

BAB III

ANALISIS DAN PERANCANGAN

III.1. Analisis Sistem

Pada tahapan analisis dan perancangan ini bertujuan menganalisa kebutuhan pengembangan aplikasi media pembelajaran kompresi dengan algoritma *LZW*. *Input* yang diproses dalam aplikasi yang dirancang berupa karakter teks dengan memberikan index untuk karakter yang sama agar memperkecil ukuran data, dalam perancangan hanya memberikan informasi teks, bukanlah *file*. Desain dan implementasi ini meliputi desain data, deskripsi sistem, desain proses dan implementasi desain dan semua yang diperlukan dalam aplikasi kompresi yang dirancang

III.1.1. Analisa Input

Dalam sistem media pembelajaran pengkompresian teks yang akan di implementasikan dalam aplikasi adalah menggunakan algoritma *LZW*. Dengan membaca tiap karakter yang dimasukan dari teks yang dimasukan. Tiap karakter lalu diproses hingga berurutan hingga karakter berikutnya. Menghasilkan index baru jika ditemukan kesamaan index, sehingga memperkecil ukuran dari sebuah data, dalam proses yang dikembangkan hanya menampilkan proses kerja algoritma *LZW* sebagai media pembelajaran agar pengguna dapat mengetahui bagaimana proses dari sebuah algoritma *LZW*.

III.1.2. Analisa Proses

Permasalahan yang dibahas adalah membuat suatu simulasi data kompresi dengan menggunakan algoritma *Lempel Ziv Welch* (LZW). Masalah kompresi data dengan metode *Lempel Ziv Welch* (LZW) muncul ketika kompresi sedang berlangsung. Pembahasan masalah lebih ditekankan pada proses indeks kerja algoritma *Lempel Ziv Welch* (LZW). Berikut ini contoh pengaplikasian kompresi dan dekompresi dengan menggunakan algoritma LZW. Contoh berikut ini hanya akan menjelaskan pengkompresian dan dekompresi terhadap beberapa karakter dengan mengkodekan menurut kode *ASCII*.

Kompresi *String* dalam Algoritma LZW

Sebagai contoh, *string* “MAMA_MASAK_NASI” akan dikompresi dengan LZW. Isi *dictionary* pada awal proses diset dengan tujuh karakter dasar yang ada: “M”, “A”, “_”, “S”, “K”, “N”, “I”. *bit dictionary* yang dipakai untuk contoh ini adalah 8 *bit*. tahapan kompresi ditunjukkan pada Tabel III.1 berikut.

Diketahui :

Input String: MAMA_MASAK_NASI

bit dictionary: 8 *bit* (default nilai *bit* 1 (satu) karakter)

Tabel III.1 Tahapan Proses Kompresi

No	Input	String+Char	IN Dictionary	Temporary	Dictionary		Output
					Code	Char	
1	M	M	-	M	-	-	None
2	A	MA	No	A	256	MA	M
3	M	AM	No	M	257	AM	A
4	A	MA	Yes(256)	MA	-	-	None
5	_	MA_	No	_	258	MA_	MA
6	M	_M	No	M	259	_M	_
7	A	MA	Yes(256)	MA	-	-	None
8	S	MAS	No	S	260	MAS	MA
9	A	SA	No	A	261	SA	S
10	K	AK	No	K	262	AK	A
11	_	K_	No	_	263	K_	K
12	N	_N	No	N	264	_N	_
13	A	NA	No	A	265	NA	N
14	S	AS	No	S	267	AS	A
15	I	SI	No	I	268	SI	S
16							I

Total awal *bit* disimpan tanpa kompresi = Total *input* * *bit dictionary*

(Jumlah Karakter) x (Nilai bit per 1 (satu) karakter)

$$15 \times 8 = 120 \text{ bit}$$

Besar *file* setelah dikompresi = Total *output* * *bit dictionary*

(Jumlah Karakter *Dictionary*) x (Nilai bit per 1 (satu) karakter)

$$\underline{12 \times 8 = 96 \text{ bit}}$$

Hasil kompresi dari *string* MAMA_MASAK_NASI yaitu:

