

# APLIKASI MP3 PLAYER BERDASARKAN POLLING SMS

Berda Setya A. \*, Agung Budi P. \*\*, Aghus Sofwan\*\*

**Abstrak** - Perangkat lunak yang dapat memainkan file audio dan video, yang biasa disebut media player, sudah sangat dikenal oleh para pengguna komputer. Salah satu format file audio yang sangat populer untuk dimainkan menggunakan media player adalah mp3. Ukuran file mp3 yang relatif kecil dibandingkan format audio lainnya menjadi faktor yang sangat menunjang tersebar luasnya tipe file ini. Perangkat lunak ini memiliki berbagai fitur yang ditawarkan oleh para pengembangnya, mulai dari skin (tampilan), format file yang didukung, fasilitas konversi format file, dan lain-lain.

Aplikasi Mp3 Player Berdasarkan Polling SMS ini merupakan pengembangan perangkat lunak media player untuk memainkan file lagu dengan format mp3 atau Mp3 Player. Aplikasi ini dapat memainkan file lagu dengan format mp3 berdasarkan hasil polling SMS yang masuk ke suatu ponsel penerima secara otomatis. Aplikasi ini dirancang menggunakan suatu bahasa pemodelan perancangan perangkat lunak yaitu UML (Unified Modeling Language) dan diimplementasikan menggunakan bahasa pemrograman Java 2 SDK, SE 1.4.2. Aplikasi ini terdiri dari tiga program, yaitu program untuk menampilkan ikon system tray pada taskbar Windows, program untuk mengatur waktu aktivasi program utama, dan program untuk memainkan lagu berdasarkan polling SMS yang merupakan program utama pada aplikasi ini.

## I. PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan perangkat lunak yang dapat memainkan file-file audio dan video, atau media player, sudah sangat pesat. Fitur-fitur yang ditawarkan oleh player ini pun semakin beragam, mulai dari *skin* (tampilan) yang makin atraktif hingga kemampuan mencari file-file audio maupun video dengan format tertentu dalam lingkungan internet.

Salah satu format file audio yang sangat populer dan didukung untuk dimainkan menggunakan media player adalah mp3. Tipe file ini sangat dikenal masyarakat karena ukurannya yang relatif kecil dibandingkan format file lainnya dengan durasi yang sama, sehingga menjadikan tipe file ini lebih praktis dan efisien untuk dimainkan, ditransfer maupun disimpan.

Sama halnya dengan perkembangan perangkat lunak media player, SMS (*Short Messaging Service*) yang merupakan salah satu fitur sistem seluler GSM (*Global System for Mobile Communication*), dimana dikembangkan dan distandarisi oleh ETSI (*European Telecommunication Standard Institute*), juga sebagai fasilitas yang banyak digunakan masyarakat sekarang karena SMS memiliki tarif yang sangat murah dibandingkan jika melakukan percakapan secara langsung dengan nomor yang dituju.

### 1.2 Tujuan Tugas Akhir

Tujuan yang ingin dicapai pada tugas akhir ini adalah merancang dan membangun suatu aplikasi Mp3 Player yang dapat memainkan lagu berdasarkan jumlah pemilih terbanyak melalui polling SMS secara otomatis.

### 1.3 Pembatasan Masalah

Pembatasan masalah dalam tugas akhir ini meliputi:

1. Aplikasi yang dibangun terdiri dari tiga program, yaitu program utama Mp3 Player Berdasarkan Polling SMS, program untuk menentukan waktu aktivasi program utama, dan program untuk menampilkan ikon pemanggil program waktu aktivasi program utama pada taskbar Windows.
2. Aplikasi ini diuji menggunakan Sistem Operasi Windows XP Professional.
3. Bahasa pemrograman yang digunakan adalah Java 2 SDK, SE v1.4.2.
4. Jumlah maksimal dari lagu yang dipolling adalah 15 lagu.
5. Port yang digunakan dalam pengujian untuk koneksi antara komputer dan ponsel penerima adalah port serial COM1 dan dihubungkan menggunakan kabel data RS232.
6. Ponsel yang digunakan sebagai penerima SMS dalam pengujian adalah Siemens C55.
7. Format file dari lagu yang dipolling dan dimainkan adalah \*.mp3.

