

**APLIKASI MULTIMEDIA DENGAN MENGGUNAKAN
BAHASA PEMOGRAMAN VISUAL BASIC 6.0**

LAPORAN TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Persyaratan
Menyelesaikan Program Studi Strata Satu (S1)
Teknik Informatika

Disusun Oleh:

RIKA NOVITA

01503-007



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2008**

LEMBAR PENGESAHAN

Yang bertandatangan di bawah ini menyatakan bahwa Laporan Tugas Akhir dari mahasiswa berikut ini:

Nama : Rika Novita
NIM : 01503-007
Fakultas : Ilmu Komputer
Program Studi : Teknik Informatika
Judul : Pengembangan Aplikasi Audio Player Dengan Penambahan Fungsi Equalizer Menggunakan Visual Basic 6.0

Telah disidangkan, diperiksa dan disetujui sebagai Laporan Tugas akhir.

Jakarta, Desember 2008

Menyetujui,

Abdusy Syarif, ST., MT.
Pembimbing I Tugas Akhir

Mengesahkan,

Bambang Jokonowo, SSi. MT
Pembimbing II Tugas Akhir

Mengetahui,

Abdusy Syarif, ST., MT.
Ketua Program Studi
Teknik Informatika

Raka Yusuf, ST. MTI
Koordinator Tugas Akhir
Teknik Informatika

ABSTRAK

Pemanfaatan komputer dimasa sekarang ini telah berkembang disegala sisi kehidupan manusia, salah satu pemanfaatan komputer pada masa sekarang digunakan dalam bidang industri, pendidikan, informasi, pertahanan keamanan, hiburan dan lain-lain. Salah satu pemanfaatan komputer dibidang hiburan adalah pembuatan aplikasi-aplikasi pemutar audio player yang dimaksudkan untuk mengurangi tingkat kejenuhan dalam menjalankan aktivitas sehari-hari.

Aplikasi program yang banyak dikembangkan belakangan ini lebih memudahkan pemakai dalam mengambil keputusan yang diperlukan dengan cepat, dan biaya yang dibutuhkan dalam membangun sebuah aplikasi menjadi lebih murah dengan semakin banyaknya bahasa pemrograman yang bisa dikembangkan oleh para *programmer*.

Pembuatan aplikasi Audio Player dengan Visual Basic 6.0 ini bertujuan untuk mengembangkan aplikasi yang ada dengan menambahkan fitur *equalizer* agar dapat mendapatkan kualitas suara yang lebih baik.

Kata Kunci : Aplikasi, Audio Player, Equalizer

ABSTRACT

Now days, exploiting of computer have expanded all human life side. Computer at this time has been used in the industry field, education, information, defense of security, entertainment, etc. One of the exploiting of computer in entertainment area is making of audio player applications that meant to lessen saturation storey in running daily activity.

Application program which developed lately has given more facilitate to users in taking needed decision swiftly, and expense which required in developing an application become cheaper with more programming languages that could be developed by programmers.

The aim of making Audio Player Application with Visual Basic 6.0 is to develop an existing application by adding an equalizer feature so can get the quality of better voice.

Keyword: Application, Audio Player, Equalizer.

DAFTAR ISI

Lembar Pengesahan	i
Abstrak	ii
Abstract	iii
Kata Pengantar.....	iv
Lembar Pernyataan	vi
Daftar Isi.....	vii
Daftar Gambar	x
Daftar Tabel.....	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Pembahasan.....	2
1.3 Rumusan Masalah	3
1.4 Batasan Masalah.....	3
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Pengertian Multimedia.....	6
2.2 Pengertian Audio dan Suara.....	7
2.3 Pengertian <i>Equalizer</i>	10
2.4 Pengertian <i>Sekuensial Linear</i>	10
2.5 Defenisi Komputer	12

2.6	Pengertian Pemrograman	13
2.6.1	Program	13
2.6.2	Bahasa Pemrograman.....	13
2.7	Pengertian Bahasa Pemrograman Visual Basic	15
2.7.1	Visual Basic 6.0	16
2.7.2	Sejarah Visual Basic	16

3.2.2.4 Diagram <i>Sequence</i> untuk <i>Use Case</i> mengontrol file...	39
3.2.3 Pemodelan Diagram <i>Activity</i>	40
3.2.3.1 Diagram <i>Activity</i> untuk <i>Use Case</i> memutar file.....	41
3.2.3.2 Diagram <i>Activity</i> untuk <i>Use Case</i> mengatur <i>volume</i>	41
3.2.3.3 Diagram <i>Activity</i> untuk <i>Use Case</i> mengatur <i>equalizer</i> ...	42
3.2.3.4 Diagram <i>Activity</i> untuk <i>Use Case</i> mengontrol file.....	44
3.3 Perancangan Tampilan Pada Aplikasi Audio Player.....	45
3.3.1 Rancangan Menu Utama	45
BAB IV IMPLEMENTASI DAN PEGUJIAN	46
4.1 Implementasi	46
4.3.1 Lingkungan Implementasi.....	46
4.3.2 Pengkodean.....	47
4.3.3 Antarmuka	50
4.2 Pengujian.....	53
4.2.1 Skenario Pengujian	54
4.2.2 Hasil Pengujian	56
4.2.3 Analisis Hasil Pengujian	57
BAB V PENUTUP	59
5.1 Kesimpulan	59
5.2 Saran	59
Daftar Pustaka	61
Lampiran Listing Program	L1

DAFTAR TABEL

Tabel 2.1 Jenis Diagram Resmi UML	21
Tabel 2.2 Notasi Pemodelan Diagram <i>Use case</i>	23
Tabel 2.3 Simbol-simbol Pada Diagram Aktifitas.....	25
Tabel 2.4 Notasi Pemodelan Diagram <i>Sequential</i>	27
Tabel 3.1 Spesifikasi Naratif Untuk <i>Use case</i> Mengatur <i>Equalizer</i>	33
Tabel 3.2 Spesifikasi Naratif Untuk <i>Use case</i> Mengatur <i>Volume</i>	34
Tabel 3.3 Spesifikasi Naratif Untuk <i>Use case</i> Memutar File.....	34
Tabel 3.4 Spesifikasi Naratif Untuk <i>Use case</i> Mengontrol File.....	35
Tabel 4.1 Tabel Skenario Pengujian Fungsi.....	55
Tabel 4.2 Tabel Skenario Pengujian Kesalahan Interface	55
Tabel 4.3 Tabel Hasil Pengujian Fungsi	56
Tabel 4.4 Tabel Hasil Pengujian Kesalahan Interface	56

DAFTAR GAMBAR

Gambar 2.1 Model Sequensial Linier	12
Gambar 2.2 Bagian Dari Menu Project.....	18
Gambar 2.3 Bagian Dari Menu Components	19
Gambar 3.1 Diagram <i>Use case</i>	33
Gambar 3.2 Diagram <i>Sequence</i> Untuk <i>Use case</i> Memutar File.....	37
Gambar 3.3 Diagram <i>Sequence</i> Untuk <i>Use case</i> Mengatur <i>Volume</i>	38
Gambar 3.4 Diagram <i>Sequence</i> Untuk <i>Use case</i> Mengatur <i>Equalizer</i>	39
Gambar 3.5 Diagram <i>Sequence</i> Untuk <i>Use case</i> Mengontrol File.....	40
Gambar 3.6 Diagram <i>Aktifitas</i> Untuk <i>Use case</i> Memutar File.....	41
Gambar 3.7 Diagram <i>Aktifitas</i> Untuk <i>Use case</i> Mengatur <i>Volume</i>	42
Gambar 3.8 Diagram <i>Aktifitas</i> Untuk <i>Use case</i> Mengatur <i>Equalizer</i>	43
Gambar 3.9 Diagram <i>Aktifitas</i> Untuk <i>Use case</i> Mengontrol File	44
Gambar 3.10 Rancangan Menu Utama.....	45
Gambar 4.1 Tampilan Menu Utama Memutar File	50
Gambar 4.2 Tampilan Menu Utama Open File	51
Gambar 4.3 Tampilan Menu Utama Informasi Application	52
Gambar 4.4 Tampilan Menu Utama IT Programmer Application	52
Gambar 4.5 Tampilan Menu Exit	53

BAB I

PENDAHULUAN

1.1 Latar Belakang

Peran serta komputer dalam dunia hiburan khususnya dalam bidang musik sudah tidak bisa dipisahkan, baik itu dalam dunia rekaman maupun sebagai pemutar audio saja. Berbagai bentuk pemutar audio belakangan ini sudah menjamur, namun apabila kita mampu mencari solusi untuk mengatasi hal itu sebenarnya permasalahan diatas bukan suatu masalah besar. Karena pada masa ini peralatan pendukung untuk mewujudkan peralatan tersebut diatas sudah banyak tersedia, diantaranya pada komputer hanya dengan beberapa bahasa pemrograman saja kita sudah bisa mewujudkan suatu aplikasi pemutar audio yang terinstalasi pada *Personal Computer* (PC).

Perkembangan komputer juga tidak terlepas dari perkembangan perangkat lunak (*Software*) yang mendukungnya, dimana dengan *software* tersebut para programmer bisa menciptakan aplikasi-aplikasi yang bisa digunakan untuk membantu menyelesaikan

pekerjaan manusia dan lain-lain. Selain itu bahasa pemrograman juga bisa digunakan untuk membuat suatu aplikasi penyedia hiburan musik atau *audio player* berbasis komputer yang bisa menjalankan file-file seperti : MP3, MIDI, WAV, Mpeg dan Wma.

Atas dasar hal tersebut maka penulis berniat untuk membuat sebuah aplikasi yang bertemakan penyediaan hiburan atau audio player dengan menambahkan fitur *equalizer*, adapun judul yang penulis ambil adalah Pengembangan Aplikasi Audio Player Dengan Penambahan Fungsi Equalizer Menggunakan Visual Basic 6.0. tema ini diambil karena penulis memperhatikan bahwa begitu banyak orang-orang disekitar kita yang selalu disibukkan dengan pekerjaan yang melelahkan dan banyak menyita waktu, sehingga membuat kita jenuh dan kurang bersemangat dalam melaksanakan pekerjaan dan menjalankan aktifitas lainnya. Oleh karena itu untuk mengurangi tingkat ketegangan dalam menjalankan aktifitas diperlukan sarana yang bisa memberikan hiburan, seperti aplikasi penyedia hiburan atau *audio player* yang dimaksudkan untuk mengurangi kepenatan atau stress dalam melaksanakan pekerjaan dan menjalankan aktifitas sehari-hari.

1.2 Tujuan Pembahasan

Adapun tujuan pembahasan masalah yang penulis utarakan adalah :

1. Dapat mewujudkan suatu aplikasi pemutar audio yang terinstalasi pada *personal komputer* (PC).
2. Bisa menjalankan file-file audio seperti MP3, MIDI, WAV Mpeg dan Wma.
3. Dapat menambahkan fitur *equalizer* pada aplikasi.

1.3 Rumusan Masalah

Berdasarkan uraian latar belakang masalah dapat dirumuskan bagaimana cara fitur *equalizer* dapat berkesesuaian dengan aplikasi yang telah dibuat.

1.4 Batasan Masalah

Dalam penyusunan laporan tugas akhir ini, penulis memberikan batasan masalah sebagai berikut:

1. Laporan tugas akhir ini hanya membahas mengenai proses pembuatan aplikasi multimedia khususnya audio.
2. Aplikasi yang dibuat dengan menggunakan bahasa pemrograman Visual Basic 6.0 hanya terbatas pada menjalankan file audio (MP3).
3. Bagaimana cara fitur *equalizer* dapat berkesesuaian dengan aplikasi yang telah dibuat.

1.5 Metode Penelitian

Dalam melakukan perancangan aplikasi pengembangan ini, metodologi perangkat lunak yang digunakan oleh penulis adalah model sekuensial linier. Model sekuensial linier meliputi aktivitas-aktivitas sebagai berikut:

1. Rekayasa Sistem

Rekayasa sistem meliputi pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis dan desain.

2. Analisis Kebutuhan Perangkat Lunak

Tahap ini merupakan tahap pengumpulan kebutuhan yang diintensifkan dan difokuskan pada perangkat lunak.

3. Desain

Proses desain adalah menerjemahkan syarat atau kebutuhan ke dalam sebuah presentasi perangkat lunak sehingga dapat diperkirakan kualitasnya sebelum dimulai pemunculan kode.

4. Pengkodean

Pengkodean merupakan tahapan dimana desain yang diterjemahkan ke dalam bentuk mesin yang dapat dibaca oleh komputer.

5. Pengujian

Pengujian merupakan tahapan dimana perangkat lunak yang dibuat diujikan untuk menemukan kesalahan-kesalahan dan memastikan input memberikan hasil yang aktual sesuai dengan kebutuhan.

1.6 Sistematika Penulisan

Untuk memahami lebih jelas tentang penulisan Tugas Akhir ini, maka penulis mengelompokkan materi penulisan menjadi lima bab yang masing-masing saling berkaitan antara bab satu dengan bab lainnya, sehingga menjadi satu kesatuan yang utuh, dengan sistematika penulisan sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini berisikan tentang latar belakang, tujuan pembahasan, rumusan masalah, batasan masalah, metode penelitian dan sistematika penulisan.

BAB II : LANDASAN TEORI

Pada bab ini memuat teori-teori yang dipergunakan dalam penulian Tugas Akhir ini, seperti pengertian audio dan suara, equalizer, multimedia, aplikasi, pemograman Visual Basic 6.0 dan beberapa pengertian lainnya yang berkaitan dengan teori-teori yang digunakan pada pembuatan aplikasi ini.

BAB III : ANALISIS DAN PERANCANGAN

Pada bab ini akan dijelaskan mengenai hal-hal yang berhubungan dengan analisa kebutuhan sistem, rancangan proses implementasi, dan perancangan antarmuka

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Pada bab ini berisi tentang implementasi dan pegujian dari aplikasi yang telah dibuat.

BAB V : PENUTUP

Pada bab ini berisi kesimpulan dan saran oleh penulis dari seluruh penjabaran yang diuraikan pada bab-bab sebelumnya.

BAB II

LANDASAN TEORI

2.1 Pengertian Multimedia

“*The American Heritage Dictionary*” mendefinisikan bahwa Multimedia adalah kombinasi dari penggunaan beberapa media seperti film, slide, musik, penerangan dengan teks, image, khususnya untuk tujuan pendidikan dan hiburan. Unsur-unsur seperti teks audio (narasi, dialog, sound effects), musik, film, video, fotografi, animasi dan grafik merupakan media pendukung yang tergabung dan terintegrasi menjadi satu kesatuan karya multimedia. Dunia komputer umumnya perkembangan teknologi multimedia tidak hanya tergantung pada perkembangan *software* semata, tetapi juga terkait dengan perangkat elektronik pendukung lainnya.

2.2 Pengertian Audio dan Suara

Audio adalah segala bunyi yang terdengar oleh telinga manusia. Audio juga merupakan gelombang getaran suara yang sampai dan dapat ditangkap oleh gendang telinga. Banyaknya getaran itu diukur dengan *hertz* atau disingkat *Hz*. Daya tangkap pendengaran manusia sendiri adalah antara 20 *Hz* sampai 20.000 *Hz* (20 *Khz*) dimana dalam teknologi komputer, audio sendiri lebih dikenal dengan sebutan file *audio* yang artinya adalah file yang hanya berisi rekaman suara.