Decimal : 77-65-256-32-256-83-65-75-32-78-65-83-73

Heksa : 4D-41-4D-41-20-4D-41-53-41-4B-20-4E-41-53-49

Biner : 01001101-01000001-01001101-01000001-00100000-01001101
01000001-01010011-01000001-01001011-00100000-01001110
01000001-01010011-01001001

Hasil Rasio Kompresi

$$\begin{aligned} \text{Rasio} &= \frac{\text{Ukuran File Terkompres}}{\text{Ukuran File Asli}} \times 100 \% \\ &= \frac{96}{120} \times 100 \% \\ &= 80 \% \end{aligned}$$

III.1.3. Analisa Output

Dari hasil analisa *input* dan analisa proses pada akhirnya akan menghasilkan *output*/hasil keluaran yang diterima pengguna, dari setiap karakter yang telah dikompresi dengan algoritma LZW diubah menjadi indeks baru dengan penomoran indeks yang dihasilkan dari proses aplikasi, pembelajaran memberikan pemahaman kepada pengguna dengan menampilkan animasi 2D, pengguna dapat melihat pemberian indeks dan penomoran baru untuk memperkecil data hingga akhir proses kompresi

III.2. Strategi Pemecahan Masalah

Untuk membangun aplikasi kompresi teks sesuai penggunaan algoritma *LZW* sebagai media pembelajaran. Beberapa strategi pemecahan masalah dalam perancangan adalah sebagai berikut :

- 1 *Input* dan *Output* merupakan sebuah teks masukan yang dapat diproses oleh aplikasi.
- 2 Proses kompresi dan dekompresi pada uji coba hanya dilakukan pada tiap karakter teks.
- 3 *Interface* media pembelajaran adalah animasi 2D proses indek karakter hingga menghasilkan kompresi *LZW*.
- 4 *Output* dari aplikasi adalah animasi sebagai media pembelajaran dan teks hasil kompresi.

III.3. Perancangan Sistem

Pada perancangan aplikasi menjelaskan mengenai rancangan dan hal-hal yang dikerjakan serta fitur-fitur yang akan dipakai pada aplikasi tersebut. Hal ini bertujuan untuk menjelaskan tahapan-tahapan yang dikerjakan, prosedur penggunaan, disain tampilan, serta spesifikasi sistem dari segi perangkat lunak maupun perangkat keras yang digunakan dalam proses perancangan.

III.3.1. Analisa Kebutuhan fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang berisi untuk melengkapi perancangan. Kebutuhan fungsional juga berisi informasi-informasi apa saja yang harus ada dan dihasilkan. Berikut kebutuhan fungsional yang

terdapat pada rancangan aplikasi yang dibangun :

1. Mengimplementasikan penggunaan bahasa pemrograman *Visual Studio 2010* dalam membuat aplikasi media pembelajaran algoritma *LZW*.
2. Aplikasi dapat menggambarkan penerapan algoritma *LZW* sebagai media pembelajaran.
3. *Input* dan *output* berupa *file* teks yang dapat diproses dengan algoritma *LZW*.

