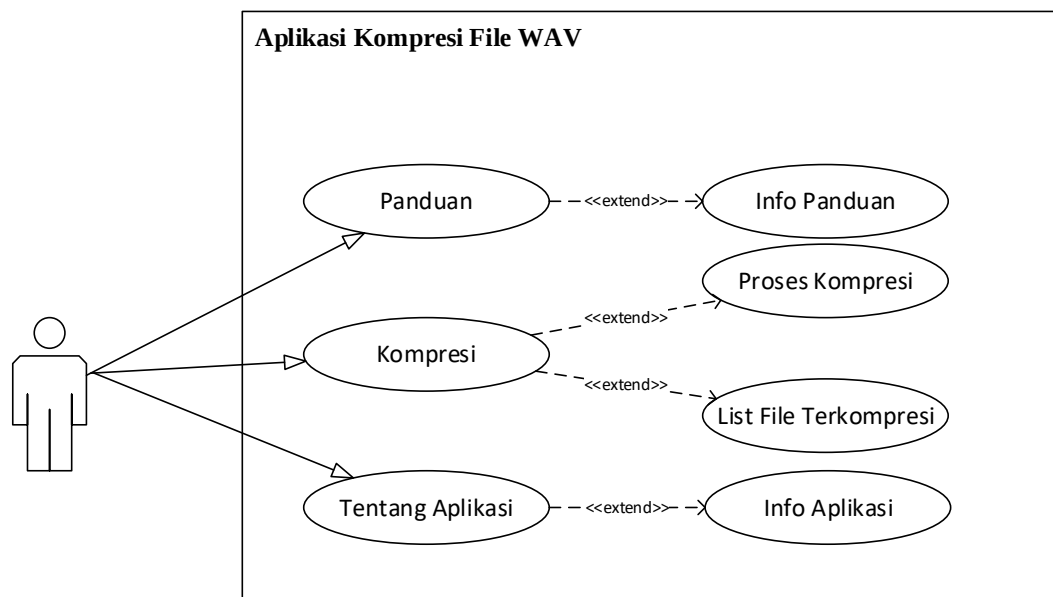
III.2.1.Desain Sistem

III.2.2.Desain Sistem Secara Global

Desain sistem secara global menggunakan bahasa pemodelan UML yang terdiri dari *Usecase Diagram*, *Activity Diagram* dan *Sequence Diagram*.

III.2.2.1. Usecase Diagram

Diagram pertama pada bagian desain ini adalah diagram *usecase*, dimana dalam diagram menggambarkan interface utama dari aplikasi yang saling berhubungan, seperti yang terlihat pada gambar III.2. berikut ini :



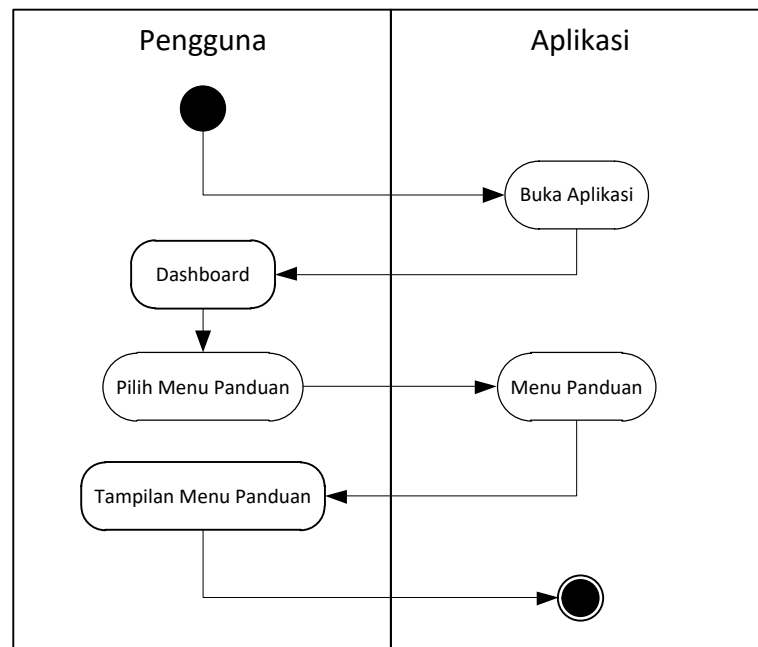
Gambar III.2. Use Case Diagram Aplikasi Kompresi File WAV

III.2.2.2. Activity Diagram

Bisnis proses yang telah digambarkan pada *use case diagram* dijabarkan dengan *Activity diagram* :