Dimana file *audio* dalam komputer terbagi kedalam beberapa format diantaranya (MB. Rahimsyah,323-452) :

1. File WAV

File *.WAV (WAveForm)* adalah file untuk menyimpan suara yang direkam dengan teknologi dari microsoft atau akhiran / ekstensi nama file suara yang paling sering dipakai untuk menyimpan data suara seperti musik, suara manusia atau suara lain. Dalam lingkungan windows, format file ini telah diakui sebagai format audio standar. Keuntungan dari file *.WAV* adalah kita bisa menyimpan suara apa saja dari berbagai sumber, seperti mikrofon, pemutar CD atau *tape recorder*. Sedangkan kelemahan dari file ini yang paling mencolok adalah dari segi ukuran dimana ukurannya sangat besar. Misalkan sebuah lagu yang bila diputar hanya membutuhkan waktu beberapa menit saja, bisa memakan ruangan harddisk puluhan *megabyte*. Dalam memainkan file Wave dengan Visual Basic, bisa menggunakan control program yang telah disediakan oleh Visual Basic atau *software* multimedia lainnya dan dapat juga dimainkan dengan memanfaatkan fasilitas *library "Winmm.dll"*

2. File MIDI

Sama seperti *file* WAV, *file* MIDI (*Musical Instruments Digital Interface*) yang berakhiran .MID juga berhubungan dengan suara atau MIDI adalah sesuatu yang dibuat untuk menyimpan atau memainkan musik dengan synthesizer digital, yang membedakan adalah teknik penyimpanannya. Keuntungan file .MID dibandingkan dengan file .WAV adalah ukurannya yang sangat kecil, sedangkan kerugiannya tidak semua suara bisa disimpan dalam *file* .MID hanya dari alat musik tertentu saja yang bisa disimpan. Dalam memainkan *file* .MIDI dengan Visual Basic, bisa menggunakan kontrol program yang telah disediakan oleh Visual Basic atau software lainnya dan dapat juga dimainkan dengan memanfaatkan fasilitas *library* "Winmm.dll".

3. File MP3

File MP3 adalah *file* dengan format perekaman suara digital yang dikompresi. Saat ini perusahaan Thomson Multimedia SA dan Fraunhofer Institute, pencipta format MP3 berhasil merilis *coder* and *decoder* (*codec*) untuk *format* MP3pro. Kini MP3 dapat dilipat gandakan tingkat kompresinya dari semula, tetapi dengan kualitas suara yang sama. Untuk biasa mendengarkan file MP3 dibutuhkan media player, salah satunya dimiliki oleh program Visual Basic yaitu dengan Multimedia MCI (*Musical Instruments Digital Interface*).

Suara adalah fenomena fisik yang dihasilkan oleh getaran benda atau getaran suatu benda yang berupa sinyal analog dengan amplitudo yang merubah secara kontinyu terhadap waktu. Suara berhubungan erat dengan rasa mendengar, suara atau bunyi biasanya merambat melalui udara dan tidak bisa merambat melalui ruang hampa udara. Suara dihasilkan oleh getaran suatu benda, selama benda bergetar perbedaan tekanan

terjadi diudara sekitarnya. Pada osilasi yang terjadi dinamakan sebagai gelombang. Gelombang mempunyai pola sama yang berulang pada interval tertentu, yang disebut sebagai periode.

Suara berkaitan erat dengan :

1. Frekuensi

- a. banyaknya periode dalam 1 detik
- b. satuan *Hertz (Hz)* atau *cycles per second (cps)*
- c. panjang gelombang suara (wavelength) dirumuskan c/f dimana c adalah kecepatan rambat bunyi dan f adalah frekuensi

berdasarkan frekuensi suara dibagi menjadi empat yaitu infra sound (0Hz-20Hz), pendengaran manusia (20Hz-20KHz), ultrasound (20KHz-1GHz) dan hypersound (1GHz-10THz). Manusia membuat suara dengan frekuensi 50Hz-10KHz, sedangkan sinyal suara musik memiliki frekuensi 20Hz-20KHz. Sistem multimedia menggunakan suara yang berada dalam range pendengaran manusia. Suara yang berada pada range pendengaran manusia sebagai audio dan gelombangnya sebagai *accoustic signals*. Suara diluar range pendengaran manusia dapat dikatakan suara *noise* (getaran yang tidak teratur dan tidak berurutan dalam berbagai frekuensi dan tidak dapat didengar manusia)

2. Amplitudo

- a. Keras lemahnya bunyi atau tinggi rendahnya gelombang.
- b. Satuan amplitodu adalah decibel (db).
- c. Bunyi dapat merusak telinga jika tingkat volumenya lebih besar dari 85 dB dan pada ukuran 130 dB akan mampu membuat hancur gendang telinga.

3. Velocity

- a. Kecepatan perambatan gelombang bunyi sampai ketelinga pendengar.
- b. Satuan yang digunakan adalah m/s
- c. Pada udara kering dengan suhu 20°C (68°F)m kecepatan rambat suara sekitar 343 m/s

2.3 Pengertian *Equalizer*

Equalizer merupakan alat untuk memperbaiki kualitas frekuensi yang diterima suatu rangkaian transmisi. Alat ini biasanya dirangkaikan bersama alat transmisi lain. *Equalizer* sebagai salah satu perangkat yang sangat menunjang untuk menghasilkan suara yang lebih berkualitas. Bila suara *equalizer* sudah dirasa cukup maka kita tidak perlu mengubah frekuensi dari *equalizer* itu sendiri, tetapi apabila terlalu tinggi kita tinggal mengurangi frekuensi *equalizer* pada wilayah frekuensi dari *bass* sampai *treble*. Sebenarnya dalam mengubah frekuensi *equalizer* sangat berpengaruh pada selera masing-masing pendengar khususnya audio player (MP3)

2.4 Model Sekuensial Linier

Model sekuensial linier untuk rekayasa perangkat lunak, atau sering disebut juga dengan "siklus kehidupan klasik" atau "model air terjun". Model sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, pengujian dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional,

model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut menurut (Roger S. Pressman, 2002:37—38):

1. Rekayasa dan pemodelan sistem/informasi

Pada tahapan ini dilakukan proses pengumpulan kebutuhan pengguna pada tingkatan sistem dengan sejumlah kecil analisa serta desain tingkat puncak.

2. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perekayasa perangkat lunak harus memahami domain informasi, tingkah laku, unjuk kerja, dan antarmuka (*interface*) yang diperlukan.

3. Desain

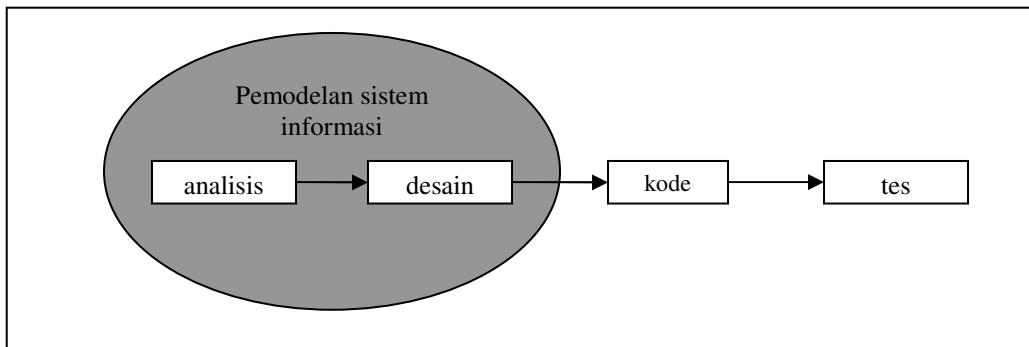
Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut program yang berbeda, struktur data, arsitek perangkat lunak, representasi antarmuka, dan detail prosedural. Proses desain menerjemahkan kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

4. Generasi kode

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan dengan mekanis. Artinya dilakukan pengkodean (*coding*) berdasarkan desain yang telah dibuat.

5. Pengujian

Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa masukan yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan. Untuk lebih jelas dapat kita lihat pada gambar 2.1



Gambar 2.1. Model Sekuensial Linier (Pressman, 2002:37)

2.5 Defenisi Komputer

Didalam pengolahan sistem kendali (*control system*) terdapat suatu alat yang disebut komputer. Istilah komputer (*computer*) diambil dari bahasa latin yaitu compare yang berarti menghitung (*to compute* atau *reckon*).

Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan supaya secara otomatis menerima dan menyimpan data input, memprosesnya dan menghasilkan output dibawah suatu pengawasan atau

langkah-langkah instruksi –instruksi program yang tersimpan dimemori (*storage program*).

2.6 Pengertian Pemrograman

Adapun pengertian dari pemrograman adalah proses pengimplementasian urutan langkah untuk menyelesaikan masalah dalam bentuk program yang memiliki rancang bangun terstruktur dan tidak berbelit sehingga mudah ditelusuri, dipahami dan dikembangkan oleh siapa saja dengan menggunakan suatu bahasa. Pemrograman bisa dibagi menjadi dua yaitu pemrograman terstruktur dan pemrograman linear (Budi Sutedjo, Michael AN, 2000:21).

2.6.1 Program

Program adalah kata, ekspresi, pernyataan atau kombinasinya yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman sehingga dapat dieksekusikan oleh komputer.

2.6.2 Bahasa Pemrograman

Pemrograman merupakan suatu proses guna mengimplementasikan algoritma dengan menggunakan suatu bahasa pemrograman. Satu hal yang cukup penting sebelum seorang pemrogram mulai menyusun program adalah memilih bahasa pemrograman yang akan digunakan.

Bahasa Pemrograman merupakan prosedur atau tata cara penulisan program. Pada bahasa Pemrograman terdapat dua faktor penting, yaitu *syntax* dan *semantik*.

Syntax adalah aturan-aturan gramatikal yang mengatur tata cara penulisan kata, ekspresi dan pernyataan, sedangkan *semantik* adalah aturan-aturan untuk menyatakan suatu arti.

Fungsi bahasa pemrograman adalah sebagai media untuk menyusun dan memahami serta sebagai alat komunikasi antara program dengan komputer, meskipun dapat juga digunakan sebagai alat komunikasi antara orang satu dengan yang lain.

Secara umum bahasa pemrograman dapat dibagi dalam empat kelompok yaitu :

a. Bahasa Tingkat Rendah (*Low Level Language*)

Bahasa tingkat rendah merupakan bahasa pemrograman yang berorientasi pada mesin. Pemrograman yang menggunakan bahasa ini harus dapat berfikir berdasarkan logika mesin pada komputer, sehingga bahasa ini dinilai kurang fleksibel dan sulit untuk dipahami oleh pemula. Contohnya adalah bahasa *assembly*.

b. Bahasa Tingkat Menengah (*Middle Level Language*)

Bahasa tingkat menengah merupakan bahasa pemrograman yang menggunakan aturan-aturan gramatikal dalam penulisan ekspresi atau pernyataan dengan standar bahasa yang mudah dipahami oleh manusia serta memiliki instruksi-instruksi tertentu yang dapat langsung diakses oleh komputer. Contohnya adalah bahasa C.

c. Bahasa Tingkat Tinggi (*High Level Language*)

Bahasa tingkat tinggi merupakan bahasa pemrograman yang menggunakan aturan-aturan gramatikal dalam penulisan ekspresi atau pernyataan dengan standar bahasa

yang mudah dipahami oleh manusia. Contohnya adalah bahasa *pascal*, *Fortran*, *Cobol*, *Power Basic* dan lain-lain.

d. Bahasa Berorientasi Objek (*Object Oriented Language*)

Bahasa Berorientasi Objek merupakan bahasa pemrograman yang mengandung “kapsul-kapsul” yang berisi fungsi-fungsi untuk menyelesaikan masalah. Dengan bahasa ini pemrogram tidak lagi harus menuliskan secara detail semua pernyataan dan ekspresi seperti bahasa tingkat tinggi, melainkan cukup dengan memasukkan kriteria-kriteria yang dihendaki saja. Bahasa ini kemudian akan menggunakan “kapsul-kapsul” tersebut untuk memecahkan permasalahan itu. Contohnya adalah bahasa C++, Visual Foxpro, Visual Basic, Borland Delphi, Java, VB Net dan lain-lain.

Agar komputer dapat memahami program yang disusun dengan suatu bahasa pemrograman, maka dibutuhkan suatu penerjemah yaitu *interpreter* atau *compiler*.

2.7 Pengertian Bahasa Pemrograman Visual Basic

Microsoft Visual Basic 6.0 pada dasarnya adalah sebuah bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah atau instruksi-instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Basic kini seakan-akan menjadi kiblat bagi para software developer dan menjadi salah satu bahasa pemrograman yang wajib dipelajari oleh berbagai kalangan, jika mereka ingin sukses di dunia komputer. (Wiryanto Dewobroto, 2003:3)

Visual Basic (yang sering jug disebut VB) selain disebut sebagai sebuah bahasa pemrograman, juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis windows. Beberapa kemampuan atau manfaat dari Visual Basic diantaranya seperti :

1. Untuk membuat program aplikasi berbasis windows.
2. Untuk membuat objek-objek pembantu program seperti misalnya *control activeX*, *file help*, aplikasi internet dan sebagainya.
3. Menguji program (*debugging*) dan menghasilkan program akhir berakhiran *.exe* yang bersifat *executable* atau dapat langsung dijalankan.

2.7.1 Visual Basic 6.0

Visual Basic (atau sering disingkat VB) adalah perangkat lunak untuk menyusun program aplikasi yang bekerja dalam lingkungan sistem operasi Windows. Dengan Visual Basic kita bisa memanfaatkan kemampuan Windows secara optimal. Dengan kecanggihan yang ditawarkan oleh Visual Basic kita akan merasakan begitu mudahnya menyusun program aplikasi dengan tampilan grafis yang mcnawan dalam waktu yang relatif singkat.

2.7.2 Sejarah Visual Basic

Visual Basic dikembangkan dari bahasa *QuickBasic* yang dijalankan dalam sistem operasi DOS. Versi awal diciptakan oleh Alam Cooper kemudian menjualnya ke Microsoft yang mengambil alih pengembangan produk dengan memberi nama sandi "*Thunder*". Akhirnya Visual Basic menjadi bahasa pemrograman utama di lingkungan

Windows. Sebelum mencapai kepopuleran seperti sekarang, Visual Basic telah dirilis beberapa kali seperti berikut:

- a. Versi 1.0 dirilis bulan Mei 1991.
- b. Januari 1992 dilengkapi *control custom (Professional Toolkit)*.
- c. Versi 2.0 yang lebih “canggih” dirilis pada bulan Oktober 1992.
- d. Versi 3.0 dirilis tahun 1993 dengan “*Developmen tool*” andal.
- e. Versi 4.0 dirilis tahun 1995 dilengkapi *Object Oriented Programming*, mulai dipakai untuk mengembangkan aplikasi berbasis *Windows*.
- f. Versi 5.0 dirilis tahun 1997 dengan *Integrated Development Environment (IDE)*.
- g. Versi 6.0 dirilis tahun 1998 bersama dengan Microsoft Visual Studio.
- h. Versi .Net dirilis tahun 2001, yaitu Visual Basic.Net bersama dengan Microsoft Visual Basic.Net.