## II. TINJAUAN UMUM CARA KERJA SMS, AT COMMAND, JAVA MEDIA FRAMEWORK, DAN JAVA COMMUNICATIONS API

### 2.1 Cara Kerja SMS

SMS dikirim dari pengirim ke penerima melewati SMSC dengan prinsip *Store and Forward*, dimana pesan yang dikirim ke SMSC akan disimpan terlebih dahulu hingga masa validitas tertentu terpenuhi jika ponsel nomor yang dituju dalam keadaan mati ataupun diluar jangkauan operator, setelah ponsel nomor yang dituju sudah aktif atau berada dalam jangkauan operator maka pesan akan diteruskan oleh SMSC kepada penerima. Apabila pesan yang tersimpan di SMSC sudah melewati masa validitas yang ditentukan, pesan tersebut akan dihapus dan tidak akan diteruskan kepada nomor yang dituju.



Gambar 2.1 Skema Pengiriman SMS.

\* Mahasiswa Jurusan Teknik Elektro UNDIP

\*\* Staf Pengajar Jurusan Teknik Elektro UNDIP

Dengan adanya SMSC, pengirim SMS juga dapat mengetahui status dari pesan yang dikirim, apakah pesan tersebut sudah diterima atau belum.

Pada pengiriman dan penerimaan SMS, ada 2 mode format SMS yang digunakan oleh operator maupun terminal. Mode yang pertama adalah mode PDU (*Protocol Data Unit*), dimana format pesan dalam bentuk oktet heksadesimal dan oktet semidesimal dengan panjang mencapai 160 (7 bit) atau 140 (8 bit) karakter. Sedangkan mode yang kedua adalah mode teks, dimana pesan dalam bentuk teks asli. Akan tetapi, tidak semua operator GSM ataupun terminal di Indonesia mendukung format pesan mode teks. Oleh karena itu, format pesan yang digunakan dalam penyusunan tugas akhir ini adalah mode PDU.

Dalam proses pengiriman pesan, dikenal dua jenis *mobile*, yaitu Ponsel Pengirim (*Mobile Originated*) dan Ponsel Penerima (*Mobile Terminated*). Kedua *mobile* tersebut memiliki metode yang berbeda dalam pengolahan pesan. Pada Ponsel pengirim, metode yang digunakan adalah *encodec*, sedangkan pada Ponsel penerima, metode yang digunakan adalah *decodec*. Pada metode *encodec*, pesan dalam bentuk teks akan diubah dalam format PDU. Hal itu disebabkan SMSC hanya dapat menerima pesan dalam format PDU sehingga pesan yang akan dikirim harus diubah terlebih dahulu kedalam format PDU dan proses pengubahan ini berlangsung pada terminal atau ponsel. Sedangkan metode *decodec* adalah kebalikan dari *encodec*, dimana pesan diterima dan SMSC dalam format PDU kemudian diubah menjadi format teks.

Selain metode pengolahan pesan yang berbeda, Ponsel Pengirim dan Ponsel Penerima juga memiliki skema format SMS PDU yang berbeda, dimana skema ini sudah diatur dan distandarisasi oleh ETSI. Pada makalah ini hanya akan dibahas mengenai SMS PDU Penerima karena jenis SMS inilah yang digunakan pada aplikasi yang dibangun. SMS PDU Penerima adalah pesan yang dikirim dari SMSC ke ponsel tujuan dalam format PDU. Pada aplikasi yang dibangun pada tugas akhir ini, SMS PDU Penerima inilah yang akan digunakan, untuk kemudian diubah kedalam bentuk teks agar dapat dibaca. Skema dari format SMS PDU Penerima adalah:

SCA	PDU Type	OA	PID	DCS	SCTS	UDL	UD
-----	----------	----	-----	-----	------	-----	----

Gambar 2.2 Skema Format SMS PDU Penerima.

