

MAKALAH SEMINAR TUGAS AKHIR

APLIKASI KONVERSI SATUAN PADA PONSEL MENGGUNAKAN *JAVA 2 MICRO EDITION (J2ME)*

Sintha Saptiwiningrum*, Agung Budi P.**, Aghus Sofwan**

Abstrak - Java 2 Micro Edition (J2ME) merupakan sebuah versi yang direduksi dari Java API dan Java Virtual Machine (JVM) yang didesain untuk dapat dioperasikan dalam komputer dan mikrokomputer. Kelahiran platform J2ME timbul karena dibutuhkan adanya sebuah platform komputasi yang mengakomodasi piranti consumer electronics dan embedded system. Dengan kata lain, J2ME diarahkan untuk aplikasi pada piranti komputasi kecil seperti ponsel, Personal Digital Assistan (PDA), palm, pager, dan lain-lain. Dengan diperkenalkan bahasa pemrograman Java untuk piranti mobile tersebut, kita dapat dengan mudah membuat beberapa aplikasi sesuai dengan kehendak atau keinginan kita. Untuk itu, kemampuan piranti dengan ‘Micro Edition’ dimanfaatkan untuk membuat suatu perangkat lunak yaitu aplikasi konversi satuan.

Dalam dunia pemrograman, perencanaan dan perancangan terhadap suatu perangkat lunak merupakan faktor penting yang paling berpengaruh dalam memenuhi kebutuhan suatu aplikasi yang handal. Java merupakan bahasa pemrograman yang berbasis pada objek, untuk itu dalam perancangannya diperlukan pula suatu pemodelan. Pemodelan adalah proses merancang piranti lunak sebelum melakukan pengkodean (coding). Pemodelan dalam pembuatan sebuah perangkat lunak merupakan faktor yang sangat penting karena berkaitan dengan arsitektur (bentuk antarmuka), kehandalan terhadap kesalahan, serta kemampuan menghasilkan perangkat lunak yang dapat berinteraksi dengan pengguna.

Kata kunci : J2ME, ponsel, konversi satuan, pemodelan

I. PENDAHULUAN

1.1 Latar Belakang

Java 2 Micro Edition (J2ME) adalah salah satu bahasa pemrograman yang memberikan revolusi yang cukup signifikan bagi teknologi telekomunikasi bergerak secara keseluruhan. Dengan kompatibilitasnya yang tinggi, pertumbuhan perangkat yang mendukung J2ME pun juga akan bergerak mengikuti perkembangan aplikasi-aplikasi baru yang muncul silih berganti. Teknologi yang ditawarkan oleh Java yaitu ‘*write once run anywhere*’, dimana pemrograman Java (baik *source* program maupun

hasil kompilasinya) sama sekali tidak bergantung pada sistem operasi dan platform yang digunakan (*platform independence*).

Dalam perkembangannya Java juga digunakan sebagai standar bahasa pemrograman *mobile device* yang telah didukung oleh banyak vendor telepon genggam seperti Nokia dengan 7650, 3650, 3660, Siemens dengan M50 dan SL45i, Ericson dengan P800 dan P910. Salah satu ciri khas perangkat-perangkat ini adalah jumlah memori dan kapasitas penyimpanan yang terbatas, serta kemampuan antarmuka pengguna yang terbatas. Aplikasi-aplikasi yang dapat dikembangkan oleh J2ME ini dapat berupa aplikasi bisnis, perkantoran, permainan (*game*) ataupun konversi satuan. Oleh karena itu, sebagai teknologi baru, J2ME memberikan peluang baru bagi para developer atau pemula yang ingin berkecimpung dalam bisnis aplikasi komunikasi bergerak ini.

1.2 Tujuan Penulisan

Tujuan yang ingin dicapai dalam pelaksanaan Tugas Akhir ini adalah merancang dan membuat program konversi satuan yang dapat diaplikasikan pada ponsel yang telah mendukung Java, dengan menggunakan bahasa pemrograman *Java 2 Micro Edition (J2ME)*.

1.3 Batasan Masalah

Untuk menyederhanakan pembahasan dalam Tugas Akhir ini, maka diberikan beberapa batasan masalah sebagai berikut :

1. Proses pembuatan aplikasi program Konversi Satuan menggunakan *Java 2 Micro Edition (J2ME)* yang mendukung MIDP 1.0.
2. Aplikasi *Wireless J2ME* yang dibuat berupa MIDlet Konversi dan dalam aplikasi ini hanya menggunakan 7 (tujuh) macam besaran yaitu Daya, Energi, Kecepatan, Luas, Massa, Panjang dan Volume.
3. Perancangan sistem dalam Tugas Akhir ini menggunakan bahasa pemodelan *Unified Modelling Language (UML)*.