1. *Activity Diagram* Menu Panduan

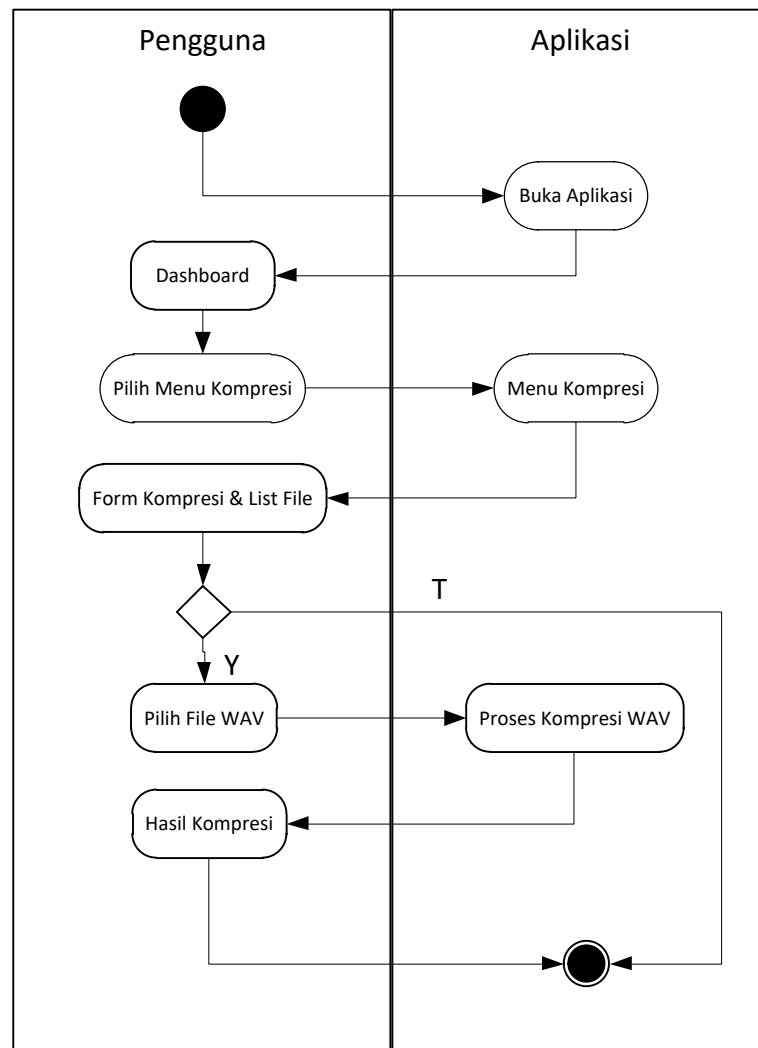
Activity Diagram berikut menggambar tahapan atau kegiatan yang berlangsung jika pengguna aplikasi mengakses memilih menu panduan.



Gambar III.3. Activity Diagram Menu Panduan

2. Activity Diagram Menu Kompresi

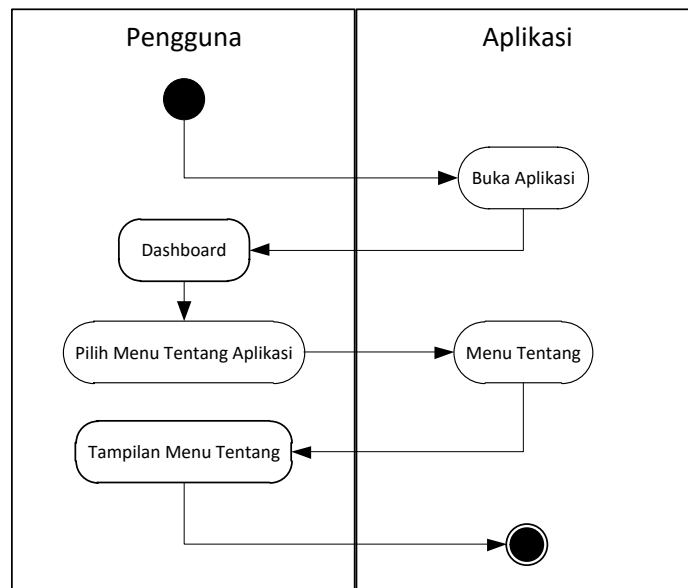
Aktivitas berikut ini adalah merupakan aktifitas yang berjalan jika pengguna aplikasi memilih menu kompresi filr WAV, seperti yag dapat dilihat pada gambar III.4 berikut ini :