## 2.2 AT Command

*AT Command* adalah perintah-perintah standar yang digunakan untuk melakukan komunikasi antara komputer dengan ponsel melalui serial port. Melalui *AT Command*, data-data yang ada di dalam ponsel dapat diketahui, mulai dari vendor ponsel, kekuatan sinyal, membaca pesan, mengirim pesan, dan lain-lain. Tabel 2.7 menunjukkan beberapa perintah *AT Command* yang sering digunakan.

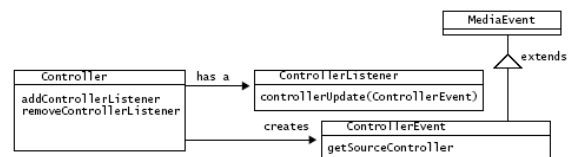
Tabel 2.1 Beberapa Perintah *AT Command*

AT Command	Keterangan
AT	Mengecek apakah ponsel telah terhubung
AT + CMGF	Menetapkan format mode dari terminal
AT + CSCS	Menetapkan jenis <i>encoding</i>
AT + CNMI	Mendeteksi pesan SMS baru masuk secara otomatis
AT + CMGL	Membuka daftar SMS yang ada pada kartu SIM
AT + CMGS	Mengirim pesan SMS
AT + CMGR	Membaca pesan SMS
ST + CMGD	Menghapus pesan SMS

## 2.3 Java Media Framework

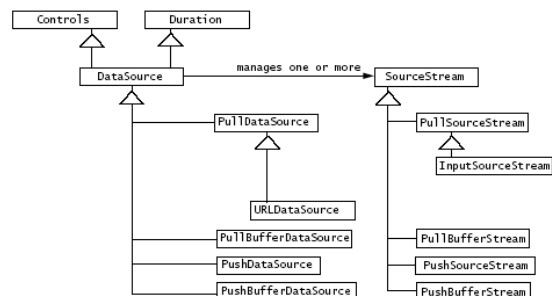
*Java Media Framework* (JMF) adalah suatu antarmuka pemrograman aplikasi (API) untuk menggabungkan data media seperti audio dan video kedalam aplikasi dan applet Java. JMF menyediakan arsitektur yang digabungkan dan protokol pesan untuk mengatur akuisisi, pemrosesan, dan pengiriman data media berbasis waktu. JMF dirancang untuk mendukung tipe-tipe media yang paling standar, seperti AIFF, AU, AVI, GSM, MIDI, MPEG, *QuickTime*, RMF, dan WAV.

JMF menggunakan even terstruktur yang melaporkan mekanisme untuk menjadikan program berbasis JMF selalu memiliki informasi mengenai keadaan terbaru dari sistem media dan memungkinkan program berbasis JMF merespon berbagai kondisi kesalahan pada pengendalian media, seperti kondisi tidak tersedianya *resource* dan data. Tiap kali objek JMF melaporkan kondisi terbaru, objek tersebut membutuhkan *MediaEvent*. *MediaEvent* adalah suatu sub-kelas untuk mengidentifikasi berbagai tipe even.



Gambar 2.3 Model even JMF.

*Media player* JMF biasanya menggunakan *DataSources* untuk mengatur transfer isi media. Suatu *DataSource* mengenkapsulasi lokasi media dan protokolnya, serta perangkat lunak yang digunakan untuk mengirim media. Suatu *DataSource* diidentifikasi oleh JMF *MediaLocator* ataupun URL (*Universal Resource Locator*). Model data pada JMF dapat dilihat pada Gambar 2.4 di bawah ini.



Gambar 2.4 Model data JMF.