* Mahasiswa Jurusan Teknik Elektro UNDIP

** Staf Pengajar Jurusan Teknik Elektro UNDIP

Aplikasi konversi satuan ini hanya disimulasikan menggunakan emulator dari J2ME WTK 1.0.4 yaitu DefaultColorPhone, emulator Motorola SL VR_L7 dan emulator Nokia Seri 60 (EPOC) serta diaplikasikan hanya untuk ponsel yang mempunyai platform *Java Enable Phone* dengan *real device test* yaitu Nokia 3650, untuk transfer data menggunakan konektivitas *bluetooth*.

II. TEKNOLOGI JAVA

2.1 Edisi Java

Sun Microsystem telah mendefinisikan tiga platform Java, yaitu :

- *Java 2 Standart Edition* (J2SE) : didesain untuk komputer *desktop* dan komputer *workstation*.
- *Java 2 Enterprise Edition* (J2EE) : edisi ini ditujukan untuk aplikasi berbasis server.
- *Java 2 Micro Edition* (J2ME) : didesain untuk piranti dengan memori, layar display, dan power pemrosesan yang terbatas.

2.2 Pemrograman Java

Java adalah sebuah bahasa pemrograman komputer berbasis pada *Object Oriented Programming* (OOP). Salah satu keunggulan Java adalah sifatnya yang '*platform independence*', artinya Java (baik *source* program maupun hasil kompilasinya) sama sekali tidak bergantung pada sistem operasi dan platform yang digunakan, dengan kata lain Java bersemboyan "*Write Once Run Anywhere*".

Lingkungan *development* Java terdiri atas dua bagian yaitu *Java compiler* dan *Java Interpreter*^[8]. *Java Compiler* menerjemahkan *source* program ke dalam *bytecode*. Hasil kompilasi, yaitu program Java, akan bisa dijalankan dengan bantuan *Java interpreter*.

2.3 Java 2 Micro Edition (J2ME)

J2ME merupakan subset dari *Java 2 Standart Edition* (J2SE) yang ditujukan untuk implementasi pada peralatan *embedded system* dan *handheld*^[2].

J2ME terdiri atas beberapa komponen, yaitu :

1. Java API, merupakan kumpulan *library* untuk menjalankan dan mengembangkan program Java pada *handheld device*.
2. *Java Virtual Machine* (JVM), untuk menjalankan program-program Java pada emulator atau perangkat genggam.
3. Tools lain untuk pengembangan aplikasi Java, semacam *emulator Java Phone*.

2.3.1 Konfigurasi J2ME

Konfigurasi adalah spesifikasi yang secara detail menjelaskan tentang sebuah *virtual machine* dan kumpulan API dasar yang digunakan dalam kelas-kelas tertentu dari sebuah peralatan^[2]. Dan karena disini ponsel sebagai perangkat *mobile* yang digunakan, maka digunakan konfigurasi *Connected Limited Device Configuration* (CLDC).

CLDC adalah konfigurasi untuk peralatan *wireless* dengan kapasitas memori yang kecil dengan koneksi jaringan yang tidak tetap atau terputus-putus, seperti pada ponsel^[2].

2.3.2 Profil J2ME

Sebuah profil dibangun dalam sebuah konfigurasi dengan menambahkan beberapa API khusus agar dihasilkan sebuah lingkungan yang lengkap untuk membangun aplikasi^[2]. Salah satu spesifikasi profil yang dibuat dengan berbasis pada CLDC adalah *Mobile Information Device Profile* (MIDP).

CLDC dengan profil MIDP 1.0 mempunyai beberapa keterbatasan, diantaranya adalah :

1. Tidak ada dukungan *floating point*.
2. Penanganan kesalahan/*exception* yang terbatas dimana CLDC hanya mendefinisikan 3 (tiga) kelas.