2.8 Dasar-dasar Pemrograman Multimedia

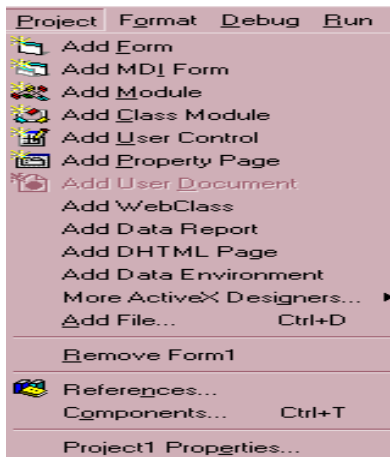
Visual Basic merupakan salah satu bahasa pemrograman yang mendukung konsep multimedia. Sejak diluncurkan pertama kali, Visual Basic yang bekerja di sistem operasi Windows terus menambah asilitas untuk pemrograman multimedia. Salah satu fasilitas tersebut adalah sebuah kontrol Active X yang disertakan dalam Visual Basic Edition dengan nama *Control Multimedia MCI (Media Control Interface)*, misalnya *CD player*, *VCR* dan *video Player*.

Sebenarnya dalam Visual Basic masih terdapat tiga Kontrol lain seperti *MCIWndx Coontrol*, *Microsoft Active Movie Control*, *Windows Media Player*. Tetapi dari

kesemuanya, control Multimedia MCI adalah yang terlengkap serta merupakan kontrol yang langsung berhubungan dengan Visual Basic yang melibatkan property dan event.

Sebelum mendesain program multimedia dengan Visual Basic terlebih dahulu membuka program visual basic dan aktifkan control multimedia yang ingin digunakan. Langkah-langkah untuk membuka program visual basic dan mengaktifkan kontrol-kontrol multimedia adalah sebagai berikut :

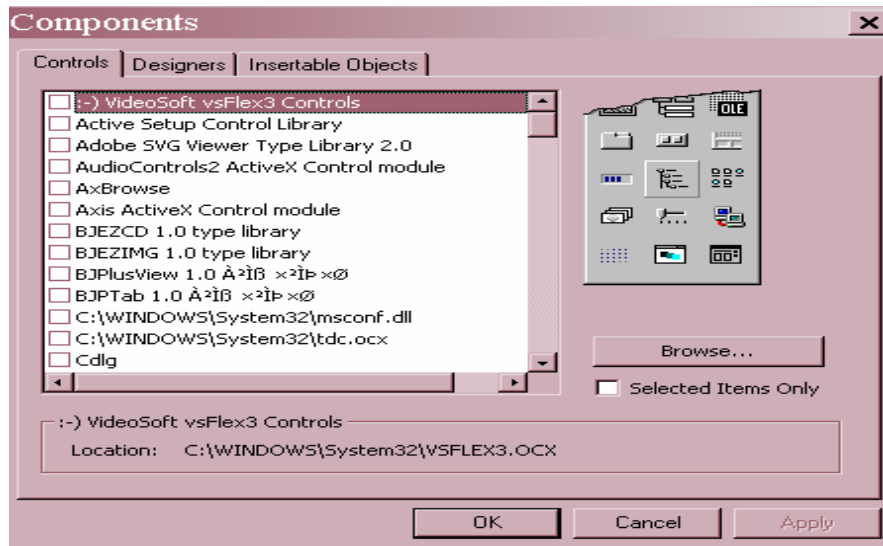
1. Klik microsoft visual studio yang ada pada tombol start, kemudian pilih microsoft visual basic.
2. Selanjutnya akan muncul *New Project*, pilih *Standart EXE* kemudian klik tombol open.
3. Klik menu Project dan klik menu popup Components atau mempersingkat kerja, tekan kombinasi tombol Ctrl+T.



Gambar 2.2 Bagian Dari Menu Project

4. Selanjutnya Visual Basic menampilkan kotak dialog Components.

5. Klik tab Controls dan lakukan *Scrolup* (gulung keatas) untuk menampilkan kontrol-kontrol multimedia.
6. Untuk memilih kontrol-kontrol multimedia, beri tanda ceklis (✓) pada kolom yang tersedia *components*



Gambar 2.3 Bagian Dari Menu Components

2.9 Microsoft Windows XP

Microsoft windows XP merupakan sistem operasi berbasis garis (gambar) dengan berbagai fasilitas, khususnya dalam berintegrasi dengan internet serta dengan kemudahan pengoperasiannya. *Microsoft Windows XP* ini merupakan salah satu produk unggulan dari *Microsoft Coep* yang secara resmi dikeluarkan pada tanggal 25 Oktober 2001. *Microsoft Windows XP* yang selanjutnya disingkat dengan *Windows XP* ini merupakan kelanjutan dari *Windows* versi sebelumnya dengan berbagai fasilitas yang ada didalamnya, baik merupakan fasilitas penyempurnaan maupun fasilitas terbaru yang tidak ada pada

versi-versi sebelumnya. Hal inilah yang menyebabkan semakin banyak pengguna sistem operasi *Windows XP* ini dalam masyarakat.

Dalam *Windows XP* terdapat tiga edisi, yaitu : *XP Home Edition*, *Professional Edition*, *64-Bit Edition*. Sedangkan edisi yang paling banyak dipergunakan oleh *user* adalah *Professional Edition*.

2.10 UML (Unified Modelling Language)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software (<http://www.omg.org>).

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Romi Satria Wahono dan Sri Dharwiyanti, 2003).

UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan *syntax* UML mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan.

UML sebagai sebuah bahasa yang memberikan vocabulary dan tatanan penulisan kata-kata dalam 'MS Word' untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang mempunyai *vocabulary* dan konsep tatanan/aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnya UML adalah sebuah bahasa standard untuk pengembangan sebuah software yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan software.

Menurut Munawar (2005:17) UML adalah alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan klien, programmer, dan tiap orang yang terlibat dalam proses pengembangan tersebut.

Di proyek pengembangan sistem apapun, fokus utama dalam analisis dan perancangan adalah model dan diagram. Model menggambarkan pandangan yang lengkap tentang suatu sistem pada suatu tahapan tertentu dan dari perspektif tertentu. Sedangkan diagram menggambarkan atau mendokumentasikan beberapa aspek dari sebuah sistem. Sebuah model mungkin mengandung satu atau lebih diagram. Untuk model sederhana, satu diagram mungkin akan mencukupi. Akan tetapi biasanya sebuah model terdiri dari banyak diagram. UML terdiri atas 13 jenis diagram resmi seperti tertulis dalam Tabel 2.1

Tabel 2.1 Jenis diagram resmi UML

No.	Diagram	Kegunaan
1.	<i>Activity</i>	<i>Behavior prosedural</i> dan <i>parallel</i>
2.	<i>Class</i>	<i>Class, fitur</i> , dan hubungan-hubungan
3.	<i>Communication</i>	Interaksi antar objek; penekanan pada jalur
4.	<i>Component</i>	Struktur dan koneksi komponen
5.	<i>Composite structure</i>	Dekomposisi <i>runtime</i> sebuah <i>class</i>
6.	<i>Deployment</i>	Pemindahan artifak ke node
7.	<i>Interaction overview</i>	Campuran <i>sequence</i> dan <i>activity</i> diagram
8.	<i>Object</i>	Contoh konfigurasi dari contoh-contoh
9.	<i>Package</i>	Struktur hirarki <i>compile-time</i>
10.	<i>Sequence</i>	Interaksi antar objek; penekanan pada <i>sequence</i>
11.	<i>State machine</i>	Bagaimana <i>event</i> mengubah objek selama aktif
12.	<i>Timing</i>	Interaksi antar objek; penekanan pada <i>timing</i>
13.	<i>Use case</i>	Bagaimana pengguna berinteraksi dengan sebuah sistem

Diagram UML yang akan dibahas pada bab ini adalah diagram use case (*use case* diagram), diagram aktifitas (*activity* diagram) dan diagram sekuensial (*sequence* diagram).

2.10.1 Diagram Use Case

Diagram *use case* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. Diagram *use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan (Fowler, 2005:141). Diagram use case menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem, dan bukan bagaimana (Dharwiyanti dan Wahono, 2003:4).

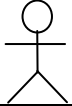
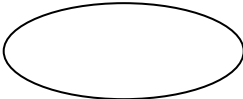

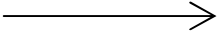

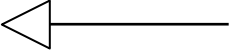
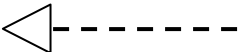
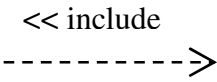
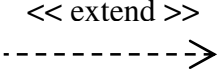
Menurut Munawar (2005:63) *use case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi

antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian skenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan aktor. Aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem. Aktor mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case*. Di dalam *use case* terdapat Stereotype yaitu sebuah model khusus yang terbatas untuk kondisi tertentu. Untuk menunjukkan *Stereotype* digunakan simbol "<<" di awalnya dan ">>" diakhirnya. <<*include*>> digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya. Biasanya <<*include*>> digunakan untuk menghindari penggandaan suatu *use case* karena sering dipakai. <<*extend*>> digunakan untuk menunjukkan bahwa suatu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu dipenuhi.

Di dalam *use case* juga terdapat generalisasi. Generalisasi diantara aktor adalah spesialisasi aktor yang bisa berpartisipasi di semua *use case* yang diasosiasikan dengan aktor yang lebih umum. Tabel 2.2 menampilkan notasi-notasi dalam pemodelan diagram *use case* menurut Booch, Rumbaugh, dan Jacobson (1998:187).

Tabel 2.2. Notasi pemodelan diagram use case

No.	Notasi	Keterangan
1		Aktor
2		Use Case
3		Batas sistem (<i>system boundary</i>)
4		Garis penghubung (pengendali arah)
5		Gabungan (<i>association</i>)
6		Generalisasi (<i>generalization</i>)
7		Realisasi (<i>realization</i>)
8		Stereotype penyertaan (<i>include</i>)
9		Stereotype perluasan (<i>extend</i>)

Menurut Munawar (2005:179) setiap *use case* harus dideskripsikan dalam dokumen yang disebut dengan dokumen aliran kejadian (*flow of event*). Dokumen ini mendefinisikan apa yang harus dilakukan oleh sistem ketika aktor mengaktifkan *use case*.

Struktur dari dokumen *use case* ini bisa macam-macam, tetapi umumnya deskripsi ini paling tidak harus mengandung:

1. Deskripsi singkat (*brief description*).
2. Aktor yang terlibat.
3. Kondisi awal (*precondition*) yang penting bagi use case untuk memulai.
4. Deskripsi rinci dari aliran kejadian yang mencakup:
 - a. Aliran utama (*main flow*) dari kejadian yang bisa dirinci lagi.
 - b. Aliran bagian (*sub flow*) dari kejadian.
 - c. Aliran alternatif untuk mendefinisikan situasi perkecualian.
5. Kondisi akhir yang menjelaskan state dari sistem setelah *use case* berakhir.



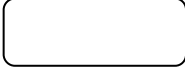



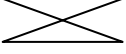

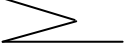
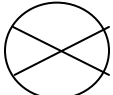
Dokumen *use case* ini berkembang selama masa pengembangan. Di awal-awal penentuan kebutuhan sistem, hanya deskripsi singkat saja yang ditulis. Bagian-bagian lain dari dokumen ini ditulis secara gradual dan iteratif. Akhirnya sebuah dokumen lengkap bisa didapatkan di akhir fase spesifikasi. Biasanya pada fase spesifikasi ini sebuah prototipe yang dilengkapi dengan tampilan layar bisa ditambahkan. Pada tahap berikutnya, dokumen *use case* ini bisa digunakan untuk membuat dokumentasi untuk implementasi sistem.

2.10.2 Diagram Aktifitas

Menurut Munawar (2005:109) diagram aktifitas adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Diagram aktifitas mempunyai peran seperti halnya diagram alur (*flowchart*), akan tetapi perbedaannya dengan *flowchart* adalah diagram aktifitas bisa mendukung perilaku paralel

sedangkan *flowchart* tidak bisa. Berikut pada Tabel 2.3 adalah simbol-simbol yang sering digunakan pada saat pembuatan diagram aktifitas.

Tabel 2.3. Simbol-simbol pada diagram aktifitas

No	Simbol	Keterangan
1.		Titik awal
2.		Titik akhir
3.		Aktifitas (<i>activity</i>)
4.		Pilihan untuk mengambil keputusan
5.		<i>Fork</i> ; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
6.		<i>Rake</i> ; menunjukkan adanya dekomposisi
7.		Tanda waktu
8.		Tanda pengiriman
9.		Tanda penerimaan
10.		Aliran akhir (<i>Flow final</i>)


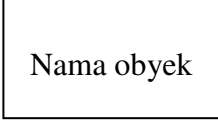
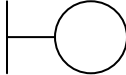
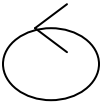

2.10.3 Diagram Sekuensial



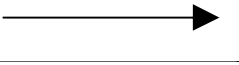


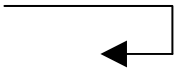
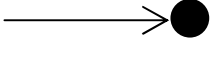
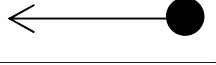
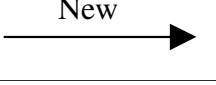
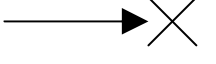
Menurut Munawar (2005:87) diagram sekuensial digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang diletakkan diantara objek-objek ini di dalam *use case*.

Komponen utama diagram sekuensial terdiri atas objek atau disebut juga peserta (*participant*) yang dituliskan dengan kotak segi empat bernama, pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan garis tegak lurus.

Setiap obyek terhubung dengan garis titik-titik yang disebut garis hidup (*lifeline*). Sepanjang garis hidup terdapat kotak yang disebut penggerakan (*activation*). Tabel 2.4 memperlihatkan notasi-notasi dalam pemodelan diagram sekuensial.