## 2.4 Java Communications API

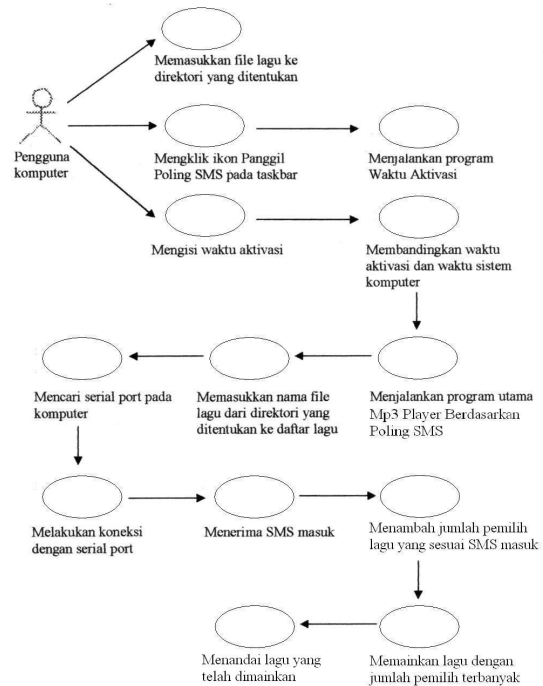
Java Communications API digunakan untuk membuat suatu aplikasi komunikasi bersifat *platform-independent* untuk berbagai teknologi seperti pesan suara, *fax*, dan *smartcard*. Java Communications API versi 2.0 mendukung komunikasi dengan serial port RS232 dan paralel port IEEE 1284.

Paket `javax.comm` pada Java Communications API memiliki kemampuan sebagai berikut:

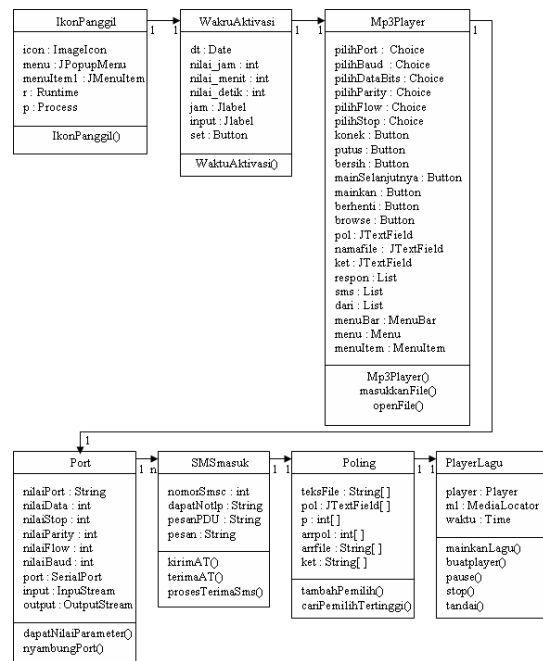
- Menemukan satu per satu port-port yang terdapat pada sistem. `Method` statis `CommPortIdentifier`. `getPortIdentifier` mengembalikan objek untuk menemukan port satu per satu, yang berisi objek `CommPortIdentifier` untuk tiap port yang ada. Objek `CommPortIdentifier` ini merupakan mekanisme utama untuk mengendalikan akses ke sebuah port komunikasi.
- Membuka dan meminta kepemilikan port-port komunikasi dengan menggunakan `method-method high-level` dalam objek `CommPortIdentifier` nya.
- Memecahkan pertentangan kepemilikan port antar aplikasi Java. Even-even dipicu untuk menandai aplikasi-aplikasi yang berkaitan dengan pertentangan kepemilikan dan memungkinkan pemilik port untuk melepaskan kepemilikan. `PortInUseException` dilempar ketika suatu aplikasi gagal membuka sebuah port.
- Menampilkan I/O sinkron pada port-port komunikasi. Kelas-kelas *low-level* seperti `SerialPort` dan `ParallelPort` memiliki `method-method` untuk mengatur I/O pada port-port komunikasi.
- Menerima even-even yang menggambarkan perubahan keadaan port komunikasi. Contohnya, saat serial port memiliki perubahan keadaan untuk *Carrier Detect*, *Ring Indicator*, *DTR*, dan lain-lain. Objek `SerialPort` memicu `SerialPortEvent` yang menceritakan perubahan keadaan.

## III. PERANCANGAN PERANGKAT LUNAK

Perancangan perangkat lunak pada Tugas Akhir ini menggunakan salah satu bahasa pemodelan yaitu UML (*Unified Modeling Language*).



Gambar 3.1 Diagram use-case perancangan perangkat lunak.



Gambar 3.2 Diagram kelas perancangan perangkat lunak.