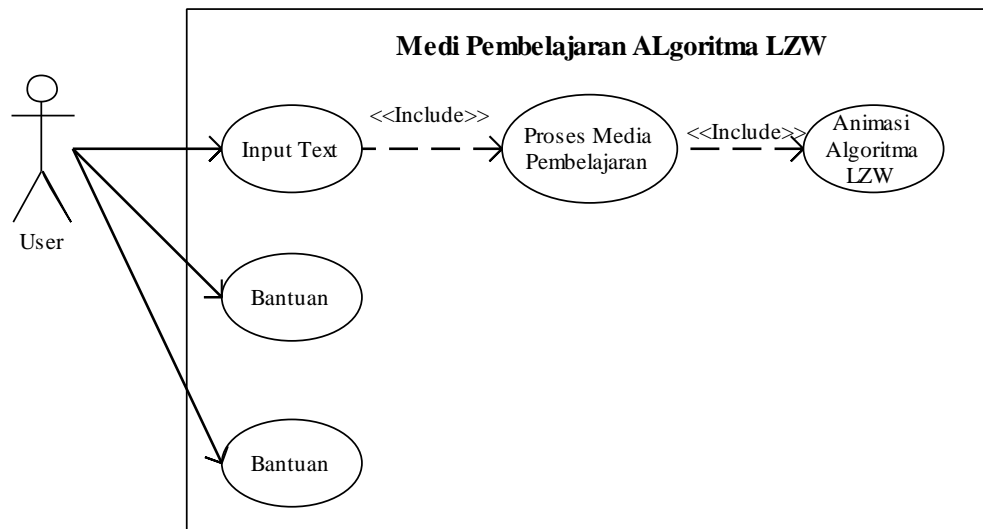
III.3.2. Analisa Kebutuhan Nonfungsional

Dalam perancangan aplikasi media pembelajaran algoritma *LZW*, beberapa perangkat yang penulis gunakan agar aplikasi berjalan baik, yaitu sebagai berikut :

1. Perangkat Lunak (*Software*)
 - a. *Operating System Windows Seven.*
 - b. *Visual Studio 2010* sebagai bahasa program yang digunakan.
2. Perangkat Keras (*Hardware*)
 - a. Komputer yang setara dengan *Intel pentium Dual Core.*
 - b. *Mouse, keyboard, dan Monitor*

III.3.3. Use Case Diagram

Use case diagram menggambarkan aktor yang menggunakan aplikasi dan perilaku pengguna, seperti pada gambar III.1 berikut.

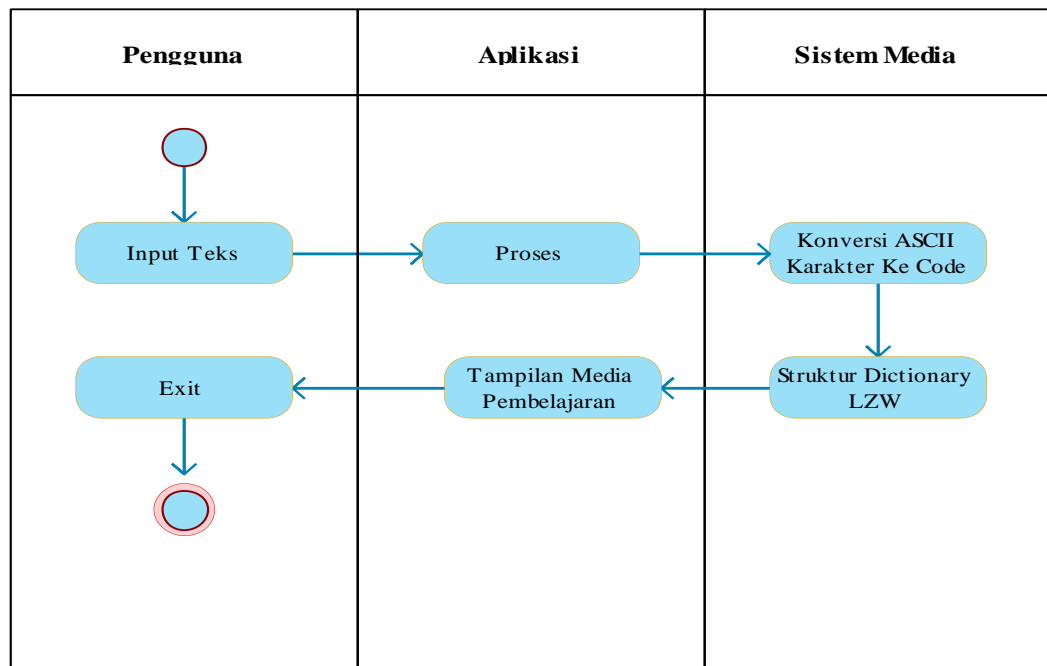


Gambar III.1. Use Case Diagram Pengguna Aplikasi

Kegiatan aktor atau pengguna pada aplikasi kompresi teks, pengguna dapat memilih melakukan proses kompresi terhadap teks yang telah dimasukan. Dari proses akan menampilkan media animasi 2D yang menjelaskan bagaimana proses kerja algoritma LZW.