Tabel 2.4. Notasi pemodelan diagram sekuensial

No.	Notasi	Keterangan
1		Aktor
2		Obyek
3		Batas (<i>boundary</i>)
4		Kendali (<i>control</i>)
5		Entitas (<i>entity</i>)

6		Penggerakan (<i>activation</i>)
7		Garis hidup (<i>lifeline</i>)
8		Pesan selaras (<i>Synchronous message</i>)
9		Pesan tidak selaras (<i>Asynchronous message</i>)
10		Pesan kembali yang tidak selaras (<i>asynchronous return message</i>)
11		Pesan rekursif (<i>self message</i>)
12		Pesan hilang (<i>lost message</i>)
13		Pesan ditemukan (<i>found message</i>)
14		Pesan pembuatan obyek baru
15		Pesan penghapusan obyek

2.11 Interaksi Manusia dan Komputer

Berdasarkan pendapat Schneiderman [8], Interaksi Manusia dan Komputer (IMK) atau Human Computer Interaction (HCI) adalah disiplin ilmu yang berhubungan dengan perancangan, evaluasi, dan implementasi sistem komputer interaktif untuk digunakan oleh manusia, serta studi fenomena-fenomena besar yang berhubungan dengannya. Jadi dapat dikatakan bahwa interaksi manusia dan komputer dititik beratkan pada perancangan dan evaluasi antarmuka pemakai (*user interface*). Antarmuka pemakai adalah bagian sistem komputer yang memungkinkan manusia berinteraksi dengan komputer. Ada tiga kategori pedoman desain HCI, yaitu: (Pressman: 2002:471-474)

1. Interaksi umum. Pedoman interaksi umum diantaranya meliputi:
 - a. Konsisten: Gunakan format yang konsisten untuk misalnya pada pemilihan menu, masukan perintah, tampilan data.
 - b. Mintalah verifikasi terhadap sembarang aksi destruktif yang signifikan. Misal berikan pertanyaan kepada pemakai ketika ia meminta penghapusan *file*.
 - c. Ijinkan kemudahan pembatalan sebagian besar aksi.
 - d. Kurangi jumlah informasi yang harus diingat di antara aksi-aksi.
 - e. Usahakan adanya efisiensi dalam dialog, gerakan, dan pemikiran.
 - f. Memaafkan kesalahan: Sistem harus melindungi dirinya sendiri dari kesalahan yang dapat menyebabkan kegagalan pada sistem.
 - g. Kategorikan aktivitas menurut fungsi dan atur geografi layar secara sesuai.
 - h. Sediakan fasilitas *help*.
2. Tampilan informasi. Pedoman tampilan informasi diantaranya meliputi:
 - a. Hasilkan pesan kesalahan yang berarti.

- b. Gunakan huruf besar dan kecil untuk membantu pemahaman.
 - c. Jangan membanjiri pemakai dengan data, gunakan format yang representasi.
3. *Input* data. Pedoman-pedoman yang fokus pada *input* data diantaranya meliputi:
- a. Minimalkan jumlah aksi *input* yang dibutuhkan dari pemakai.
 - b. Jagalah konsistensi di antara tampilan informasi dan *input* data.
 - c. Sediakan *help* untuk membatu semua aksi *input*.

2.12 Pengujian

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean (Pressman, 2002:525).

2.12.1 Pengujian *Black Box*

Salah satu bentuk pengujian black-box adalah metode Partisi ekivalensi yaitu metode pengujian *black box* yang membagi domain *input* dari suatu program ke dalam kelas data dari mana *test case* dapat dilakukan (Pressman, 2002:556). Pengujian *black box* berusaha menemukan kesalahan diantaranya fungsi-fungsi yang hilang atau tidak benar, kesalahan *interface*, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan kinerja, inisialisasi dan kesalahan terminasi

2.12.2 Pengujian *White Box*

Pengujian *white box* ditujukan untuk mengetahui fungsi yang ditentukan dimana produk dirancang untuk melakukan pengujian dapat dilakukan untuk memperlihatkan

bahwa masing-masing fungsi beroperasi sepenuhnya, pada waktu yang sama mencari kesalahan-kesalahan pada setiap fungsi. Dengan metode ini perancang sistem dapat melakukan pengujian jalur yang memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali, menggunakan semua keputusan logis pada sisi benar (*true*) dan salah (*false*).

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis

Perkembangan teknologi sudah semakin pesat, bahkan hampir sebagian besar aktifitas dan kesibukan kita sehari-hari berhubungan dengan komputer. Seiring berkembangnya waktu dan pesatnya teknologi, kita dapat memanfaatkan fasilitas sarana hiburan khususnya aplikasi *player / audio*. Dengan membuat suatu aplikasi *player / audio* dan menambahkan fitur *equalizer*, dapat kita gunakan untuk mengurangi kepenatan atau stress dalam menjalani pekerjaan dan aktifitas sehari-hari.

Aplikasi ini dibuat sesuai kebutuhan untuk mendapatkan kualitas suara yang lebih baik. Dengan memutar file *audio* kita dapat mengatur frekuensi *equalizer* sesuai selera masing-masing pendengar.

Dalam membangun program aplikasi *player / audio* ini, penulis menganalisa akan kebutuhan dasar sistem antara lain :

1. Analisa keluaran : pada menu memutar file dapat menghasilkan suara atau lagu sebagai hasil proses memutar file yang memiliki format file MP3, MIDI, WAV, Mpeg dan Wma. Pada menu equalizer dan menu volume dapat mengubah atau mengatur tinggi rendahnya frekuensi.
2. Analisa masukan : sistem dapat membaca file suara atau lagu dalam aplikasi *player / audio*. Dan sistem dapat mengubah atau mengatur tinggi rendahnya frekuensi.

3. Proses

Proses-proses yang mesti terdapat pada aplikasi *player / audio* yang ingin dikembangkan adalah :

- a. Aplikasi dapat digunakan untuk memutar suara atau lagu berformat file MP3, MIDI, WAV, Mpeg dan Wma.
- b. Aplikasi dapat mengubah atau mengatur tinggi rendahnya frekuensi khususnya pada equalizer dan volume.

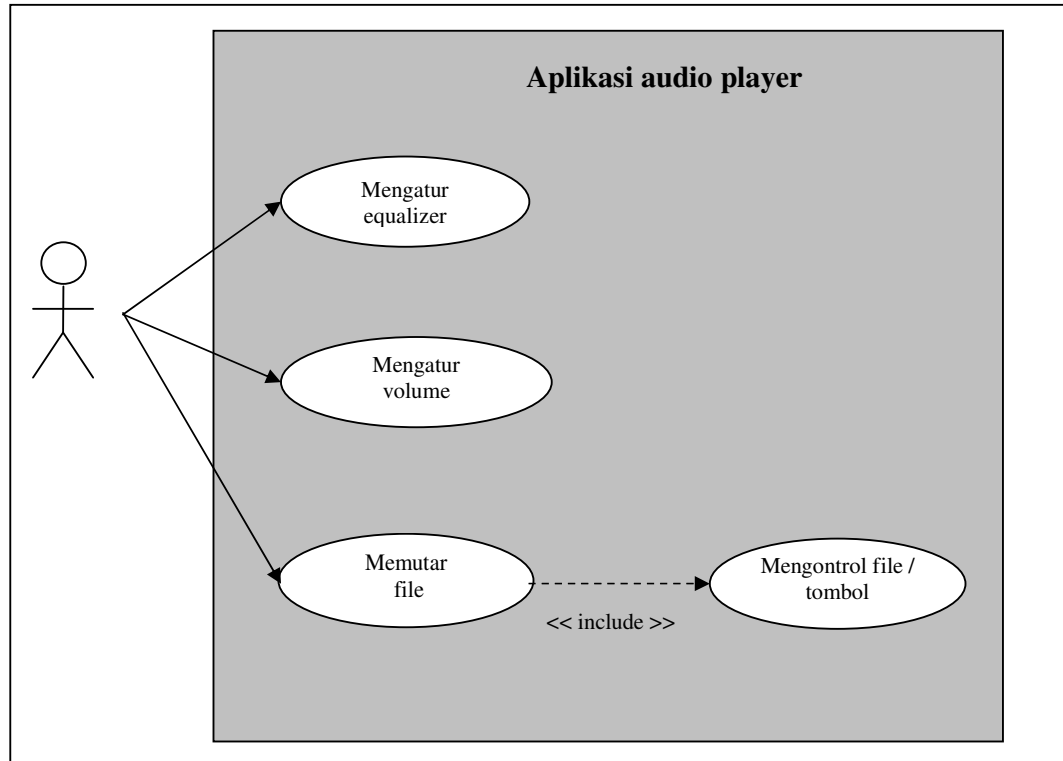
3.2 Perancangan

Pada bagian ini akan dijelaskan secara rinci perancangan aplikasi *player* meliputi pemodelan *use case diagram*, *sequence diagram* dan *activity diagram*.

3.2.1 Pemodelan Diagram Use Case

Use case diagram menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada di luar sistem (*actor*), diagram *use case* juga menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia

luar. Secara umum pemodelan *Use case* pada aplikasi ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram *Use Case*

Tabel 3.1. Spesifikasi naratif untuk Use Case mengatur equalizer

Use Case	: Mengatur equalizer
Deskripsi Singkat	: Use Case ini memungkinkan user untuk mengatur frekuensi equalizer dengan cara menekan dan menarik tombol pada mouse
Aktor	: User
Kondisi Awal	: Suara atau lagu telah dimulai dan sedang berjalan
Aliran Utama	: 1. user menekan dan menarik tombol pada mouse untuk mengatur frekuensi equalizer

Tabel 3.1. (lanjutan) Spesifikasi naratif untuk Use Case mengatur equalizer

	:	2. User dapat memilih frekuensi equalizer yang mana akan diatur dari pilihan kelima frekuensi equalizer
Aliran Alternatif	:	-
Kondisi Akhir	:	Jika use case berhasil dijalankan maka User dapat mendapatkan frekuensi equalizer yang diinginkan. Namun jika Use Case tidak dijalankan, maka frekuensi equalizer tidak berubah.

Tabel 3.2. Spesifikasi naratif untuk Use Case mengatur volume

Use Case	:	Mengatur volume
Deskripsi Singkat	:	Use Case ini memungkinkan user untuk mengatur frekuensi volume dengan cara menekan dan menarik tombol pada mouse
Aktor	:	User
Kondisi Awal	:	Suara atau lagu telah dimulai dan sedang berjalan
Aliran Utama	:	1. User menekan dan menarik tombol pada mouse untuk mengatur frekuensi volume
Aliran Alternatif	:	-
Kondisi Akhir	:	Jika use case berhasil dijalankan maka User dapat mendapatkan frekuensi volume yang diinginkan. Namun jika Use Case tidak dijalankan, maka frekuensi volume tidak berubah

Tabel 3.3. Spesifikasi naratif untuk Use Case memutar file

Use Case	:	Memutar file
Deskripsi Singkat	:	Use Case ini memungkinkan user untuk memutar file suara atau lagu

Tabel 3.3. (lanjutan) Spesifikasi naratif untuk Use Case memutar file

Aktor	:	User
Kondisi Awal	:	File (suara atau lagu) belum berjalan
Aliran Utama	:	<ol style="list-style-type: none"> 1. User mengakses menu dan membuka file, kemudian memilih directori musik yang ada pada komputer 2. User memilih tombol play untuk menjalankan file 3. User dapat menghentikan atau mengakhiri file dengan menekan tombol pause atau stop
Aliran Alternatif	:	-
Kondisi Akhir	:	Jika Use Case berhasil dijalankan maka user dapat mendengarkan file yang diputar yakni suara atau lagu. Namun jika Use Case tidak dijalankan, maka user tidak dapat mendengarkan file (suara atau lagu)

Tabel 3.4. Spesifikasi naratif untuk Use Case mengontrol file / tombol

Use Case	:	Mengontrol file / tombol
Deskripsi Singkat	:	Use Case ini memungkinkan user untuk mengontrol file / tombol
Aktor	:	User
Kondisi Awal	:	Suara atau lagu telah dimulai dan sedang berjalan
Aliran Utama	:	<ol style="list-style-type: none"> 1. User mengklik tombol play untuk memutar file 2. User mengklik tombol pause untuk menghentikan file 3. User mengklik tombol stop untuk mengakhiri file
Aliran Alternatif	:	-

Tabel 3.4. (lanjutan) Spesifikasi naratif untuk Use Case mengontrol file / tombol

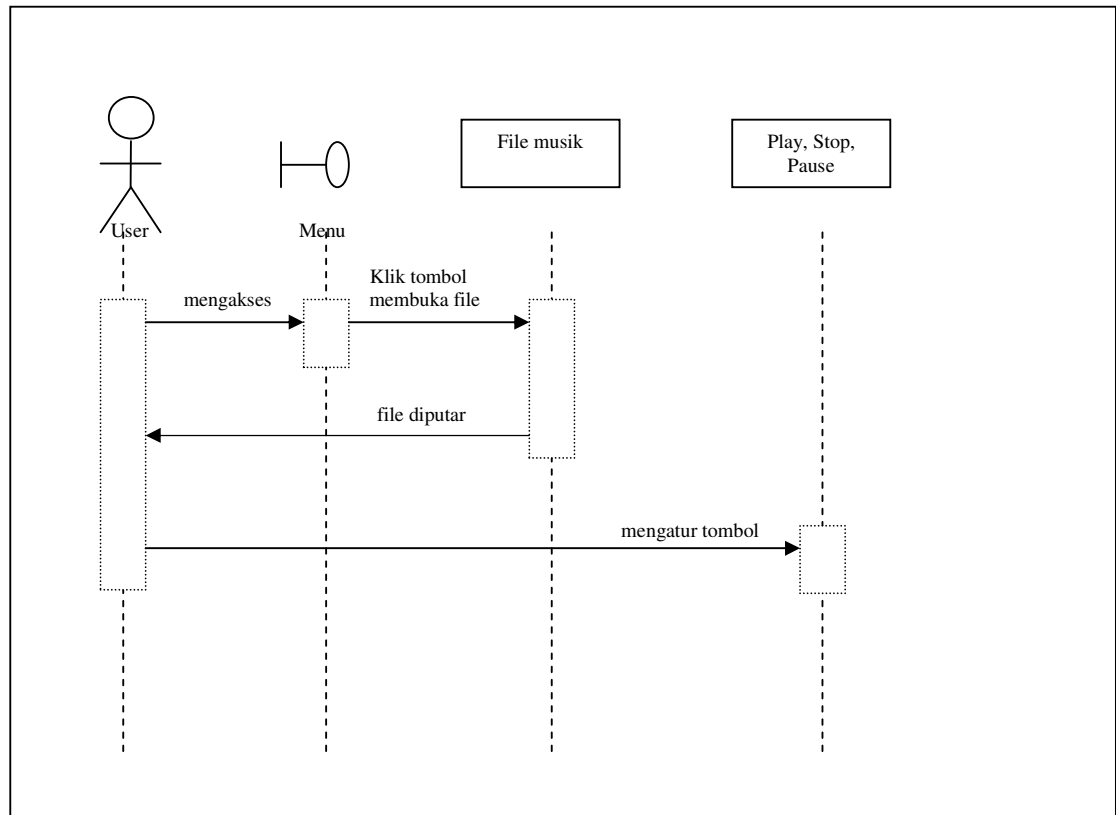
Kondisi Akhir	: Jika Use Case berhasil dijalankan maka user dapat mendengarkan file yang diputar yakni suara atau lagu. Namun jika Use Case tidak dijalankan, maka user tidak dapat mendengarkan file (suara atau lagu)
---------------	---

3.2.2 Pemodelan Diagram *Sequence*

Diagram *Sequence* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna) berupa pesan (*message*) yang digambarkan terhadap waktu.