## IV. PENGUJIAN DAN ANALISIS

Pengujian aplikasi Mp3 Player Berdasarkan Polling SMS ini dilakukan dengan metode pengujian *white box* dan *black box* yang meliputi pengujian *loop*, pengujian pernyataan kondisional, dan pengujian output sistem.

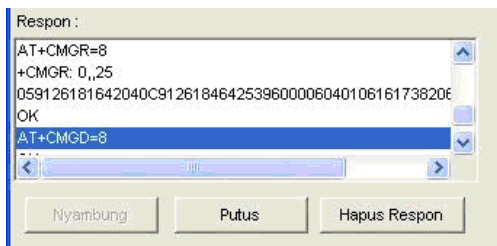
### 4.1 Perangkat Keras

Perangkat keras yang digunakan untuk pengujian aplikasi yaitu:

- 1 (satu) set perangkat komputer dengan spesifikasi:
  - prosesor Intel Pentium IV/1,7GHz
  - RAM 256 MB
  - Sistem operasi Windows XP Professional
  - Soundcard PCI Sound Vision
- 1 (satu) ponsel Siemens C55, sebagai ponsel penerima
- 1 (satu) kabel data RS232 sebagai penghubung antara ponsel penerima dan port serial pada komputer.

#### 4.2 Kelas SMSmasuk

Seperti tampak pada Gambar 4.1 dibawah ini, saat ada pesan yang masuk ke ponsel penerima, kelas SMSmasuk langsung melakukan pembacaan terhadap pesan tersebut dan ditampilkan pada *list* Respon. Pesan yang ditampilkan ini tentunya masih dalam format PDU karena belum dilakukan perubahan ke bentuk teks.

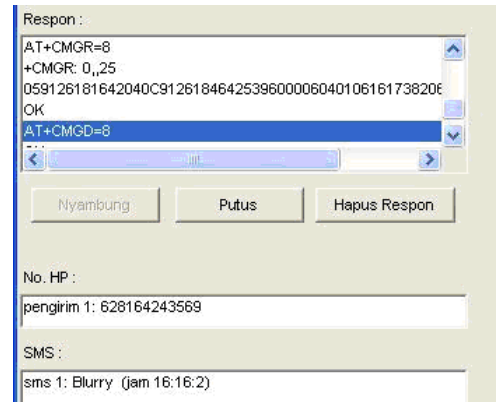


Gambar 4.1 Membaca pesan masuk.

Seperti halnya aktivitas mendengarkan pesan masuk pada kelas Port, untuk membaca pesan masuk juga dibutuhkan *AT Command*. Dalam hal ini, *AT Command* yang digunakan adalah AT+CMGR.

Setelah ketiga aktivitas di atas, kelas SMSmasuk akan menghapus pesan yang telah dibaca pada ponsel seperti diagram aktivitas pada tahap perancangan. Untuk menghapus pesan, juga diperlukan salah satu *AT Command*, yaitu AT+CMGD.

Aktivitas yang terakhir dari kelas ini adalah menampilkan nomor ponsel pengirim dan pesan yang telah dibaca pada monitor. Hal ini dapat dilihat pada Gambar 4.2.



Gambar 4.2 Menampilkan nomor pengirim dan pesan yang dibaca.

Dari gambar di atas, tampak bahwa nomor pengirim ditampilkan pada list No.HP, sedangkan pesan yang dibaca ditampilkan pada list SMS.

#### 4.3 Kelas Poling

Pada Gambar 4.3 tampak aktivitas kelas Poling, yaitu menambah jumlah pemilih untuk nama file lagu yang sesuai pesan masuk.



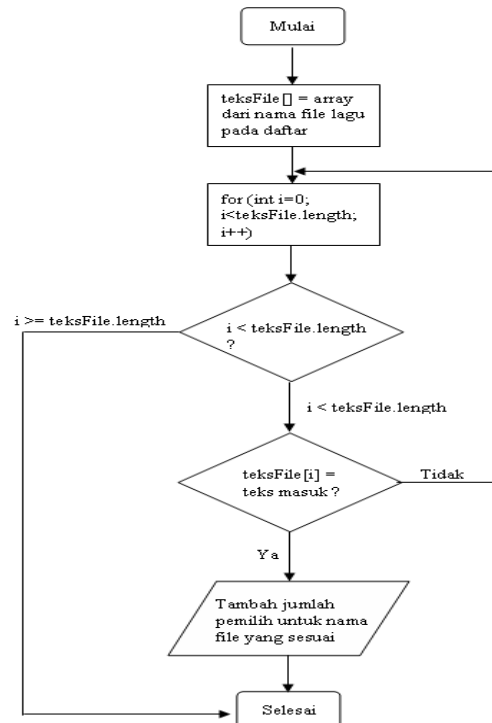
Gambar 4.3 Menambah jumlah pemilih.