III.3.4. Activity Diagram Proses Kompresi

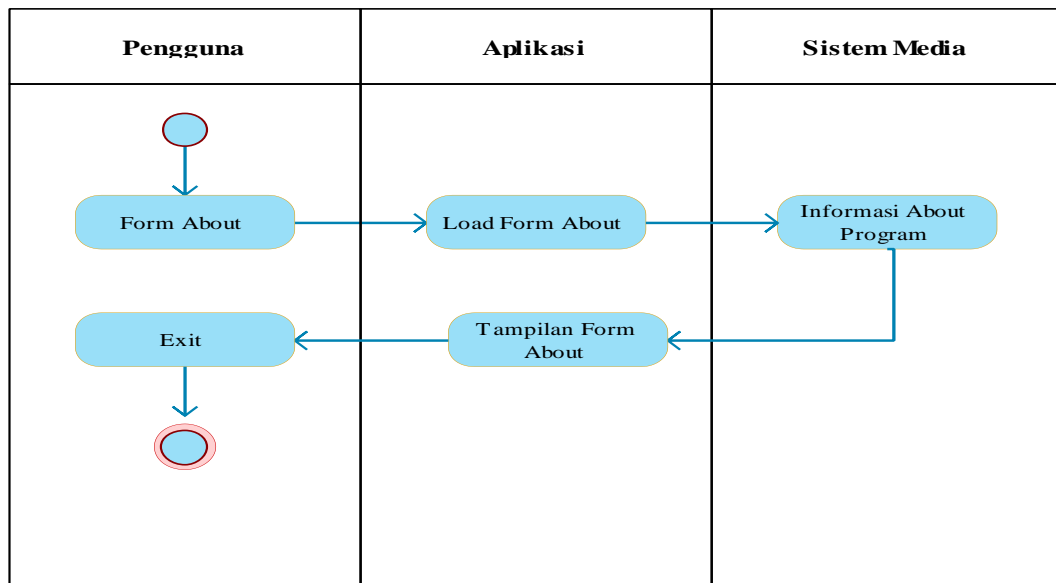
Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir aktivitas berawal. Adapun rancangan diagram aktivitas untuk proses kompresi dari aplikasi yang dirancang adalah sebagai berikut.



Gambar III.2. Activity Diagram Proses Media

III.3.5. Activity Diagram Proses Form About

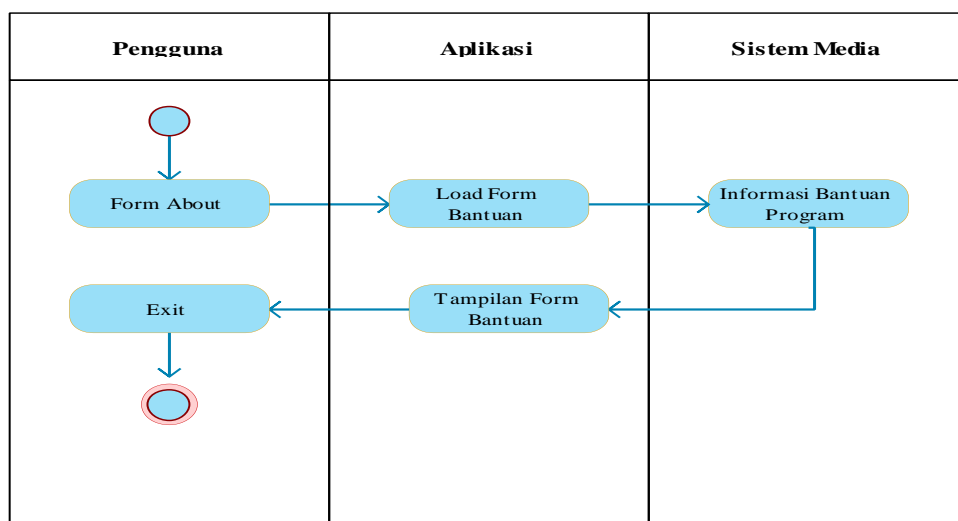
Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir aktivitas berawal. Adapun rancangan diagram aktivitas untuk menampilkan *form about* dari aplikasi yang dirancang adalah sebagai berikut.



