

PEMANFAATAN SEBUAH APLIKASI GITAR PLAYER YANG MENGGUNAKAN TEKNOLOGI JAVA MOBILE (J2ME) PADA HANDPHONE DAN SEJENISNYA

Oleh
Adina Sastra Sembiring, S.Pd., M.Pd

Abstrak

Teknologi mobile merupakan teknologi yang sangat populer pada saat ini. Saat ini banyak aplikasi mobile yang beredar. Melihat perkembangan aplikasi mobile yang semakin pesat maka proyek akhir ini mengangkat judul tentang Aplikasi Gitar Player Menggunakan Teknologi Java Mobile Pada Handphone yaitu bagaimana membuat sebuah aplikasi untuk memainkan alat musik gitar di perangkat mobile. Terdapat 4 mode dalam aplikasi ini, yaitu Single Chromatic dan Single Diatonic yaitu membuntikan nada/not serta Full Chromatic dan Full Diatonic yaitu cara membunyikan akor-akor atau kunci-kunci nada, serta terdapat fungsi record dan playback. Single Chromatic merupakan salah satu menu yang memainkan lagu per-nada/not dalam tangga nada kromatik, sedangkan Single Diatonic merupakan salah satu menu yang memainkan lagu per-nada/not dalam tangga nada diatonik. Full Chord Chromatic memainkan akor-akor kromatik yang ada pada lagu. Full Chord Diatonic memainkan akor-akor diatonik yang ada pada lagu. Cara kerja dari aplikasi ini adalah dengan cara mengambil/me-load file audio (.mid) dan file visual (*.png) pada direktori masing-masing. Pada Single Chord dan Full Chord terdapat fasilitas playback (memainkan kembali nada-nada yang telah disusun), save (menyimpan nada/not yang telah disusun dan dapat dimainkan kembali pada menu Load Nada/Chord. Pada menu Full Chord Chromatic terdapat fasilitas options untuk memberikan pilihan jenis akor yang dimainkan (Chord Major/Minor).*

Kata Kunci : Teknologi java mobile, tinjauan umum tentang nada, perencanaan sistem, uji coba dan analisa.

A. PENDAHULUAN

Teknologi *mobile* merupakan teknologi baru yang sangat praktis. Teknologi ini mampu memberikan kemudahan bagi para pengguna perangkat *mobile*. Kemudahan inilah yang membuat teknologi *mobile* semakin banyak diminati, sehingga perkembangannya semakin pesat. Perkembangan teknologi *mobile* yang semakin pesat dari tahun ke tahun merupakan alasan bagi para vendor untuk terus bersaing menciptakan inovasi-inovasi

*) Adina Sastra Sembiring, S.Pd., M.Pd. adalah Dosen Pendidikan Seni Musik Fakultas Bahasa dan Seni Universitas Negeri Medan yang telah mengampu mata kuliah Gitar di Program Studi Pendidikan Seni Musik FBS Universitas Negeri Medan.

perangkat komunikasi dalam aktivitas mobile yang semakin canggih dan didukung oleh fitur-fitur tambahan yang memenuhi kebutuhan penggunaannya. Fitur-fitur yang telah ada pada perangkat *mobile* diantaranya seperti radio, mp3, 3G, *video call*, *camera*, GPRS, *java* versi tertentu yang mampu menjalankan berbagai aplikasi *java* seperti *game*, *edit foto*, kamus, kalkulator, memainkan musik dan lain-lain.

Perkembangan yang pesat dalam dunia teknologi, khususnya teknologi *mobile* tersebut juga dialami oleh dunia musik. Dunia musik zaman sekarang tidak kalah canggih dari dunia teknologi. Saat ini belajar musik tidak harus selalu mengikuti kursus-kursus musik yang ada, melainkan bisa belajar melalui media seperti komputer, bahkan bisa langsung *live* melalui *video streaming*. Saat ini para vendor