Hasil pengujian kondisional (hasil pengujian dengan metode *white box*) dan output yang diharapkan (hasil pengujian dengan metode *black box*) pada aktivitas menambah jumlah pemilih ini dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil pengujian kondisional dari kelas Poling.

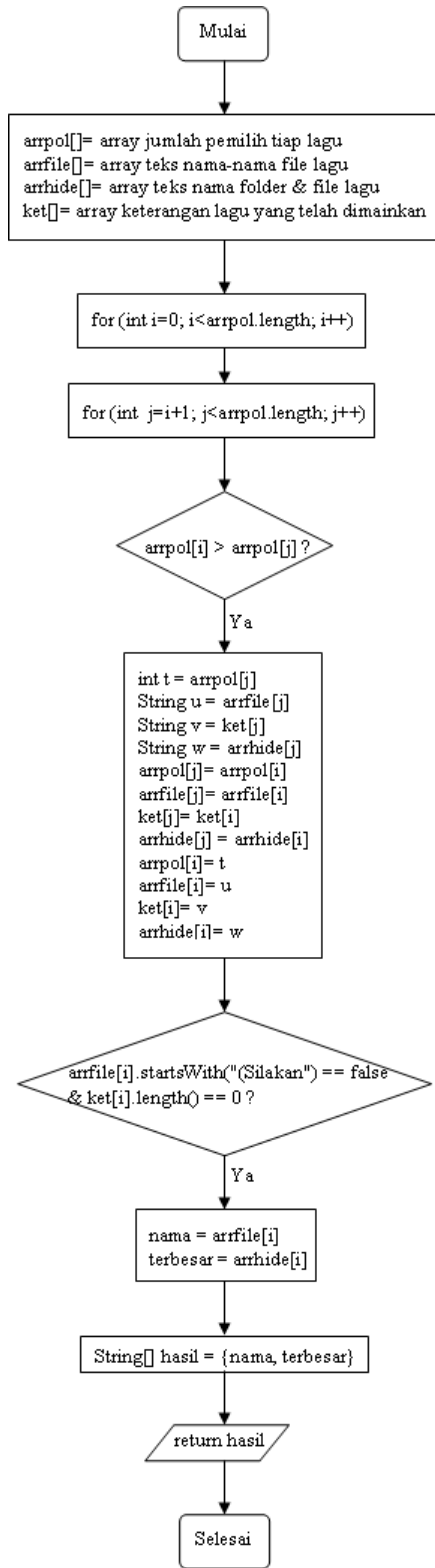
Kondisi	Output hasil pengujian	Output yang diharapkan
Teks pesan masuk tidak sesuai dengan nama file lagu	Tidak menambah jumlah pemilih	Tidak menambah jumlah pemilih
Teks pesan masuk sesuai dengan nama file lagu	Menambah jumlah pemilih untuk file lagu yang sesuai	Menambah jumlah pemilih untuk file lagu yang sesuai

Sedangkan diagram alir proses looping untuk menambah jumlah pemilih jika teks pesan masuk sesuai dengan nama file lagu pada daftar dapat dilihat dalam Gambar 4.4 berikut:



Gambar 4.4 Diagram alir proses *looping* untuk menambah jumlah pemilih.

Sedangkan diagram alir proses *looping* untuk mencari nama file lagu dengan jumlah pemilih tertinggi pada daftar lagu yang dipolling dapat dilihat dalam Gambar 4.5 di bawah ini:



Gambar 4.5 Diagram alir proses *looping* untuk mencari lagu dengan jumlah pemilih tertinggi.