Gambar III.3. Activity Diagram Form About

III.3.6. Activity Diagram Proses Form Bantuan

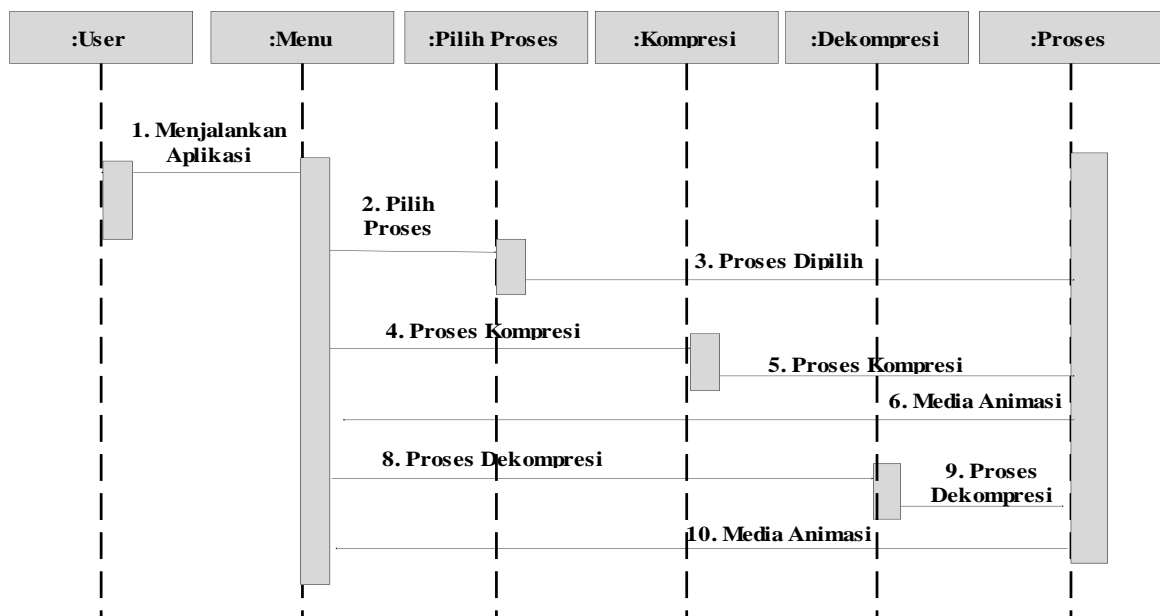
Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir aktivitas berawal. Adapun rancangan diagram aktivitas untuk menampilkan *form* bantuan dari aplikasi yang dirancang adalah sebagai berikut.



Gambar III.4. Activity Diagram Form Bantuan

III.3.7. Sequence Diagram Proses Kompresi

Sequence diagram menggambarkan kegiatan dari skenario penggunaan aplikasi, *sequence* diagram memilih proses kompresi pada media pembelajaran dapat dilihat pada gambar III.5 berikut.