aplikasi *mobile* pun mulai banyak mengembangkan aplikasi musik berbasis *mobile*. Kebanyakan aplikasi-aplikasi musik *mobile* saat ini belum sempurna dikarenakan keterbatasan yang dimiliki oleh perangkat-perangkat *mobile*. Melihat perkembangan aplikasi *mobile* tersebut maka tulisan ini mengangkat judul tentang Aplikasi Gitar Player dengan *Java Mobile* yaitu bagaimana membuat sebuah aplikasi untuk memainkan alat musik gitar di perangkat *mobile*. Aplikasi ini merupakan aplikasi pembelajaran gitar yang diperuntukkan bagi para pemula yang ingin belajar gitar dengan praktis. Aplikasi ini dibuat mengingat pembelajaran musik dalam bentuk aplikasi *mobile* sangat jarang ditemukan, terutama untuk pembelajaran gitar. Selama ini pembelajaran gitar selalu diharuskan untuk membawa gitar, tapi dengan adanya aplikasi ini diharapkan bisa belajar gitar dengan praktis, terutama pembelajaran mengenai *feeling* suara gitar.

1. J2ME

Java2 Micro Edition atau biasa yang disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak *Java* pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya. J2ME membawa *Java* ke dunia informasi, komunikasi dan perangkat komputasi selain perangkat komputer dekstop yang biasanya lebih kecil dibandingkan perangkat komputer dekstop. J2ME biasa digunakan pada telepon seluler, pager, *personal digital assistants* (PDA) dan sejenisnya.

2. CLDC

CLDC atau *Connected Limited Device Configuration* adalah perangkat dasar dari J2ME, spesifikasi dasar yang berupa *library* dan API yang diimplementasikan pada J2ME,

seperti yang digunakan pada telepon selular, *pager* dan PDA. Perangkat tersebut dibatasi dengan keterbatasan memori, sumber daya dan kemampuan memproses. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal dari *package*, kelas dan sebagian fungsi dari *Java Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat-alat tersebut, JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*).

3. MIDP

MIDP atau *Mobile Information Device Profile* adalah spesifikasi untuk sebuah profil J2ME. MIDP memiliki lapisan di atas CLDC, API tambahan untuk daur hidup aplikasi, antarmuka, jaringan dan penyimpanan resisten. Pada saat ini terdapat MIDP 1.0 dan MIDP 2.0. Fitur tambahan MIDP 2.0 dibanding MIDP 1.0 adalah API untuk multimedia. Pada MIDP 2.0 terdapat dukungan memainkan *tone*, *tone sequence* dan *file WAV* walaupun tanpa adanya *Mobile Media API* (MMAPI).

4. MIDlet

MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari kelas *javax.microedition.midlet*. MIDlet yang didefinisikan pada MIDP. MIDlet berupa sebuah kelas abstrak yang merupakan subkelas dari bentuk dasar aplikasi sehingga antarmuka antara aplikasi J2ME dan aplikasi manajemen pada perangkat dapat terbentuk.

5. Emulator Ponsel Java

Untuk menjalankan MIDlet, programmer tidak perlu memiliki dan mencobanya pada ponsel. Cukup dengan emulator dari ponsel yang dapat berjalan pada PC. Dalam hal ini, emulator yang digunakan adalah *Sony Ericsson SDK 2.5.0.6 for the Java™ ME Platform*.

B. PEMBAHASAN

1. TEKNOLOGI JAVA MOBILE

1.1. LIGHT WEIGHT USER INTERFACE TOOLKIT (LWUIT)

Lightweight User Interface Toolkit (LWUIT) adalah adalah lightweight widget library dari SUN Microsystem yang terinspirasi oleh *swing* namun di desain untuk device dengan kapasitas terbatas seperti mobile phone dan PDA. LWUIT mendukung kemampuan pluggable theme, hirarki komponen dan kontainer dan abstraksi dibawah GUI toolkit.

Interface internal dan kelas abstrak menyediakan *abstraction* dari interface dan API pada profile sehingga memudahkan portability dan migrasi untuk device dan profile di masa mendatang. LWUIT menghindari *lowest common denominator* yakni mengimplementasikan beberapa fitur pada platform *low-end* dan membawa hasil yang lebih baik pada platform *high-end*.

1.2. MMAPI

Mobile Media API (MMAPI) memfasilitasi pembuatan nada, memainkan serta merekam audio dan video pada *device* yang cocok. Memainkan atau merekam sebuah media ditangani oleh dua objek : *DataSource* dan *Player*. *DataSource* menangani detail cara mendapatkan data dari *source* yang tersedia. Source dapat berasal dari file JAR atau jaringan (melalui protokol HTTP), *record* dari RMS, *streaming connection* dari sebuah server atau sumber proprietary lain. Player tidak perlu terlalu mempermasalahkan darimana data berasal atau bagaimana cara mendapatkannya. Player hanya perlu membaca data yang berasal dari *DataSource*, memproses, menampilkan dan memainkan *playback* media pada output *device*.