3.2.2.1 Diagram *Sequence* untuk Use Case memutar file

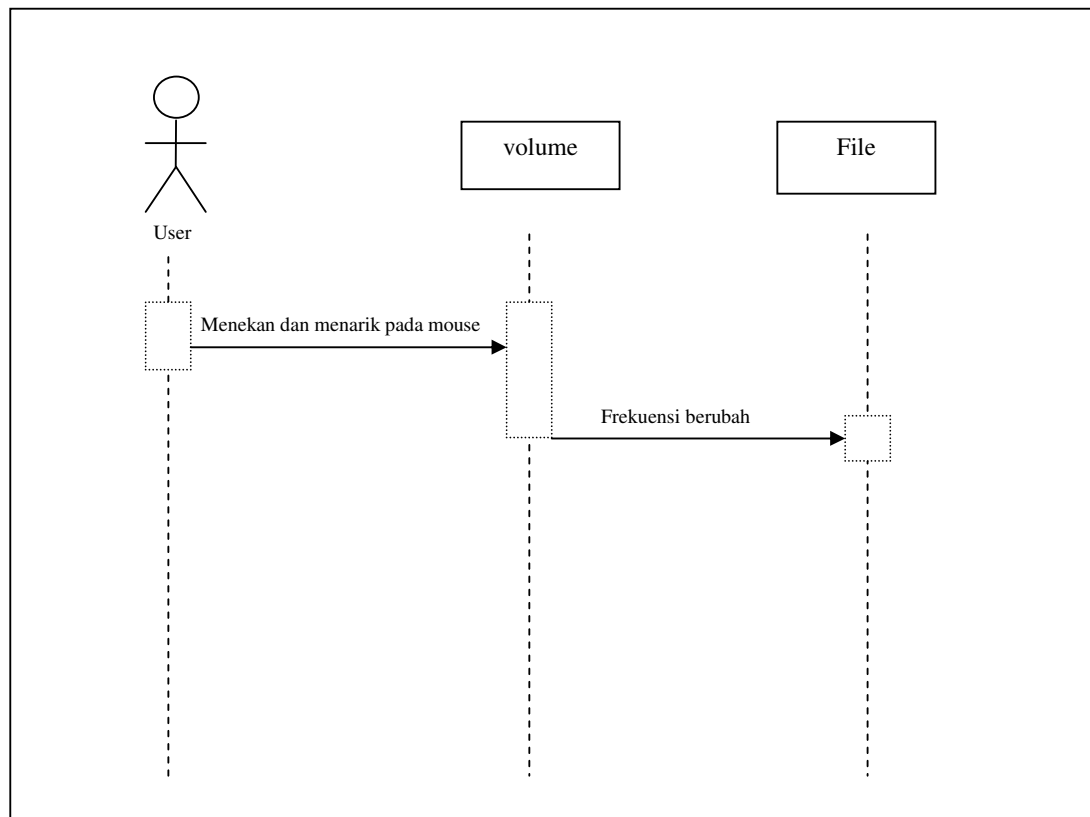
User dapat memutar file yang memiliki format file MP3, MIDI, WAV, Mpeg dan Wma. dengan terlebih dahulu mengakses menu open file kemudian mencari alamat file directori musik setelah dapat file musik yang diinginkan dapat berjalan. Setelah file musik berjalan user dapat mengatur tombol yakni menghentikan atau mengakhiri file dengan menggunakan tombol yang telah tersedia. Untuk lebih jelasnya dapat dilihat pada Gambar 3.2



Gambar 3.2 Diagram *Sequence* untuk *Use Case* memutar file

3.2.2.2 Diagram *Sequence* untuk *Use Case* mengatur volume

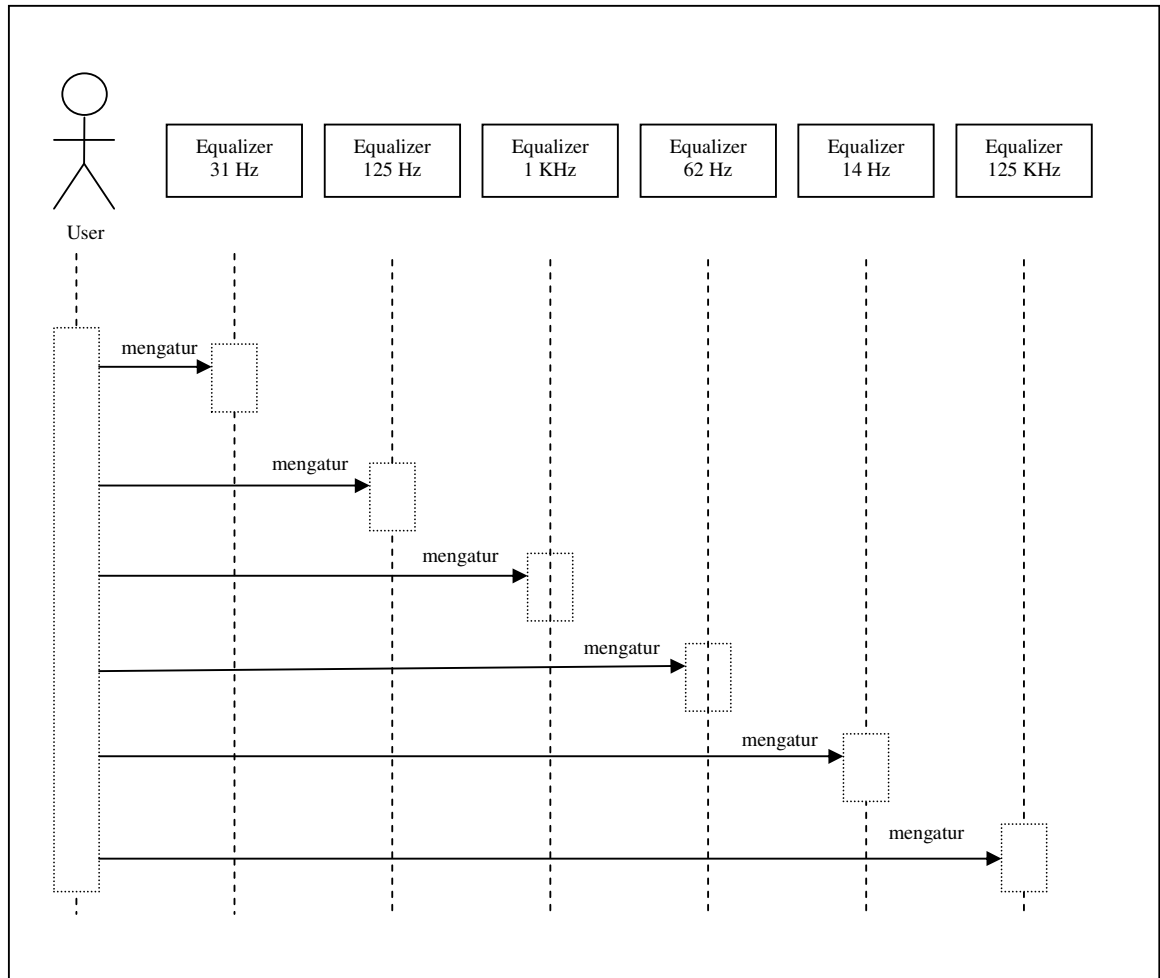
User dapat mengatur volume dengan cara menekan dan menarik tombol mouse pada bagian volume musik, secara otomatis file musik yang sedang berjalan akan berubah frekuensinya sesuai kebutuhan user. Untuk lebih jelasnya dapat dilihat pada Gambar 3.3



Gambar 3.3 Diagram *Sequence* untuk *Use Case* mengatur *volume*

3.2.2.3 Diagram *Sequence* untuk *Use Case* mengatur *equalizer*

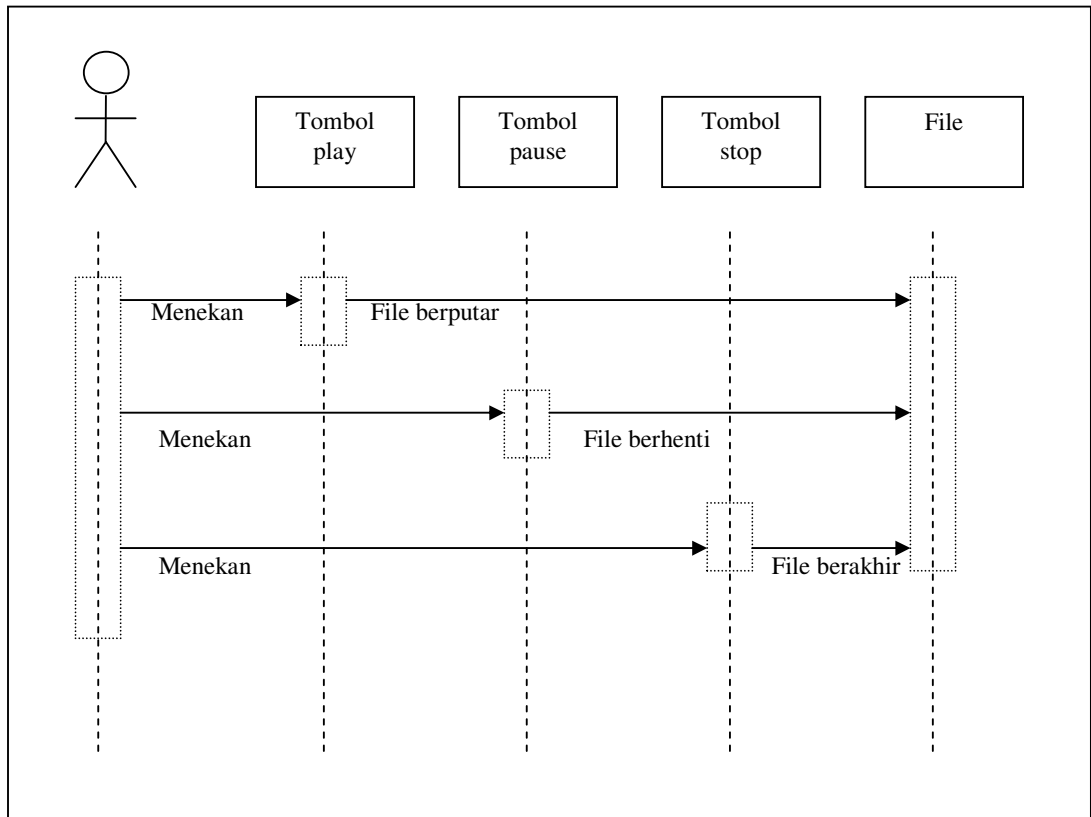
User dapat mengatur *equalizer* dengan cara memilih dari 6 *equalizer* yang ada yang mana user inginkan ada perubahan. Setiap *equalizer* memiliki frekuensi yang berbeda yakni 31Hz, 125 Hz, 1KHz, 62Hz, 14Hz dan 125 KHz. Untuk lebih jelasnya dapat dilihat pada Gambar 3.4



Gambar 3.4 Diagram *Sequence* untuk *Use Case* mengatur equalizer

3.2.2.4 Diagram *Sequence* untuk *Use Case* mengontrol file atau tombol

User dapat menekan tombol play maka file akan berputar, jika tombol pause maka file akan berhenti dan jika user menekan tombol stop maka file berakhir. Untuk lebih jelasnya dapat dilihat pada Gambar 3.5



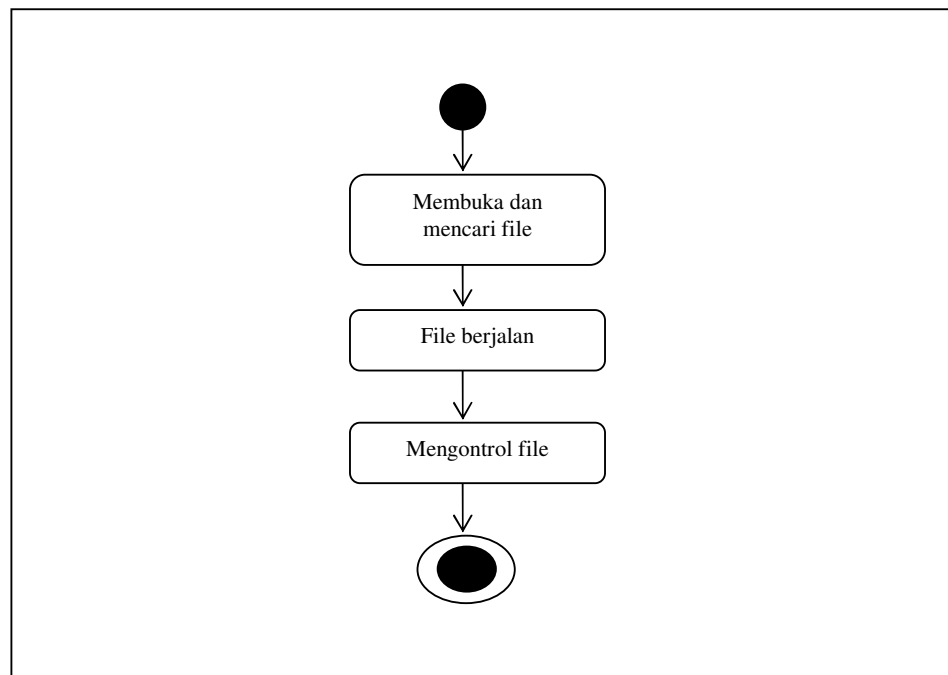
Gambar 3.5 Diagram *Sequence* untuk *Use Case* mengontrol file atau tombol

3.2.3 Pemodelan Diagram *Activity*

Diagram *activity*/aktifitas merupakan pemodelan aliran kerja (*workflow*) sebuah proses, atau urutan aktifitas dalam suatu proses. Dengan diagram aktifitas ini kita akan lebih dapat memahami alur dari proses-proses tersebut atau dengan kata lain diagram ini akan menggambarkan langkah-langkah dari setiap aplikasi yang disediakan dalam sistem.