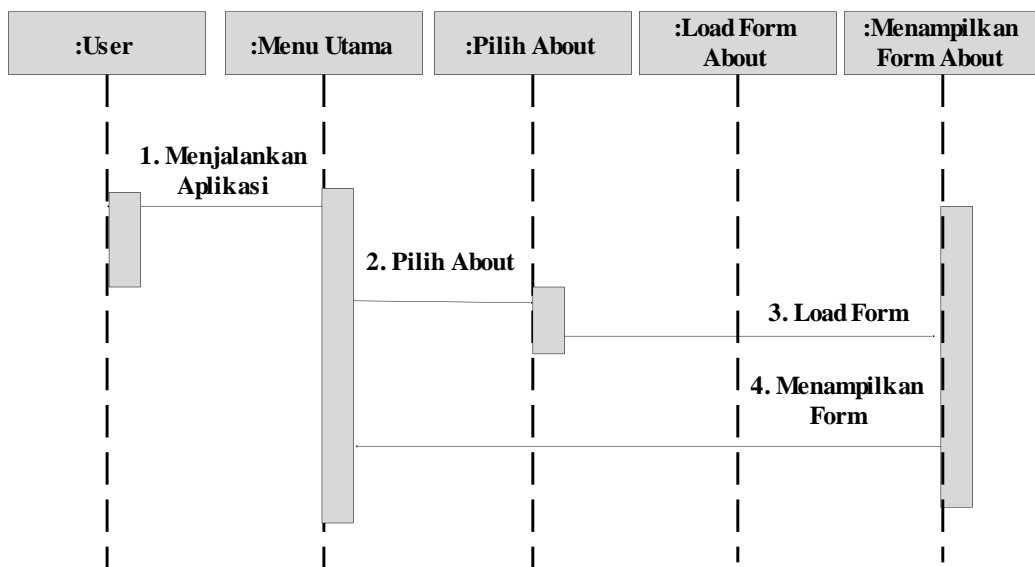


Gambar III.5. Sequence Diagram Proses Kompresi

Pengguna berinteraksi melalui pilihan proses yang ada pada menu utama, dapat dilihat pada *sequence diagram* diatas, pengguna memilih proses yang disediakan yaitu proses kompresi. Setelah pilihan proses ditentukan oleh pengguna kembali pada menu utama. Proses kompresi maupun dekompresi hanya dapat dilakukan setelah pengguna memasukan teks input yang ingin diproses. Selanjutnya proses algoritma *LZW* dapat dilakukan dengan menghasilkan tampilan media yang berfungsi sebagai pembelajaran.

III.3.8. Sequence Diagram Proses *Form About*

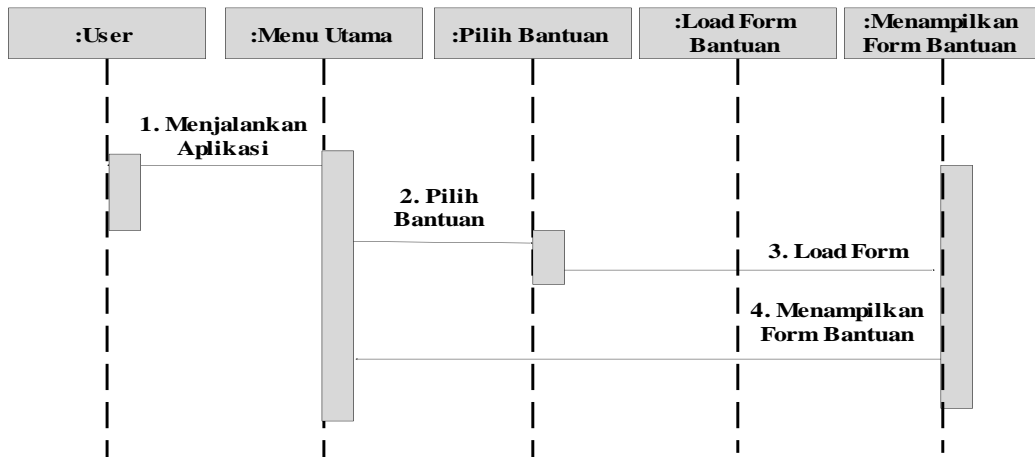
Sequence diagram menggambarkan kegiatan dari skenario penggunaan aplikasi, *sequence* diagram memilih proses *form about* pada media pembelajaran dapat dilihat pada gambar III.6 berikut.