1.3. Record Manajemen System (RMS)

MIDlet tidak menggunakan file sistem untuk menyimpan data, tetapi

menyimpan semua informasi dalam sebuah memori *non-volatile* (memori tetap) yang disebut dengan *Record Management System* (RMS). RMS (Record Management Sistem) adalah kumpulan *record*, dan *record* disimpan sebagai *array* dari *byte* dalam sebuah *record store*. RMS memiliki orientasi *record* basis data yang sederhana sehingga MIDlet dapat menyimpan informasi dan mengaksesnya.

2. TINJAUAN UMUM TENTANG NADA

Tangga nada dibagi menjadi 2 bagian yaitu, tangga nada kromatik dan tangga nada diatonik. Pada tangga nada kromatik, setiap nada berjarak $\frac{1}{2}$ atau 1 *semitone*. Berikut ini urutan nada kromatik pada nada dasar C :

Not: 1 1/2 2 2 3/2
 4 4 5 5 6 6 7
 Baca: do do# re re# mi
 fa fa# sol sol# la la# si
 Kunci: C C# D D# E F
 F# G G# A A# B

Sedangkan pada tangga nada diatonik, jarak antar nada atau biasa disebut interval nada adalah 1-1- $\frac{1}{2}$ -1-1-1 atau jika dalam *semitone* intervalnya menjadi 2-2-1-2-2-2. Berikut ini urutan nada diatonik pada nada dasar C :

Not: 1 2 3 4 5 6
 7
 Baca: do re mi fa sol la
 si
 Kunci: C D E F G A B

3. PERENCANAAN SISTEM

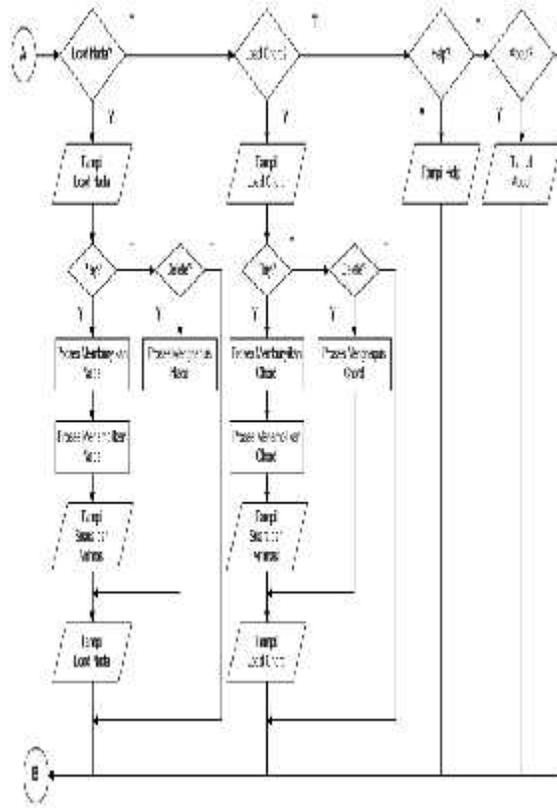
3.1. RANCANGAN APLIKASI

Dalam perancangan dan pembuatan rancangan game terdapat beberapa elemen pembangun utama, antara lain :