2.3.3 Java Virtual Machine (JVM)

JVM adalah mesin untuk sembarang aplikasi Java. JVM juga bertanggung jawab untuk keamanan, mengalokasikan memori dan mengelola eksekusi *thread*^[9]. Hal ini yang menyebabkan program yang dibuat dapat berjalan. Untuk CLDC menggunakan *K Virtual Machine*^[9]. *K Virtual Machine* adalah *virtual machine* yang sangat kecil dalam kebutuhan memorinya, yaitu mulai dari 160 kB hingga maksimal sekitar 512 kB.

2.3.4 Dasar-dasar MIDlet

MIDlet adalah aplikasi yang dibuat menggunakan J2ME dengan profil MIDP.

2.3.4.1 Lifecycle

Lifecycle dari sebuah MIDlet ditangani oleh *Application Management Software* (AMS). AMS adalah sebuah lingkungan tempat siklus dari sebuah MIDlet mampu diciptakan, dijalankan, dihentikan, maupun dihilangkan^[2].

MIDlet memiliki beberapa state, yaitu *Pause*, *Active*, dan *Destroy*. Ketika masing-masing state dipanggil, beberapa method standar yang bersesuaian

dipanggil. Method-method ini merupakan bawaan dari J2ME.

2.3.4.2 User Interface

User interface dari MIDP terdiri atas beberapa API (*Application Programming Interface*), yaitu:

1. *High Level*, berbasis pada *Screen*. Contoh dari *high level* API adalah *alert, form, list, textbox*.
2. *Low Level*, berbasis pada kelas *Canvas*.

2.3.5 Pembuatan Aplikasi MIDlet

Tahap-tahap pembuatan aplikasi MIDlet adalah sebagai berikut :

1. Membuat file `.java` yang merupakan kode sumber dari suatu aplikasi.
2. Mengkompilasi file `.java` sehingga dihasilkan file `.class`.
3. File `.class` dilewatkan ke *preventifier* untuk mengatur susunan *bytecode* dari file `.class` yang bertujuan untuk mempermudah saat dilakukan verifikasi pada CLDC *Virtual Machine*.
4. Jika tidak terdapat kesalahan dalam proses mengkompilasi dan proses verifikasi maka aplikasi dapat dijalankan menggunakan emulator.

2.3.6 Packaging (Pemaketan)

Proses *Packaging* adalah proses untuk membungkus aplikasi MIDlet menjadi sebuah file `*.JAR` dan sebuah file `*.JAD`. Pada setiap file `JAR` dapat berisi satu atau lebih aplikasi MIDlet. Sedangkan file `JAD` berguna untuk mendeskripsikan isi dari file `JAR`.

2.4 J2ME Wireless Toolkit

J2ME Wireless Toolkit merupakan tool yang menyediakan lingkungan emulator, dokumen dan contoh-contoh aplikasi Java untuk perangkat kecil.

Ktoolbar adalah bagian yang sudah dimasukkan dalam paket *J2ME WTK*, merupakan suatu *Integrated Development Environment (IDE)* yang minimal dan sederhana dengan *Graphic User Interface (GUI)* untuk kompilasi, pemaketan, dan menjalankan aplikasi^[14]. Dalam *Ktoolbar* ini juga tersedia skin-skin emulator yang dapat digunakan dan tentunya telah mendukung spesifikasi MIDP.

2.5 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (*object oriented programming*) ibaratnya seperti membangun sesuatu yang besar dan kompleks dengan menggunakan objek-objek yang telah tersedia tanpa

harus mengetahui detail dari objek-objek yang digunakan^[8].

III. PERANCANGAN SISTEM

3.1 Tahap Requirement

Tahap *requirement* bertujuan untuk mengetahui apa yang sebenarnya dibutuhkan oleh pengguna sehingga menginginkan sebuah perangkat lunak dibuat.

Perangkat lunak yang dikembangkan dinamakan konversi satuan dengan aplikasi yang bernama KonversiMIDlet. Tujuan dibuatnya perangkat lunak ini adalah untuk memudahkan user mengkonversi satuan dari suatu emulator dan perangkat genggam yang telah mendukung Java. Untuk mengetahui lebih jelas tentang aplikasi apa yang akan dibuat, terlebih dahulu dijelaskan beberapa pengertian-pengertian dasar yang berkaitan dengan pembuatan aplikasi.

Besaran adalah sesuatu yang dapat diukur dan dinyatakan dengan angka dan satuan adalah ukuran suatu besaran^[4]. Penggunaan bermacam-macam satuan untuk suatu besaran ternyata menimbulkan kesukaran. Salah satu kesukaran adalah harus beralih dari satu satuan ke satuan lainnya dengan cukup rumit.