Gambar III.6. *Sequence Diagram* Proses *Form About*

III.3.9. Sequence Diagram Proses *Form Bantuan*

Sequence diagram menggambarkan kegiatan dari skenario penggunaan aplikasi, *sequence* diagram memilih proses *form bantuan* pada media pembelajaran dapat dilihat pada gambar III.7 berikut.

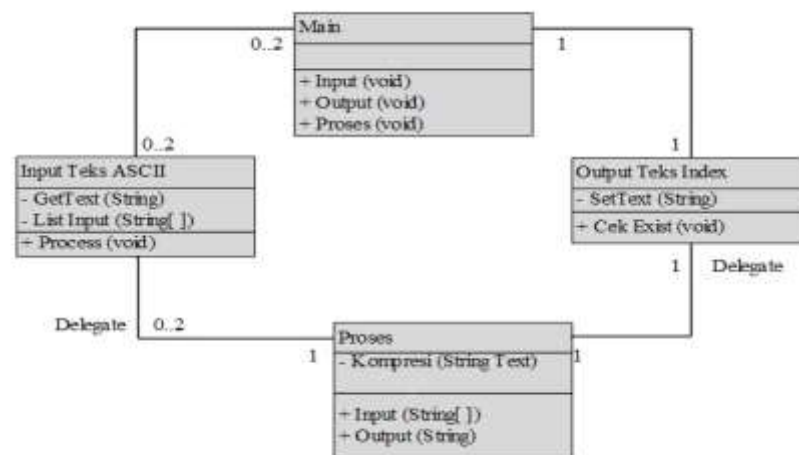


Gambar III.7. Sequence Diagram Proses Form Bantuan

Pengguna berinteraksi melalui pilihan proses yang ada pada menu utama, dapat dilihat pada *sequence diagram* diatas, pengguna memilih menu *form* bantuan yang disediakan, sehingga menampilkan *form* bantuan yang berisi informasi tentang bantuan penggunaan program yang dirancang.

III.3.10. Class Diagram

Class diagram pada perancangan aplikasi ini, dapat dilihat pada gambar III.8 berikut.



Gambar III.8. Class Diagram Sistem Perancangan Aplikasi

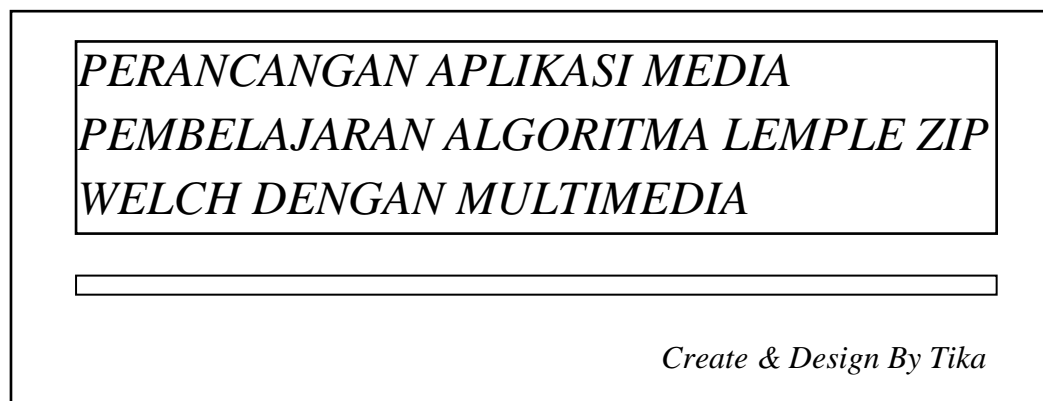
Class diagram adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* diagram juga menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan.

III.4. Perancangan Tampilan

Pada tahapan penulisan ini menjelaskan rancangan tampilan *form* yang akan dibangun pada aplikasi yang direncanakan. Adapun rancangan tampilan masing-masing halaman *form* tersebut dapat dijelaskan sebagai berikut.