3.2.3.1 Diagram *Activity* untuk *Use Case* memutar file

Diagram aktifitas untuk use case memutar file diawali dengan membuka dan mencari file pada directori musik, kemudian barulah file dapat berjalan. Pada saat file berjalan user dapat mengontrol file dengan menghentikan file dan mengakhirinya Untuk lebih jelasnya dapat dilihat pada Gambar 3.6

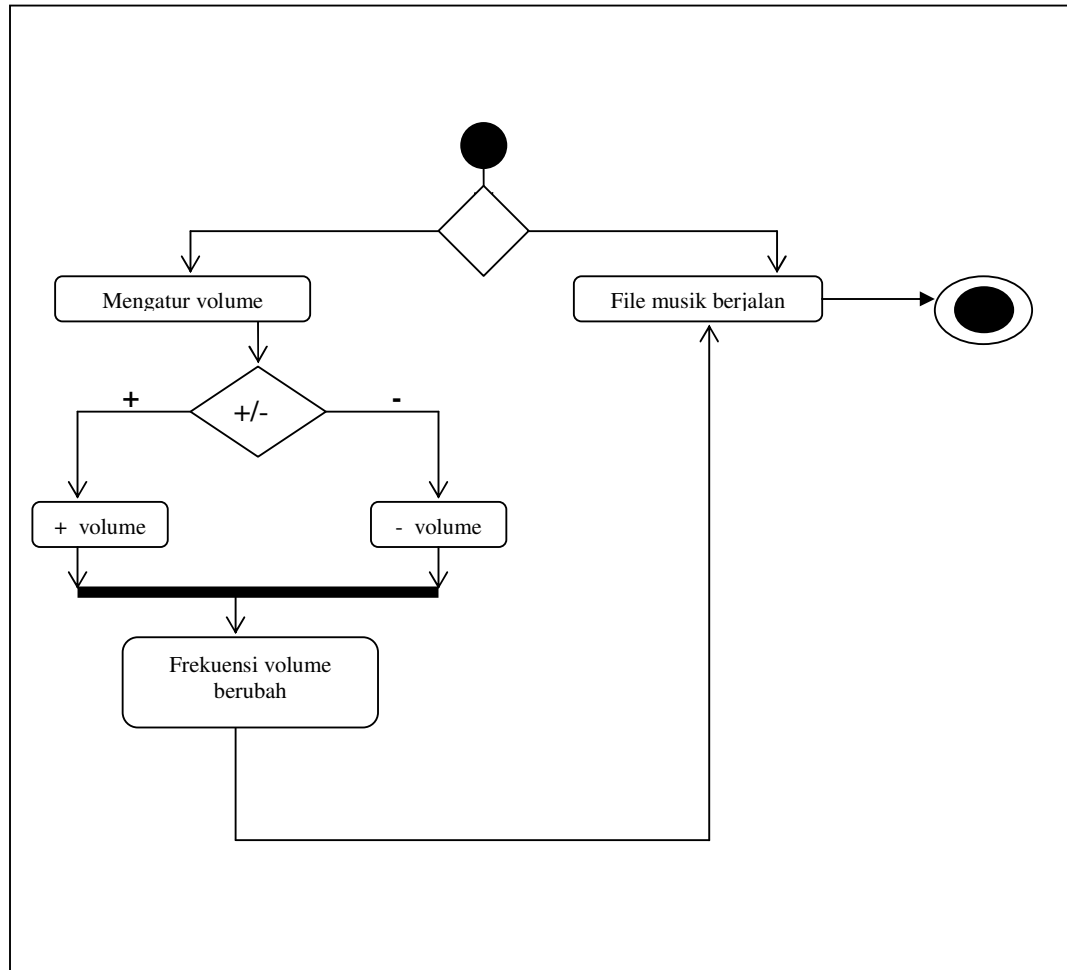


Gambar 3.6 Diagram aktivitas untuk *Use Case* memutar file

3.2.3.2 Diagram *Activity* untuk *Use Case* mengatur volume

Diagram aktifitas untuk use case mengatur volume diawali pada saat file musik sedang berjalan, apabila user menginginkan frekuensi volume berubah maka user harus menekan dan menarik tombol mouse pada bagian volume musik, tetapi jika

tidak file tetap berjalan dengan frekuensi volume awal. Selanjutnya file berakhir. Untuk lebih jelasnya dapat dilihat pada Gambar 3.7

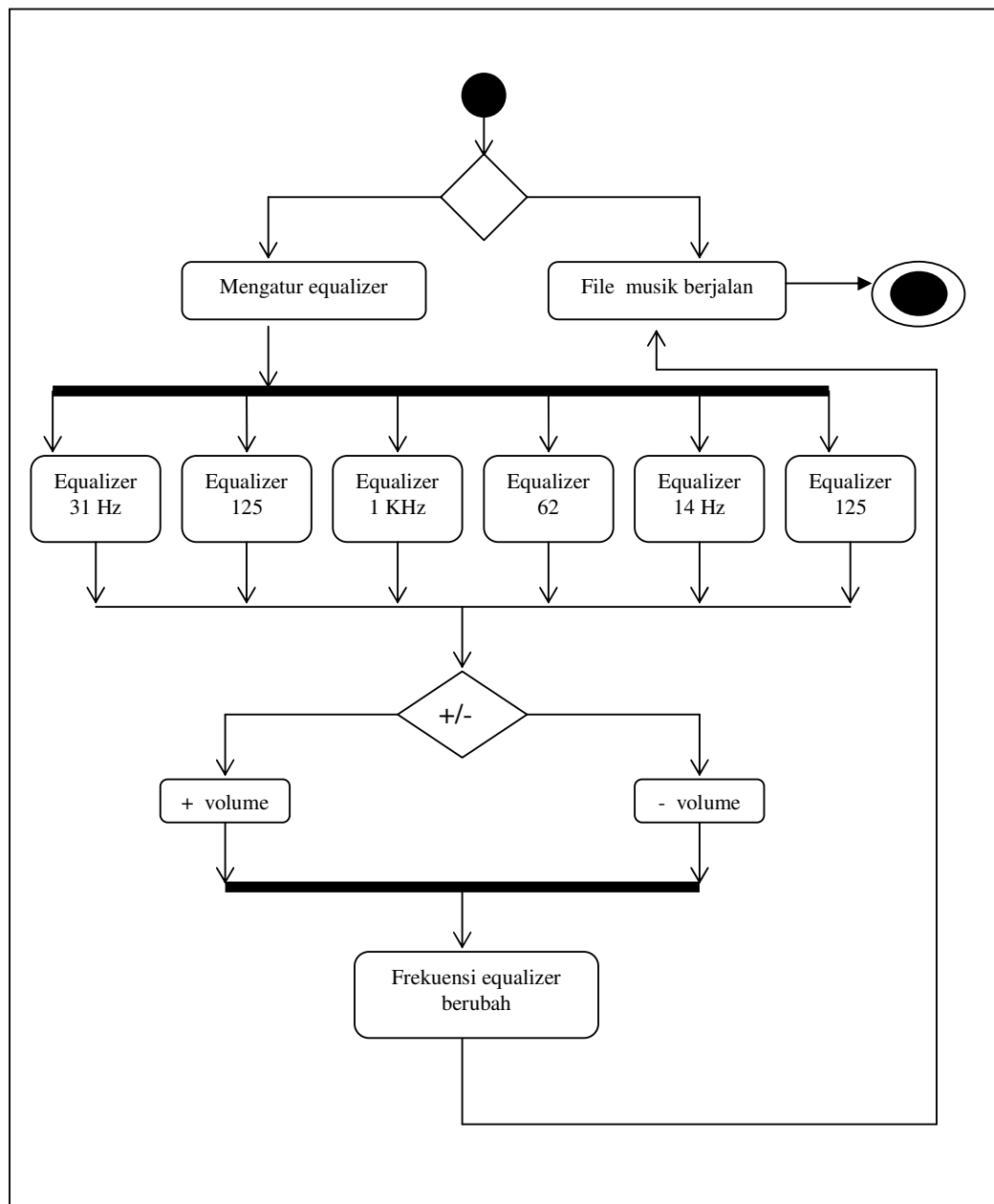


Gambar 3.7 Diagram aktivitas untuk *Use Case mengatur volume*

3.2.3.3 Diagram Activity untuk *Use Case mengatur equalizer*

Diagram aktifitas untuk use case *equalizer* diawali pada saat file musik sedang berjalan, apabila user menginginkan frekuensi *equalizer* berubah maka user harus menekan dan menarik tombol mouse dan memilih dari 6 *equalizer* musik yang ada, tetapi jika tidak file tetap berjalan dengan frekuensi *equalizer* awal. Setiap *equalizer*

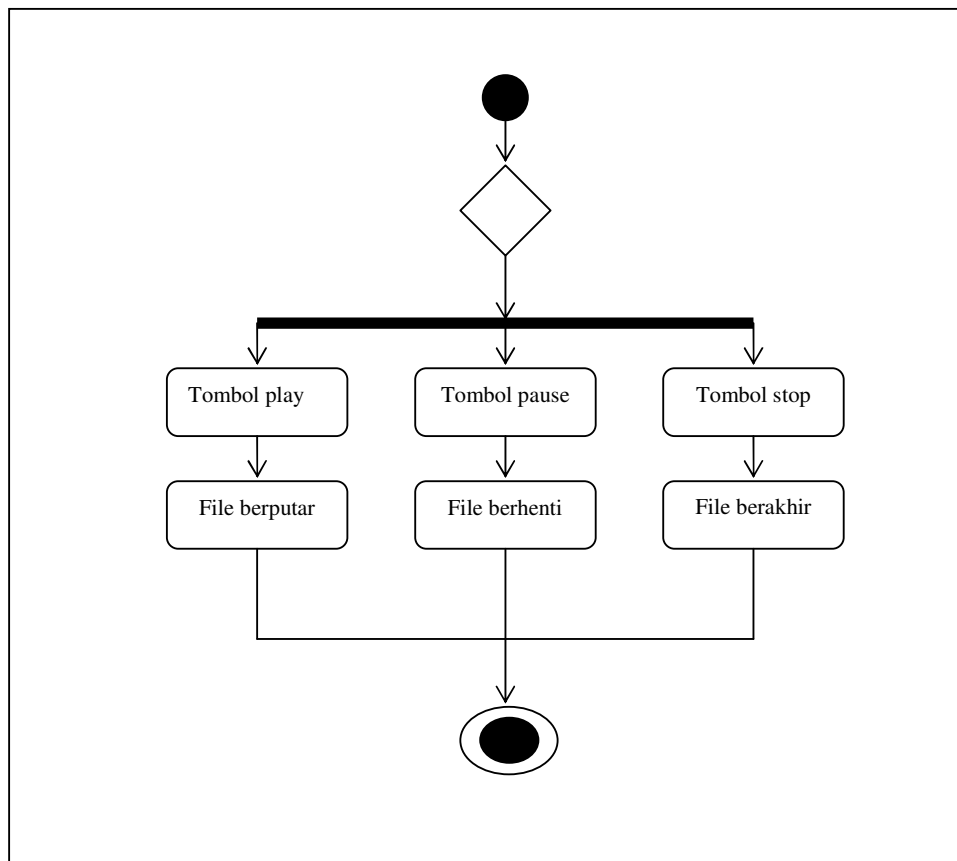
memiliki frekuensi yang berbeda yakni 31Hz, 125Hz, 1KHz, 62Hz, 14Hz dan 125KHz. Selanjutnya file berakhir. Untuk lebih jelasnya dapat dilihat pada Gambar 3.8



Gambar 3.8 Diagram aktivitas untuk *Use Case* mengatur equalizer

3.2.3.4 Diagram *Activity* untuk *Use Case* mengontrol file atau tombol

Diagram aktifitas untuk use case mengontrol file atau tombol diawali dengan mengambil keputusan jika memilih tombol play maka file akan berputar, jika memilih tombol pause maka file berhenti dan jika memilih tombol stop maka file berakhir. Selanjutnya file musikpun berakhir. Untuk lebih jelasnya dapat dilihat pada Gambar 3.9



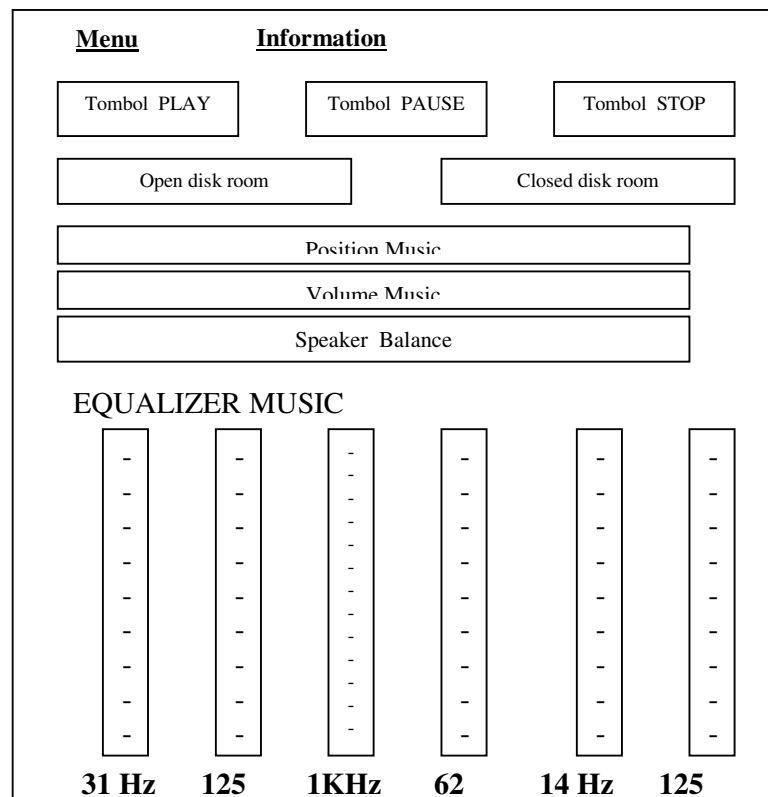
Gambar 3.9 Diagram aktivitas untuk *Use Case* mengontrol file atau tombol

3.3 Perancangan Tampilan Pada Aplikasi *Audio Player*

Pada tahap ini penulis merancang terlebih dahulu aplikasi *audio player* yang akan dibuat, aplikasi ini dirancang sesuai kebutuhan dan dirancang dengan tampilan sederhana.

3.3.1 Rancangan Menu Utama

Rancangan ini dibuat sangat sederhana dan mempunyai fungsi yang dapat memudahkan pengguna awal dalam penggunaan aplikasi ini. Di kiri atas terdapat menu utama dimana user dapat membuka file musik. Pada bagian badan aplikasi ini terdapat tombol *play*, *pause* dan *stop*, serta terdapat frekuensi *equalizer* dan volume yang dapat di ubah sesuai keinginan user pada saat musik sedang berjalan. Lihat gambar 3.10



Gambar 3.10 Rancangan menu utama

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Setelah melakukan analisis dan perancangan sistem terhadap aplikasi audio player yang akan dibangun, tahapan selanjutnya adalah implementasi dan pengujian, pada tahapan implementasi sistem terdapat dua kebutuhan yaitu spesifikasi kebutuhan sistem yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*), dan implementasi dari aplikasi audio player ini meliputi lingkungan implementasi, proses pengkodean, dan tampilan antarmuka.

4.1.1 Lingkungan Implementasi

Aplikasi audio player dibangun dan diimplementasikan pada seperangkat komputer pribadi (*Personal Computer*) dengan spesifikasi sebagai berikut:

1. Perangkat keras (*Hardware*) yang digunakan adalah:
 - a. Processor : Intel Pentium IV 2.40 GHz.

- b. Motherboard : Asus PT800/P4X533
 - c. Memori : 640 MB DDRAM
 - d. VGA : NVIDIA GeForce2 MX 64 MB
 - e. Harddisk : Seagete 40 GB 7200 RPM
2. Perangkat lunak (*Software*) yang digunakan adalah:
- a. Sistem operasi Microsoft Windows XP Profesional.
 - b. Microsoft Visual Basic 6.0 edisi professional, sebagai alat (*tool*) untuk membangun aplikasi audio player tersebut.