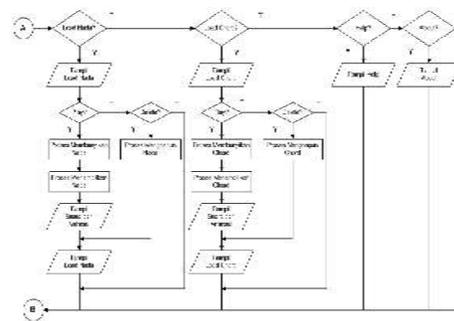
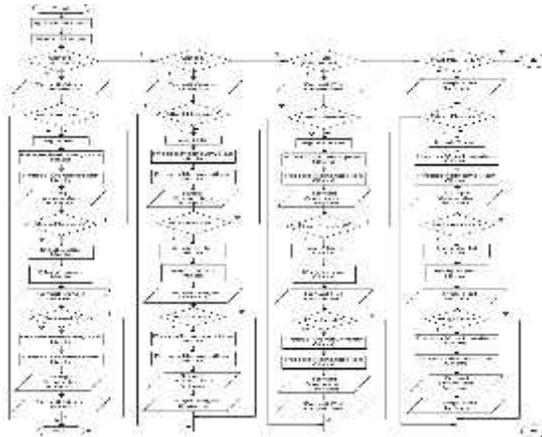
- a) Menu *Single Chromatic*, memainkan nada-nada chromatic yang ada pada gitar.
- b) Menu *Full Chromatic*, memainkan chord-chord chromatic yang ada pada gitar.
- c) Menu *Single Diatonic*, memainkan nada-nada sesuai tangga nada diatonis yang ada pada gitar.
- d) Menu *Full Diatonic*, memainkan chord-chord sesuai tangga nada diatonis yang ada pada gitar.
- e) Menu *Load Nada*, memainkan kembali nada-nada yang telah disimpan.
- f) Menu *Load Chord*, memainkan kembali chord-chord yang telah disimpan.
- g) Menu *Help*, berisi tentang bagaimana cara memainkan aplikasi ini.
- h) Menu *About*, berisi tentang pembuat aplikasi *Gitar Player Mobile* ini.

3.1.1. Flowchart Sistem

Pada bagian ini, akan dijelaskan tentang *flowchart* atau diagram alir dari program. *Flowchart* ini menggambarkan bagaimana alur dari program. Diantaranya, apa yang mula-mula dilakukan, apa yang akan dilakukan pada kondisi ini terpenuhi, atau apa yang akan dilakukan apabila kondisi ini tidak terpenuhi.



Gambar 3.1.1.a. *Flowchart* Sistem I



Gambar 3.1.1.b. *Flowchart* Sistem II

3.2. RANCANGAN DATA

Berikut merupakan data-data yang dibutuhkan dalam pembuatan aplikasi ini :

3.2.1. Data Gambar

Data gambar berupa *file* data yang berformat *"*.png"* yang digunakan sebagai gambar latar belakang aplikasi, *splashscreen*, *selector* menu. Khususnya untuk animasi gitar, gambar-gambar yang dibutuhkan adalah gambar-gambar chord gitar yang kemudian digabungkan menjadi satu *file* seperti *file* A#.png, B.png, C.png, dll. Dalam 1 *file* yang terdiri dari 1 nada (single chromatic/diatonic) atau chord (full chromatic/diatonic) ini memiliki posisi fret yang berbeda. Secara keseluruhan dalam aplikasi ini menggunakan data gambar gitar sebanyak 61 gambar gitar, yang terdiri dari 56 gambar gitar untuk Menu Single Chromatic dan Diatonic, serta 5 gambar untuk Menu Full Chromatic dan Full Diatonic. Data gambar gitar ini diambil dari Software Guitar Pro v5.2. Pada Menu Single Chromatic, Full Chromatic, Single Diatonic, dan Full Diatonic mula-mula ditampilkan gambar default dari tampilan gitar tanpa dipetik. Gambar/Visualisasi akan berubah apabila user memainkan lagu.

Gambar tersebut bisa berubah-ubah bergantung pada inputan nada/chord oleh user. Perubahan gambar juga terjadi pada saat user melakukan *Playback* dari nada/chord yang dimainkan. Pada saat *playback*, system akan melakukan pergantian visualisasi berdasarkan data nada/chord yang dimainkan user pada saat melakukan proses *recording*. Begitupula pada Menu Load Nada dan Menu Load Chord, pada kedua menu ini konsepnya sama dengan *playback*, yaitu memainkan kembali data nada/chord yang disimpan oleh user.