Gambar III.5. Activity Diagram Menu Kompresi File WAV

3. Activity Diagram Menu Tentang

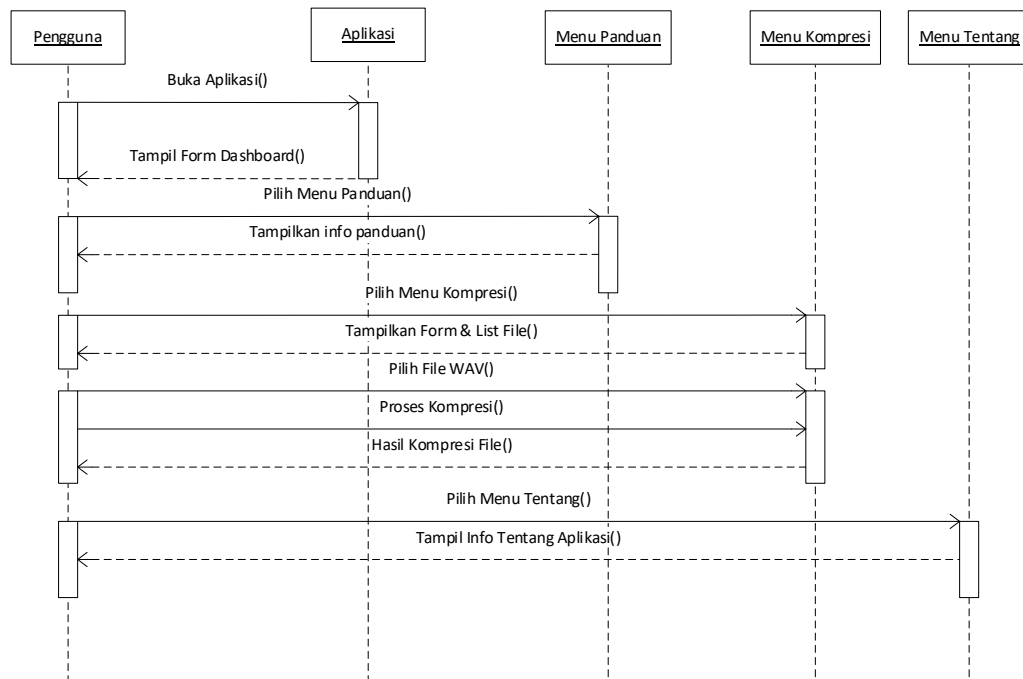
Aktivitas berikut akan menggambarkan kegiatan yang berlangsung jika pengguna dari aplikasi memilih menu Tentang aplikasi, untuk tahapannyadapat terlihat pada gambar III.6 berikut ini :



Gambar III.6. Activity Diagram Menu Tentang Aplikasi

III.3.2.3. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah – langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menghasilkan output tertentu. *Sequence* Diagram diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.



Gambar III.7. Sequence Diagram Aplikasi Kompresi File WAV

III.4. Desain Sistem Secara Detail

Tahap perancangan berikutnya yaitu desain sistem secara detail yang meliputi desain sistem.

1. Tampilan *Dashboard*

Dashboard adalah merupakan tampilan yang akan ditampilkan diawal aplikasi, pada tampilan *dashboard* ini terdapat 3 buah menu yang memiliki fungsi masing-masing, seperti Menu Panduan Aplikasi, Menu Kompresi File dan Menu Tentang Aplikasi, untuk desain tampilannya dapat dilihat pada gambar III.8 berikut ini :



Gambar III.8. Tampilan *Dashboard*

2. Tampilan *Menu* Panduan Penggunaan

Tampilan berikut ini adalah tampilan yang akan tampil jika pengguna memilih menu panduan aplikasi. Didalam menu tampilan aplikasi ini berisikan informasi mengenai bagaimana tata cara menggunakan aplikasi dan melakukan kompresi file, untuk desain rancangannya dapat dilihat pada gambar III.9 berikut ini :



Gambar III.9. Tampilan Menu Panduan Aplikasi

3. Desain Menu Kompresi File

Bagian ini adalah merupakan bagian inti dari aplikasi yang berfungsi untuk melakukan proses enkripsi file WAV, tidak terlalu banyak komponen form yang ditampilkan pada bagian ini, hanya terdapat 2 form kompone pada bagian ini yaitu *file browser* yang difungsikan untuk memilih file yang akan di kompresi yang tombol proses yang berfungsi untuk melakukan proses kompresi, pada bagian bawah form terdapat tabel yang menampilkan *history* dari file yang telah selesai di kompresi. Untuk tampilan lebih jelasnya dapat dilihat pada gambar III.10 berikut ini :



Gambar III.10. Desain *Form* Kompresi

4. Desain Menu Tentang aplikasi

Desain tampilan berikut adalah merupakan desain tampilan yang memberikan informasi mengenai fungsi manfaat dan tujuan dari aplikasi , juga memberikan informasi mengenai pengembang dari aplikasi.



Gambar III.11. Desain Menu Tentang Aplikasi