#### 4.4 Kelas PlayerLagu

Aktivitas dari kelas PlayerLagu ini adalah memainkan lagu dengan jumlah pemilih tertinggi seperti tampak pada Gambar 4.6 dibawah ini:



Gambar 4.6 Memainkan lagu dengan jumlah pemilih tertinggi.

Setelah lagu dengan jumlah pemilih tertinggi dimainkan, aktivitas selanjutnya yaitu menandai lagu yang telah dimainkan sebagaimana dapat dilihat pada Gambar 4.7.



Gambar 4.7 Menandai lagu yang telah dimainkan.

## V. PENUTUP

### 5.1 Kesimpulan

1. Berdasarkan hasil pengujian dengan metode *white box*, kode program dalam aplikasi Mp3 Player Berdasarkan Polling SMS yang telah dibangun telah cukup efektif, baik dari segi pernyataan kondisional maupun *looping* yang terdapat di dalamnya.
2. Berdasarkan hasil pengujian dengan metode *black box*, aplikasi Mp3 Player Berdasarkan Polling SMS yang dibangun telah sesuai dengan output yang diharapkan.

### 5.2 Saran

1. Aplikasi Mp3 Player Berdasarkan Polling SMS dapat lebih dikembangkan lagi dengan membuat aplikasi ini dapat melakukan polling dengan jumlah lagu yang tidak terbatas, untuk itu dibutuhkan suatu metode pembacaan daftar file dalam suatu direktori yang dilakukan setiap kali selesai memainkan sebuah file lagu
2. Diharapkan aplikasi ini dapat lebih dikembangkan agar tidak hanya dapat memainkan file audio dengan format \*.mp3 saja, akan tetapi juga berbagai file audio dan video. Hal ini dapat dilakukan dengan lebih memaksimalkan pemanfaatan kelas Controller maupun kelas-kelas lain yang mendukung dalam *Java Media Framework*.

## DAFTAR PUSTAKA

1. Govindarajan, S., *Run Java Apps from SysTray*, <http://www.pcquest.com/content/Developer/2005/105041202.asp>, April 2005.
2. Gunadi, H. dan Suhendar, A., *Visual Modeling Menggunakan UML dan Rational Rose*, Informatika, Bandung, 2002.

3. Komputer, W., *Membuat Aplikasi Profesional dengan Java*, PT. Elex Media Komputindo, Jakarta, 2005.
4. Komputer, W., *Pengembangan Aplikasi Sistem Informasi Akademik Berbasis SMS dengan Java*, Salemba Infotek, Jakarta, 2005.
5. Naughton, P., *JAVA HANDBOOK Konsep Dasar Pemrograman Java*, Penerbit Andi, Yogyakarta, 2000.
6. Prasetyo, D. D., *Tip dan Trik Pemrograman Java 2*, PT. Elex Media Komputindo, Jakarta, 2004.
7. ---, <http://java.sun.com/products/java-media/jmf/2.1.1/documentation.html>.
8. ---, <http://java.sun.com/products/java-media/jmf/2.1.1/download.html>.
9. ---, [http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML\\_tutorial/index.htm](http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/index.htm), 2001.
10. ---, <https://jdic.dev.java.net/servlets/ProjectDocumentList?folderID=2739&expandFolder=2739&folderID=0>, Februari 2005.

**Berda Setya A.**  
**L2F 000 586**



Lahir pada tanggal 17 Juli 1982 di Semarang, Menempuh pendidikan di SD Kristen Gergaji Semarang lulus tahun 1994, kemudian melanjutkan ke SMP Negeri 3 Semarang lulus tahun 1997. Lalu melanjutkan ke SMU Negeri 1 Semarang lulus tahun 2000, dan sekarang sedang menyelesaikan studi S1 pada Jurusan Teknik Elektro konsentrasi Informatika dan Komputer, Fakultas Teknik, Universitas Diponegoro Semarang.

Menyetujui dan mengesahkan

Pembimbing I

Agung Budi Prasetijo, ST, MIT  
NIP. 132 137 932

Pembimbing II

Aghus Sofwan, ST, MT  
NIP.132 163 757