3.2.2. Data Suara

Data suara berupa *file* data yang berformat *"*.mid"* yang diambil dari sebuah aplikasi gitar. Dalam 1 file suara terdapat 1

nada (single chromatic/diatonic) atau chord (full chromatic/diatonic). Pada aplikasi ini secara keseluruhan terdapat 119 file suara berformat **.mid*, dimana file ini terdiri dari 55 suara untuk Menu Single Chromatic dan Diatonic serta 64 suara untuk Menu Full Chromatic dan Full Diatonic. Untuk Menu Single Chromatic dan Single Diatonic, file suara dibagi-bagi berdasarkan nilai oktafnya (Menu Single Chromatic dan Single Diatonic ini mempunyai 5 oktaf). Sedangkan untuk Menu Full Chromatic dan Menu Full Diatonic, file suara dibagi berdasarkan jenis chord (major/minor) dan oktaf. Data suara ini diambil dari Software Guitar Pro v5.2, yaitu dengan cara mengekspor suara dari software tersebut dalam bentuk **.mid*.

Suara akan keluar bergantung pada inputan nada/chord oleh user. Suara juga akan keluar pada saat user melakukan *Playback* dari nada/chord yang dimainkan. Pada saat *playback*, system akan melakukan pergantian audio berdasarkan data nada/chord yang dimainkan user pada saat melakukan proses *recording*. Begitupula pada Menu Load Nada dan Menu Load Chord, pada kedua menu ini konsepnya sama dengan *playback*, yaitu memainkan kembali data nada/chord yang disimpan oleh user.

Pada mode permainan kromatik, baik *single* maupun *chord*, aplikasi dapat membunyikan 12 not sesuai dengan aturan tangga nada kromatik yaitu setiap nada berjarak $\frac{1}{2}$ nada. Berikut ini nada-nada yang dibunyikan pada aplikasi gitar player dalam aturan tangga nada kromatik.

Tabel 3.2.2.a. Single/Full Chromatic

Single/Full Chromatic												
Jumlah Nada dasar	1	2	3	4	5	6	7	8	9	10	11	12
C	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C#	C#	D	D#	E	F	F#	G	G#	A	A#	B	C
D	D	D#	E	F	F#	G	G#	A	A#	B	C	C#
D#	D#	E	F	F#	G	G#	A	A#	B	C	C#	D
E	E	F	F#	G	G#	A	A#	B	C	C#	D	D#
F	F	F#	G	G#	A	A#	B	C	C#	D	D#	E
F#	F#	G	G#	A	A#	B	C	C#	D	D#	E	F
G	G	G#	A	A#	B	C	C#	D	D#	E	F	F#
G#	G#	A	A#	B	C	C#	D	D#	E	F	F#	G
A	A	A#	B	C	C#	D	D#	E	F	F#	G	G#
A#	A#	B	C	C#	D	D#	E	F	F#	G	G#	A
B	B	C	C#	D	D#	E	F	F#	G	G#	A	A#

Lain halnya dengan mode permainan diatonik. Pada mode permainan diatonik baik *single* maupun *chord*, aplikasi hanya dapat membunyikan 7 not sesuai dengan aturan tangga nada diatonik yaitu setiap nada berjarak 1,1,1/2,1,1,1/2.

Tabel 3.2.2.b. Full Diatonic

Full Diatonic							
Jumlah	1	2	3	4	5	6	7
Nada dasar	C	C	D	E	F	G	A
Nada dasar	C#	C#	D	E	F	G	A
Nada dasar	D	D	E	F	G	A	B
Nada dasar	D#	D#	E	F	G	A	B
Nada dasar	E	E	F	G	A	B	C
Nada dasar	F	F	G	A	B	C	D
Nada dasar	F#	F#	G	A	B	C	D
Nada dasar	G	G	A	B	C	D	E
Nada dasar	G#	G#	A	B	C	D	E
Nada dasar	A	A	B	C	D	E	F
Nada dasar	A#	A#	B	C	D	E	F
Nada dasar	B	B	C	D	E	F	G

3.2.3. Data Record Management System (RMS)

Record Management System (RMS) digunakan untuk menyimpan data-data

nada/chord yang dimainkan pada saat *recording*. Penyimpanan nada/chord ini dilakukan pada saat user memberhentikan proses record (*stop record*). Data yang disimpan dalam Record Management System ini berupa data *String* dengan format tertentu. Format yang disimpan dalam RMS adalah sebagai berikut:

Judul_lagu;waktu_jeda1,oktaf1,semitone1,nada1;

waktu_jeda2,oktaf2,semitone2,nada2;.....;waktu_jeda_ke-n,0,0,"0"

Contohnya:

IbuKartini;7,3,0,C;15,3,0,D;20,3,0,E;27,3,0,F;32,3,0,G;42,3,0,E;48,3,0,C;64,0,0,0

Pada saat proses record, user menekan sebuah tombol angka, maka secara otomatis sistem akan menyimpan satu blok dari format tersebut (*;7,3,0,C -> ;waktu_jeda1,oktaf1,semitone1,nada1*) ke dalam sebuah variabel. Saat user menekan tombol angka lagi saat proses record, maka secara otomatis sistem akan menambahkan lagi satu blok dari format tersebut (*15,3,0,D -> ;waktu_jeda2,oktaf2,semitone2,nada2*). Begitu seterusnya sampai user menghentikan proses record (*stop record*), dan pada saat itulah sistem secara otomatis akan menambahkan blok yang terakhir (*;64,0,0,0 -> ;waktu_jeda_ke-n,0,0,"0"*) yang menandakan akhir dari proses record sekaligus akhir dari format data RMS.

Data nada/chord yang akan disimpan dalam Record Management System ditampung dalam sebuah array of *TampungNada* dimana *TampungNada* merupakan sebuah class untuk menampung data nada/chord yang akan disimpan dalam Record Management System. Pengambilan data dari database RMS, dilakukan pada saat aplikasi pertama kali melakukan startup. Pertama kali sistem akan mengambil data string yang ada dalam database RMS. Kemudian sistem akan memecah-mecah string tersebut ke dalam

tempatny masing-masing dan ditampung pada class TampungNada (untuk penyimpanan nada) dan class TampungChord (untuk penyimpanan chord).

Dalam format yang disimpan dalam Record Management System tersebut, terdapat waktu_jeda. Waktu jeda ini juga memegang peranan yang sangat penting. Waktu_jeda tersebut merupakan jeda waktu antara nada-nada/chord-chord yang dimainkan. Satuan waktu_jeda ini adalah milisecond. Waktu ini dimulai pada saat user memulai proses record. Untuk menjalankan timer ini diperlukan class HitungTimerChord yang merupakan class turunan dari class TimerTask.

Timer juga mempunyai peran penting dalam melakukan playback. Karena waktu inilah yang digunakan sebagai patokan kapan harus memainkan nada/chord yang disimpan. Untuk menghandle proses playback ini dilakukan oleh inner class yaitu class HitungTimerChordPlayback. Class ini juga merupakan class turunan dari class TimerTask. Method yang menghandle playback ini ialah method run() yang terdapat dalam class HitungTimerChordPlayback. Method ini dipanggil ketika user mulai memilih menu playback. Selain digunakan untuk playback pada Menu Single Chromatic, Full Chromatic, Single Diatonic, dan Full Diatonic, class HitungTimerChordPlayback ini juga digunakan dalam Menu Load Nada dan Load Chord.

3.3. PEMBUATAN APLIKASI

3.3.1. Desain User Interface

Dalam pembuatan aplikasi ini diperlukan desain layout atau antar muka sebagai jembatan antara user dengan aplikasi. Secara umum interface atau halaman terdiri dari beberapa bagian penting yaitu:

1. Menu Splash Screen,
2. Menu Utama,
3. Menu Single Chromatic,
4. Menu Full Chromatic,

5. Menu Single Diatonic,
6. Menu Full Diatonic,
7. Menu Load Nada,
8. Menu Load Chord,
9. Menu Help,
10. Menu About.

4. UJI COBA DAN ANALISA

Pada handphone Sony Ericsson K618i memiliki resolusi/ukuran layar 176x220 pixel, sehingga gambar/tampilan aplikasi bisa terlihat pas pada layar. Dengan ukuran layar yang pas, maka semua gambar tertata dengan rapi. Pada handphone ini, aplikasi ini berjalan dengan baik.

Sama dengan Sony Ericsson K618i, handphone Sony Ericsson Z500i ini juga memiliki resolusi layar 176x220 pixel, sehingga gambar/tampilan aplikasi bisa terlihat pas pada layar. Dengan ukuran layar yang pas, maka semua gambar tertata dengan rapi. Aplikasi berjalan kurang begitu bagus, dikarenakan handphone jenis ini menggunakan OS Java yang lama, sehingga aplikasi ini berjalan agak berat.