4.1.2 Pengkodean

Pengkodean diperlukan agar rancangan dapat diterjemahkan ke dalam bentuk mesin yang dapat dibaca dan dapat dijalankan, pada subbab ini penulis akan memberikan penjelasan dari beberapa penggalan baris kode program aplikasi audio player yang dibangun dengan menggunakan bahasa pemrograman Visual Basic versi 6.0. Penggalan kode program tersebut diantaranya :

1. Potongan kode program untuk open file

Sebelum memulai aplikasi audio player ini user harus terlebih dahulu membuka file pada directori musik di *personal computer* (PC). Berikut ini adalah potongan kode program untuk membuka file :

```
Private Sub Open_File_Click()
Dim Music
Dialog1.Filter = "All Video & Audio "
Dialog1.Flags = cdlOFNAllowMultiselect
Dialog1.ShowOpen
System_Player.FileName = Dialog1.FileName
Position_Music.Max = System_Player.MediaLenght
Timer1.Enabled = True
End Sub
```

2. Potongan kode program untuk mengatur volume

Pada saat aplikasi berjalan user dapat mengatur volume sesuai kebutuhan.

Berikut ini adalah potongan kode program untuk mengatur volume :

```
Private Sub Volume_Control_Scroll()
    Position_Music.VolumeSet (Volume_Control.Value)
    Volume_Control.ToolTipText = "Volume Control"
End Sub
```

3. Potongan kode program untuk mengatur *equalizer*

Pada saat aplikasi berjalan user dapat mengatur equalizer sesuai kebutuhan.

Berikut ini adalah potongan kode program untuk mengatur equalizer :

```
Private Sub Equalizer_Control_Scroll(Index As Integer)
    System_Player.SetEq = Index
    System_Player.ValEq = Equalizer_Control(Index).Value
End Sub
```

4. Potongan kode program untuk tombol *play*

Setelah kita membuka file dan mendapatkan file yang kita inginkan maka user dapat memutar file dengan menekan tombol *play*. Berikut ini adalah potongan kode program untuk tombol *play* :

```
Private Sub Play_Click()
    System_Player.MediaPlay
End Sub
```

5. Potongan kode program untuk tombol *pause*

Pada saat aplikasi berjalan dan jika user ingin menghentikannya maka user dapat menekan tombol *pause*. Berikut ini adalah potongan kode program untuk tombol *pause* :

```
Private Sub Pause_Click()  
System_Player.MediaPause  
End Sub
```

6. Potongan kode program untuk tombol *stop*

Pada saat aplikasi berjalan dan jika user ingin mengakhirinya maka user dapat menekan tombol *stop*. Berikut ini adalah potongan kode program untuk tombol

stop :

```
Private Sub Stop_Click()  
System_Player.MediaStop  
End Sub
```

7. Potongan kode program untuk menu exit

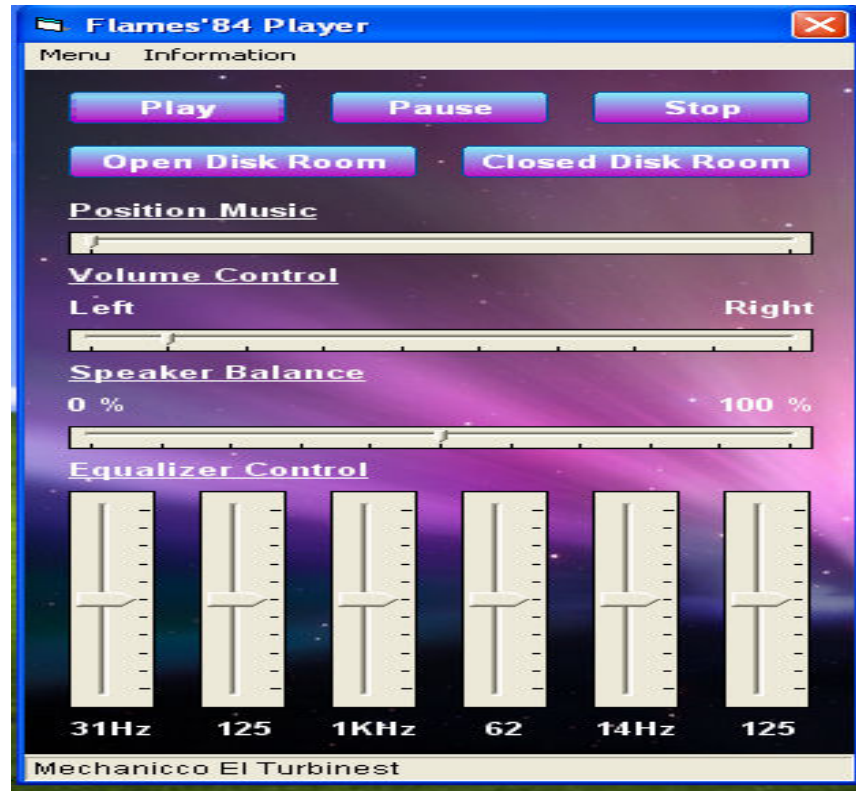
Apabila user ingin keluar pada aplikasi ini maka user bisa menggunakan menu exit. Berikut ini adalah potongan kode program untuk menu exit :

```
Private Sub Exit_Click()  
System_Player.DoStop  
Unload Me  
End Sub
```

4.1.3 Antarmuka

Pada sub bab ini penulis akan menampilkan beberapa implementasi antarmuka, diantaranya :

1. Tampilan Menu utama memutar file

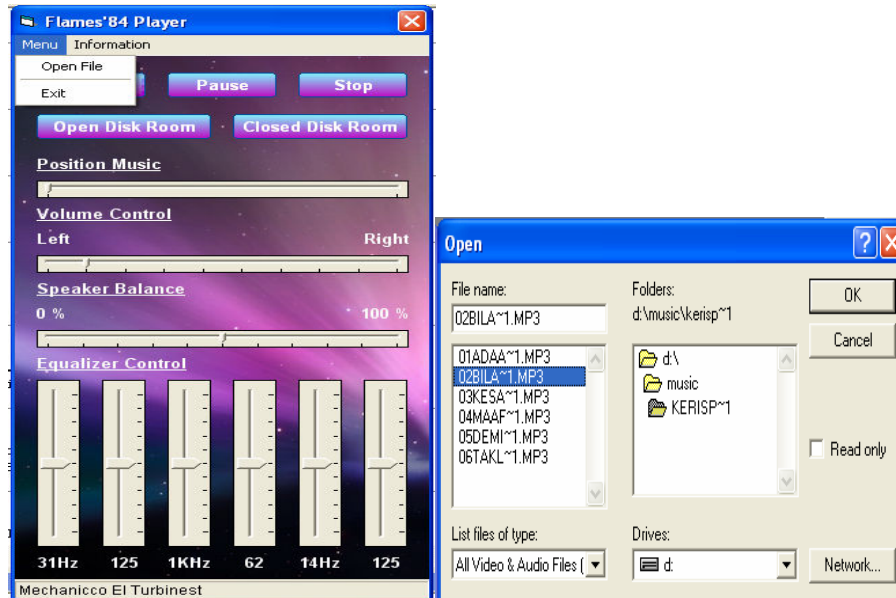


Gambar 4.1 Tampilan Menu utama memutar file

Pada menu ini terdapat menu open file yang terletak paling atas sebelah kiri dan menu informasi yang terletak paling atas sebelah kanan. Dibawahnya terdapat tombol-tombol *play*, *pause*, *stop*, *open disk room*, dan *closed disk room*. Selanjutnya terdapat position music yang berfungsi untuk memindahkan posisi musik pada saat aplikasi berjalan. Volume control yang berfungsi untuk mengatur volume dan *speaker balance* yang berfungsi untuk mengatur keseimbangan suara *speaker*. Dan pada bagian paling bawah terdapat menu *equalizer* yang terdiri dari 6 yakni 31Hz, 125 Hz, 1KHz, 62Hz, 14Hz dan 125 KHz.

2. Menu Open File

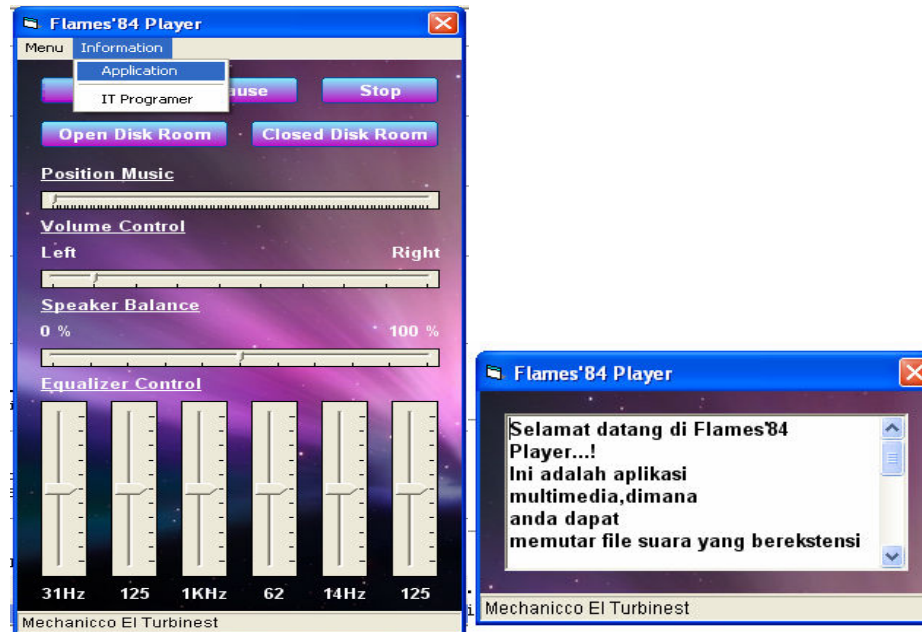
Pada menu ini user memilih menu dan mengklik open file dan mencari file yang diinginkan pada directori yang ada pada komputer. Untuk lebih jelas dapat kita lihat pada gambar 4.2.



Gambar 4.2 Tampilan Menu open file

3. Menu Informasi Application dan Menu informasi IT Programer

Pada menu ini terdapat tampilan informasi tentang aplikasi yang dibuat dan menu informasi IT Programer. Untuk lebih jelas dapat kita lihat pada gambar 4.3 dan gambar 4.4.



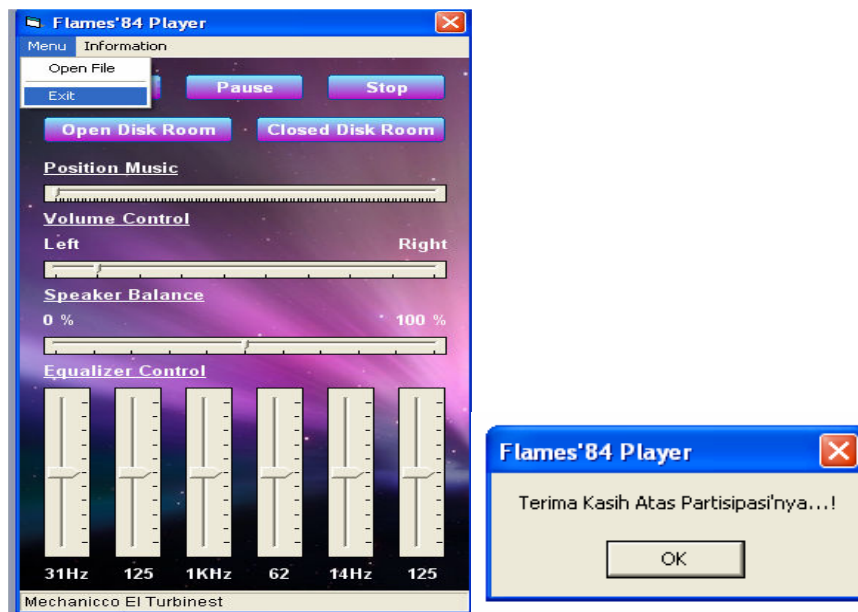
Gambar 4.3 Tampilan Menu Informasi Application



Gambar 4.4 Tampilan Menu Informasi IT Programer Application

4. Menu Exit

Apabila user ingin keluar dari aplikasi ini maka user dapat mengklik menu exit dan mengklik tombol oke. Untuk lebih jelas dapat kita lihat pada gambar 4.5



Gambar 4.5 Tampilan Menu Exit

4.2 Pengujian

Pengujian program ini dilakukan untuk menguji dan memastikan bahwa program yang telah dirancang dan dibuat dapat berfungsi dengan baik dan sesuai dengan rumusan rancangan yang telah ditetapkan sebelumnya. Tanpa adanya pengujian, maka tidak dapat diketahui apakah program yang telah dibuat sesuai dengan spesifikasi rancangan program aplikasi. Pengujian terhadap aplikasi ini dilakukan dengan menggunakan metode *Black Box*. Pengujian *Black Box* berusaha menemukan kesalahan dalam lingkup kategori sebagai berikut: (1) fungsi-fungsi

yang tidak benar atau hilang, (2) kesalahan *interface*, (3) kesalahan dalam struktur data atau akses basis data, dan (4) kesalahan kinerja (Pressman, 2002:551-552).

Pengujian-pengujian yang akan dilakukan terdiri atas skenario pengujian, hasil pengujian, dan analisis hasil pengujian yang akan dijelaskan pada subbab berikut ini.

4.2.1 Skenario Pengujian

Skenario pengujian digunakan untuk menentukan langkah-langkah dalam melakukan pengujian. Pengujian dilakukan dengan menjalankan aplikasi *audio player* terlebih dahulu. Kemudian dilakukan pengujian dengan menekan tombol-tombol yang terdapat pada jendela-jendela dari aplikasi *audio player* tersebut apakah sesuai dengan yang diharapkan. Selanjutnya pengujian dilakukan ketika aplikasi *audio player* sedang berjalan.

Skenario pengujian yang dilakukan ada 2 yaitu yang pertama adalah fungsi-fungsi yang tidak benar atau hilang dan yang kedua adalah kesalahan *interface*. Sedangkan kesalahan struktur data atau akses basis data dan kesalahan kinerja tidak dibuat skenario pengujian karena penulis tidak membahas struktur data dan kesalahan kinerjanya. Skenario pengujian terhadap aplikasi *audio player* tersebut yang lebih lengkap dapat dilihat pada Tabel 4.1 dan Tabel 4.2 di bawah ini.

Tabel 4.1. Tabel skenario pengujian fungsi

No	Antarmuka yang diuji	Bagian dari Antarmuka yang diuji	Status atau kondisi	Skenario pengujian	Hasil yang diharapkan
1.	Jendela menu utama	Fungsi equalizer	Aplikasi belum atau sedang berjalan	Menekan dan menarik mouse pada equalizer kontrol	mengatur equalizer sesuai kebutuhan user
2.	Jendela menu utama	Fungsi volume	Aplikasi belum atau	Menekan dan menarik	mengatur volume sesuai kebutuhan user

Tabel 4.1. (lanjutan) Tabel skenario pengujian fungsi

			sedang berjalan	mouse pada volume kontrol	
3.	Jendela menu utama	Fungsi posisi musik	Aplikasi sedang berjalan	Menekan dan menarik mouse pada volume kontrol	mengatur posisi musik sesuai kebutuhan user
4.	Jendela menu utama	Fungsi speaker balance	Aplikasi belum atau sedang berjalan	Menekan dan menarik mouse pada volume kontrol	mengatur speaker balance sesuai kebutuhan user

Tabel 4.2. Tabel skenario pengujian kesalahan interface

1.	Jendela menu utama	Menu open file	Aplikasi belum berjalan	Klik menu open file dan mencari file pada directori di komputer	file musik yang ingin dijalankan
2.	Jendela menu utama	Menu exit	Aplikasi belum atau sedang berjalan	Klik menu exit untuk keluar dari aplikasi	keluar dari aplikasi
3.	Jendela menu utama	Tombol play	Aplikasi atau musik belum berjalan	Klik tombol play	Aplikasi atau musik berjalan
4.	Jendela menu utama	Tombol pause	Aplikasi atau musik sedang berjalan	Klik tombol pause	Aplikasi atau musik berhenti
5.	Jendela menu utama	Tombol stop	Aplikasi atau musik sedang berjalan	Klik tombol stop	Aplikasi atau musik berakhir
6.	Jendela menu utama	informasi application	Aplikasi belum atau sedang berjalan	Klik informasi dan application	
7.	Jendela menu utama	Informasi IT programmer	Aplikasi belum atau sedang berjalan	Klik informasi dan IT programmer	

4.2.2 Hasil Pengujian

Dari hasil skenario pengujian maka dapat diperoleh hasil pengujian dari aplikasi *audio player* tersebut. Untuk lebih jelasnya lihat Tabel 4.2.