III.4.1. Tampilan *Form Splash*

Rancangan *form* ini merupakan tampilan pembuka saat menjalankan aplikasi, yang dapat dilihat pada gambar III.9.



Gambar III.9. Tampilan *Form Splash*

form ini menampilkan pembuka aplikasi sebelum *form* utama, *form* ini akan membuka *form* utama, tampil pada layar dan akan ditutup setelahnya, *form* ini hanya tampil ketika program aplikasi baru dijalankan.

III.4.2. Tampilan *Form* Utama

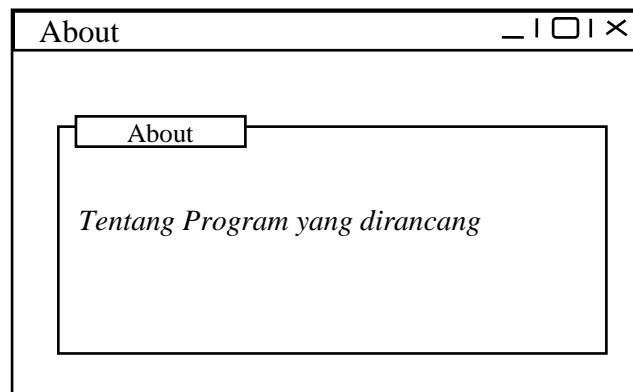
Tampilan *form* utama merupakan tampilan *form* yang fungsi sebagai media proses, didalamnya terdapat *field-field input* dan media tampilan algoritma LZW. Adapun rancangan tampilan *form* utama dapat dilihat pada gambar III.10.

The screenshot shows a software interface with a menu bar at the top containing 'Bantuan', 'Tentang Aplikasi', and 'Keluar'. Below the menu bar, there are two main sections. The left section is titled 'Masukan Teks Normal' and contains a 'Proses' dropdown menu set to 'Kompresi', followed by four input fields labeled 'Input Teks', 'Size Input', 'Output', and 'Size Output'. Below these fields are five buttons: 'Prev', 'Stop', 'Play', 'Pause', and 'Next'. The right section is titled 'Animasi' and contains the text 'Animasi Proses' and 'Kerja Algoritma LZW'.

Gambar III.10. Tampilan *Form* Utama

III.4.3. Tampilan *Form* Tentang Aplikasi

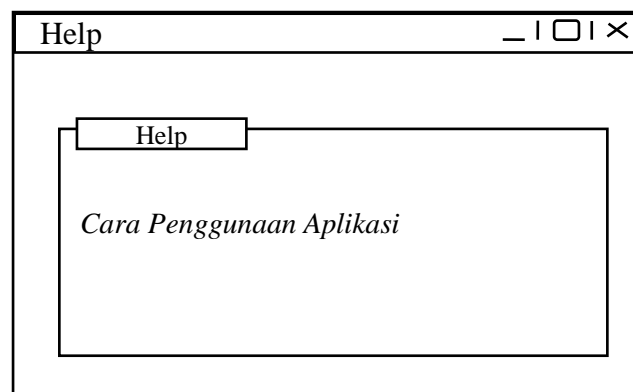
Tampilan *form* tentang aplikasi merupakan *form* yang memberikan sekilas mengenai tujuan perancangan aplikasi yang dibuat, yang dapat dilihat pada gambar III.11 berikut.



Gambar III.11. Tampilan *Form* Tentang Aplikasi

III.4.4. Tampilan *Form* Bantuan

Tampilan *form* bantuan adalah *form* yang berisi penjelasan mengenai cara penggunaan aplikasi, dapat dilihat pada gambar III.12 berikut.



Gambar III.12. Tampilan *Form* Bantuan

Informasi yang terdapat pada *form* bantuan adalah informasi cara menggunakan aplikasi yang telah dirancang dan dapat dijalankan, agar pengguna dapat dengan mudah memahami untuk menjalankannya.