Sony Ericsson C510 memiliki ukuran layar 240x320. Dengan ukuran yang lebih lebar dari ukuran aplikasi ini, maka tampilan aplikasi ini juga melebar sesuai ukuran layar. Walaupun layar terisi penuh, tetapi gambar yang ada pada aplikasi terlihat pecah. Design menu utama aplikasi pun terlihat tidak presisi, begitu juga design menu-menu seperti Single Chromatic, Full Chromatic, Single Diatonic, Full Diatonic, Help, dan About desain terlihat berantakan karena ukuran layar yang lebih besar. Secara umum aplikasi ini berjalan sangat baik. Pada handphone ini, aplikasi berjalan kurang begitu bagus.

Sama seperti Sony Ericsson C510, handphone Sony Ericsson G502 ini juga memiliki ukuran layar 240x320. Dengan ukuran yang lebih lebar dari ukuran aplikasi ini, maka tampilan aplikasi ini juga melebar sesuai ukuran layar. Walaupun layar terisi penuh,

tetapi gambar yang ada pada aplikasi terlihat pecah. Design menu utama aplikasi pun terlihat tidak presisi, begitu juga design menu-menu seperti Single Chromatic, Full Chromatic, Single Diatonic, Full Diatonic, Help, dan About desain terlihat berantakan karena ukuran layar yang lebih besar. Secara umum aplikasi ini berjalan cukup baik.



C. PENUTUP

Aplikasi *Gitar Player* ini terdiri atas 4 mode utama yaitu *single chromatic*, *full chromatic*, *single diatonic* dan *full diatonic*. Pada *full mode*, dibagi menjadi 2 sub mode yaitu *full chromatic/diatonic major* dan *full chord chromatic/diatonic minor*. Pada *single chromatic*, *full chromatic*, *single diatonic* dan *full diatonic* terdapat fitur tambahan yaitu

Record dan *Playback* serta beberapa fasilitas lain seperti *sustain*, transpose nada/chord, serta transpose oktaf. Dengan tambahan kedua fungsi tersebut, aplikasi dapat memenuhi kebutuhan *user* untuk menyimpan nada yang sedang dimainkan, baik itu dari sebuah lagu ataupun dari lagu ciptaan sendiri yang nantinya dapat dimainkan kembali.

Untuk sekedar saran yang dapat diberikan untuk pengembangan yang mungkin dapat dilakukan pada aplikasi *Gitar Player* ini adalah :

1. Untuk selanjutnya aplikasi *Gitar Player* ini dapat lebih dikembangkan lagi fitur-fitur yang terdapat di dalamnya.
2. Aplikasi *Gitar Player* ini lebih dikembangkan pada tingkat kompatibilitas, audio dan visualnya.

D. DAFTAR BACAAN

- Basuki Kurnia Dwi. *RevisedSQA-1.ppt*. Bahan Ajar Mata Kuliah Manajemen Kualitas Perangkat Lunak : PENS-ITS.
- Nugroho, Yohanes. 2005. *J2ME-MIDP*.
- Raharjo, Budi dkk. 2007. *Tuntunan Pemrograman Java untuk Handphone*. Bandung : Informatika Bandung.
- Shalahuddin M dan AS, Rosa. 2008. *Pemrograman J2ME*. Bandung : Informatika Bandung.
- Wicaksono, Ady. 2002. *Pemrograman Aplikasi Wireless Dengan Java*. Jakarta : PT.Elex Media Computindo.
- <http://www.andisun.com/category/tutorials/j2me-programming>.
- http://www.java2s.com/Tutorial/Java/0430_J2ME/1000Tones.htm.
- <http://www.java2s.com/Code/Java/J2ME/CustomItem.htm>.
- <http://www.developer.com/java/j2me/article.php/1561591>.
- http://www.java2s.com/Tutorial/Java/0430_J2ME/ControllableMIDI.htm.

[http://www.java2s.com/Tutorial/Java/0430
J2ME/PlayMIDI.htm.](http://www.java2s.com/Tutorial/Java/0430_J2ME/PlayMIDI.htm)

[http://www.java2s.com/Tutorial/Java/0430
J2ME/MIDIEvents.htm.](http://www.java2s.com/Tutorial/Java/0430_J2ME/MIDIEvents.htm)