Tabel 4.3. Tabel hasil pengujian fungsi

No	Antarmuka yang diuji	Bagian dari Antarmuka yang diuji	Status atau kondisi	Hasil pengujian
1.	Jendela menu utama	Fungsi equalizer	Aplikasi belum atau sedang berjalan	Dapat mengatur equalizer sesuai kebutuhan user
2.	Jendela menu utama	Fungsi volume	Aplikasi belum atau sedang berjalan	Dapat mengatur volume sesuai kebutuhan user
3.	Jendela menu utama	Fungsi posisi musik	Aplikasi sedang berjalan	Dapat mengatur posisi musik sesuai kebutuhan user
4.	Jendela menu utama	Fungsi speaker balance	Aplikasi belum atau sedang berjalan	Dapat mengatur speaker balance sesuai kebutuhan user

Tabel 4.4. Tabel hasil pengujian kesalahan interface

1.	Jendela menu utama	Menu open file	Aplikasi belum berjalan	User dapat file musik yang ingin dijalankan atau dapat memutar file musik
2.	Jendela menu utama	Menu exit	Aplikasi belum atau sedang berjalan	Dapat keluar dari aplikasi
3.	Jendela menu utama	Tombol play	Aplikasi atau musik belum berjalan	Dapat menjalankan Aplikasi atau musik
4.	Jendela menu utama	Tombol pause	Aplikasi atau musik sedang berjalan	Dapat menghentikan Aplikasi atau musik yang sedang berjalan
5.	Jendela menu utama	Tombol stop	Aplikasi atau musik sedang berjalan	Dapat mengakhiri Aplikasi atau musik yang sedang berjalan
6.	Jendela menu utama	informasi application	Aplikasi belum atau sedang berjalan	Dapat mengetahui tentang aplikasi yang telah dibuat.

Tabel 4.4. (lanjutan) Tabel hasil pengujian kesalahan interface

7.	Jendela menu utama	Informasi IT programmer	Aplikasi belum atau sedang berjalan	Dapat mengetahui tentang IT programmer
----	--------------------	-------------------------	-------------------------------------	--

4.2.3 Analisis Hasil Pengujian

Setelah melakukan pengujian terhadap semua menu aplikasi ini, maka dapat disimpulkan bahwa pengujian telah menunjukkan hasil keluaran (output) dan proses yang sesuai dengan rumusan rancangan aplikasi program ini. Dari hasil pengujian tersebut dapat dikatakan bahwa program ini telah berjalan sesuai dengan skenario pengujian. Hasil pengujian fungsi yang berjalan adalah fungsi *equalizer*, fungsi volume, fungsi posisi musik dan fungsi *speaker balance* sedangkan hasil pengujian kesalahan interface yang berjalan adalah aplikasi dapat membuka file lalu memutar musik, menggunakan tombol yang ada (tombol *play*, *pause* dan *stop*) dan membuka menu informasi yang berisi informasi aplikasi dan informasi IT programmer.

Namun berdasarkan pengujian secara keseluruhan didapatkan beberapa keterbatasan dari program aplikasi *audio player* tersebut yaitu sebagai berikut:

1. Berdasarkan pengujian fungsi yaitu
 - a. *Equalizer* dan *volume* yang berfungsi untuk merubah frekuensi suara file musik telah berfungsi secara baik ketika menjalankan prosedur (*Procedure*) dan fungsi (*Function*) : *Eq* dan *BASS_FX_DSPEQ*
2. Berdasarkan pengujian kesalahan interface yaitu
 - a. Aplikasi ini harus kembali ke menu open file dan mencari pada *directori* di komputer apabila ingin mengganti file lagu atau musik.

- b. Aplikasi ini belum dapat menampilkan daftar lagu dari nama penyanyi / grup band, judul lagu dan judul album.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah aplikasi audio player ini selesai dibangun dan diuji, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi audio player yang dibuat dan dikembangkan ini merupakan aplikasi audio player yang sederhana, sehingga sangat mudah dijalankan dan mudah dimengerti oleh user.
2. Aplikasi ini dapat menjalankan file-file audio seperti MP3, MIDI, WAV, Mpeg dan Wma.
3. Dalam keadaan file musik berjalan atau berhenti user dapat mengatur volume dan mengatur *equalizer* sesuai kebutuhan.

5.2 Saran

Aplikasi audio player ini yang telah dibangun dan di uji dirasakan masih memiliki beberapa kekurangan. Penulis menjadikan kekurangan tersebut sebagai

saran untuk pengembangan aplikasi selanjutnya di masa yang akan datang. Beberapa kekurangan tersebut adalah sebagai berikut:

1. Aplikasi hanya bisa menjalankan file audio, meskipun bisa menjalankan file video tetapi tidak terdapat menu untuk menampilkan gambar atau video klip musik dan disarankan untuk pengembangan selanjutnya aplikasi bisa menjalankan audio dan video lengkap dengan menampilkan gambar atau video klip.
2. Aplikasi ini disarankan agar bisa menampilkan daftar lagu dari nama penyanyi / grup band, judul lagu dan judul album.
3. Aplikasi ini disarankan bisa menyimpan file musik lebih dari satu sehingga bila ingin mengganti file user tidak perlu lagi mencari pada directori di komputer.

DAFTAR PUSTAKA

- Pressman, Roger S. 2002. *Rekayasa Perangkat Lunak Buku 1*. Yogyakarta: Penerbit Andi.
- Munawar. 2005. *Pemodelan Visual dengan UML*. Yogyakarta: Penerbit Graha Ilmu.
- Wahono, Romi Satria dan Sri Dharwiyanti. 2003. *Pengantar Unified Modeling Language (UML)*. www.ilmukomputer.com.
- Dewobroto, Wiryanto 2003. *Aplikasi Sains Dan Teknik Dengan Visual Basic 6.0*, PT. Elex Media Komputindo.
- MB, Rahimsyah 2003. *Kamus Komputer Dan Internet*, Jakarta: Penerbit Aprindo.
- Sutedjo, Budi dan Michael AN.2000. *Algoritma dan Teknik Pemrograman*, Yogyakarta: Penerbit Andi.
- Yuswanto, 2003. *Pemrograman Grafis dan Multimedia Visual Basic 6.0*, Surabaya : Prestasi Pustaka Publisher.
- [online], available: http://www.Planet_Sourcecode.com [2008, September 5]

LAMPIRAN LISTING PROGRAM

LISTING PROGRAM FORM UTAMA

```
Private Declare Function mciSendString _
Lib "winmm.dll" Alias "mciSendStringA" _
(ByVal lpstrCommand As String, _
ByVal lpstrReturnString As String, _
ByVal uReturnLength As Long, _
ByVal hwndCallback As Long) As Long

Private Sub Application_Click()
Form_Informasi.Show 1
End Sub

Private Sub Closed_Disk_Click()
Dim Y As Long
    Y = mciSendString _
("Set CDAudio Door Closed", _
vbNullString, 0, 0)
End Sub

Private Sub Form_Load()
Call Panggil
End Sub

Private Sub Form_Unload(Cancel As Integer)
MsgBox "Terima Kasih Atas Partisipasi'nya...! ", ,
"Flames'84 Player"
```

```
System_Player.DoStop
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Open_Disk_Click()
```

```
On Error Resume Next
```

```
    Dim X As Long
```

```
    X = mciSendString _
```

```
        ("Set CDAudio Door Open", _
```

```
        vbNullString, 0, 0)
```

```
End Sub
```

```
Private Sub Open_File_Click()
```

```
Dim Music
```

```
Dialog1.Filter = "All Video & Audio Files
```

```
(*.*)|*.avi;*.asf;*.mpeg;*.mpg;*.vmf;*.mlv;*.dat;*.wmv;*.wa
```

```
v;*.mp3;*.MID;*.wma;*.m3u;*.vob|Avi Files (*.avi)|*.avi|Asf
```

```
Files (*.asf)|*.asf|Mpeg Files (*.Mpeg)|*.Mpeg|Wmv
```

```
Files (*.wmv)|*.wmv|Wav Files (*.wav)|*.wav|Mp3
```

```
Files (*.mp3)|*.mp3|Midi Files (*.mid)|*.mid| Dat
```

```
Files (*.dat)|*.dat|Vmf Files (*.vmf)|*.vmf|Mlv
```

```
Files (*.mlv)|*.mlv|Mpg Files (*.mpg)|*.mpg|Wma
```

```
Files (*.Wma)|*.wma|M3u Files (*.m3u)|*.m3u|Vob
```

```
Files (*.vob)|*.vob"
```

```
Dialog1.Flags = cdIOFNAAllowMultiselect
```

```
Dialog1.ShowOpen
```

```
System_Player.FileName = Dialog1.FileName
```

```
Position_Music.Max = System_Player.MediaLenght
```

```
Timer1.Enabled = True
```

```
End Sub
```

```
Private Sub Play_Click()  
System_Player.MediaPlay  
End Sub
```

```
Private Sub Pause_Click()  
System_Player.MediaPause  
End Sub
```

```
Private Sub Stop_Click()  
System_Player.MediaStop  
End Sub
```

```
Private Sub Exit_Click()  
System_Player.DoStop  
Unload Me  
End Sub
```

```
Private Sub Position_Music_Scroll()  
System_Player.MePosSet = Position_Music.Value  
Position_Music.ToolTipText = "Music Position"  
End Sub
```

```
Private Sub Speaker_Balance_Scroll()  
System_Player.SetBalance (Speaker_Balance.Value)  
Speaker_Balance.ToolTipText = "Speaker Balance"  
End Sub
```

```
Private Sub Volume_Control_Scroll()  
Position_Music.VolumeSet (Volume_Control.Value)  
Volume_Control.ToolTipText = "Volume Control"  
End Sub
```

```
Private Sub Equalizer_Control_Scroll(Index As Integer)
System_Player.SetEq = Index
System_Player.ValEq = Equalizer_Control(Index).Value
End Sub
```

```
Private Sub IT_Programer_Click()
Form_IT_Programer.Show 1
End Sub
```

```
Private Sub Timer1_Timer()
Position_Music.Value = System_Player.MediaPos
End Sub
```

```
Sub Panggil()
Equalizer_Control(0).SelStart = 5
Equalizer_Control(1).SelStart = 5
Equalizer_Control(2).SelStart = 5
Equalizer_Control(3).SelStart = 5
Equalizer_Control(4).SelStart = 5
Equalizer_Control(5).SelStart = 5
Volume_Control.SelStart = 2
Speaker_Balance.SelStart = 0
End Sub
```

LISTING PROGRAM FORM IT PROGRAMMER

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode
As Integer)
Unload Me
```

```
End Sub
```

```
Private Sub Text_Thanks_To_Click()  
MsgBox "Maaf...! Anda Tidak Bisa Merubah'nya...! ",  
vbCritical, "Flames'84 Player"  
End Sub
```

```
Private Sub Text1_Click()  
MsgBox "Maaf...! Anda Tidak Bisa Merubah'nya...! ",  
vbCritical, "Flames'84 Player"  
End Sub
```

```
Private Sub Timer1_Timer()  
Text1 = Right(Text1, Len(Text1) - 1) + Left(Text1, 1)  
End Sub
```

LISTING PROGRAM FORM INFORMASI

```
Private Sub Text1_Click()  
MsgBox "Maaf...! Anda Tidak Bisa Merubah'nya...! ",  
vbCritical, "Flames'84 Player"  
End Sub
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode  
As Integer)  
Unload Me  
End Sub
```

LISTING PROGRAM BASS PADA EQUALIZER

```
Private Sub Bass()  
Dim f As Long  
Call BASS_ChannelGetAttributes(chan1, f, vbNull, vbNull)  
Call BASS_FX_DSP_Set(chan1, BASS_FX_DSPFXEQ)  
Call BASS_FX_DSP_Set(chan1, BASS_FX_DSPFXEQ)  
Call BASS_FX_DSP_Set(chan1, BASS_FX_DSPFXEQ)  
Call BASS_FX_DSP_Set(chan1, BASS_FX_DSPFXEQ)  
Call BASS_FX_DSP_Set(chan1, BASS_FX_DSPFXEQ)  
  
Dim eq As BASS_FX_DSPEQ  
eq.eqFreq = f  
eq.eqBandwidth = 2.5  
eq.eqGain = 0  
eq.eqBand = 0  
eq.eqCenter = 125  
Call BASS_FX_DSP_SetParameters(chan1, BASS_FX_DSPFXEQ, eq)  
eq.eqBand = 1  
eq.eqCenter = 1000  
Call BASS_FX_DSP_SetParameters(chan1, BASS_FX_DSPFXEQ, eq)  
eq.eqBand = 2  
eq.eqCenter = 8000  
Call BASS_FX_DSP_SetParameters(chan1, BASS_FX_DSPFXEQ, eq)  
  
Dim i As Integer  
For i = 0 To 5  
Next i
```

```
Dim eq As BASS_FX_DSPEQ
Call BASS_FX_DSP_GetParameters(chan1, BASS_FX_DSPFXEQ, eq)
Call BASS_FX_DSP_SetParameters(chan1, BASS_FX_DSPFXEQ, eq)
End Sub
```

DAFTAR RIWAYAT HIDUP

BIODATA:

Nama : Rika Novita
Tempat, Tanggal Lahir : Jakarta, 14 November 1984
Jenis Kelamin : Perempuan
Agama : Islam
Kewarganegaraan : Indonesia
Status : Belum Menikah
Alamat : Perum Binong Permai Blok L5 No.17 curug Tangerang
No.tlp (021) – 5983926 / 08128262629

PENDIDIKAN FORMAL:

SDN BINONG III Tangerang	Berijazah Tahun 1997
SLTP Negeri 1 Curug Tangerang	Berijazah Tahun 2000
SMU Nusantara 1 Tangerang	Berijazah Tahun 2003
Teknik Informatika Universitas Mercu Buana Meruya Selatan, Jakarta Barat	2003-2008

Demikian Daftar Riwayat Hidup ini saya buat dengan sebenar-benarnya untuk dipergunakan sebagaimana mestinya.