Untuk besaran, satuan dan faktor konversinya, dapat dilihat dalam Tabel 3.1^{[4],[10]}.

Tabel 3.1 Konversi.

No	Besaran	Satuan yang di konversi	Faktor konversi
1	Daya	dk ke kW	0,746
		kW ke dk	1,3405
2	Energi	kalori ke Joule	4,186
		Joule ke kalori	0,2389
3	Kecepatan	knot ke km/jam	1,852
		km/jam ke knot	0,54
		km/jam ke m/dtk	0,2778
		m/dtk ke km/jam	3,6
		mil/jam ke m/dtk	0,447
		m/dtk ke mil/jam	2,2372
4	Luas	ha ke km ²	0,01
		km ² ke ha	100
		acre ke ha	0,4047
		ha ke acre	2,471
		inc ² ke cm ²	6,452
		cm ² ke inc ²	0,155

		feet ² ke m ²	0,0929
		m ² ke feet ²	10,7643
5	Massa	pounds ke oz	16
		oz ke pounds	0,0625
		gr ke oz	0,0353
		oz ke gr	28,3495
		kg ke pounds	2,2046
		pounds ke kg	0,4536
6	Panjang	cm ke inc	0,3937
		inc ke cm	2,54
		m ke feet	3,3281
		feet ke m	0,3048
		km ke mil	0,6214
		mil ke km	1,6093
		yard ke m	0,9144
		m ke yard	1,0937
7	Volume	ltr ke m ³	0,001
		m ³ ke ltr	1000
		gal(US) ke ltr	3,7854
		ltr ke gal(US)	0,2642
		gal(US) ke ft ³	0,1337
		ft ³ ke gal(US)	7,4805

Untuk detail dari apa yang dibutuhkan user dari perangkat lunak ini adalah sebagai berikut :

“Dalam aplikasi konversi satuan ini *user* dapat mengubah satuan besaran, dari satu satuan ke satuan yang lain. Dari tampilan menu utama, *user* dapat memilih salah satu pilihan, yaitu ‘Tipe Besaran’ dan ‘Petunjuk’. Bila *user* memilih ‘Petunjuk’, akan ditampilkan halaman Petunjuk yang berisi instruksi penggunaan aplikasi konversi satuan ini. Bila user memilih ‘Tipe Besaran’, akan ditampilkan halaman yang berisi daftar 7 (tujuh) macam besaran, yaitu Daya, Energi, Kecepatan, Luas, Massa, Panjang dan Volume. Bila salah satu besaran telah dipilih atau diklik, maka halaman Satuan akan ditampilkan. Dalam halaman satuan, user dapat memilih salah satu satuan yang akan dikonversi, dan dalam halaman Konversi, user dapat mengetikkan sejumlah nilai pada *text field* satuan yang akan diubah. Untuk bilangan pecahan, maksimal 4 (empat) angka dibelakang koma dan ketik pada *text field* Koma. Tekan tombol Hitung, sehingga muncul nilai yang diinginkan. Yang perlu diingat disini adalah apabila mengetikkan nilai pada *textfield*

Koma terlebih dahulu meng-*clear* angka 0000 dan pastikan dalam *textfield* Koma selalu terdapat 4 (empat) digit.”

Dari uraian diatas, maka diperlukan beberapa menu/halaman yang harus disediakan, antara lain adalah :

1. Menu Utama

Berisi daftar (*list*) pilihan menggunakan List.IMPLICIT , yaitu : Tipe Besaran dan Petunjuk serta terdapat *CommandListener Exit* untuk keluar dari aplikasi.

2. Menu Besaran

Berisi daftar (*list*) besaran menggunakan List.IMPLICIT, yaitu : Daya, Energi, Kecepatan, Luas, Massa, Panjang, Volume serta terdapat *CommandListener Kembali* untuk kembali ke menu utama.

3. Menu Satuan

Berisi daftar (*list*) satuan yang menggunakan List.IMPLICIT, dan terdapat *CommandListener Menu* untuk kembali ke Menu Utama dan **Kembali** untuk kembali ke menu Besaran.

4. Menu Konversi

Ditampilkan *textfield* untuk diisi nilai satuan yang akan diubah. Terdapat *CommandListener Hitung* untuk mengetahui hasil konversi dan **Kembali** untuk kembali ke halaman Satuan.

5. Menu Petunjuk

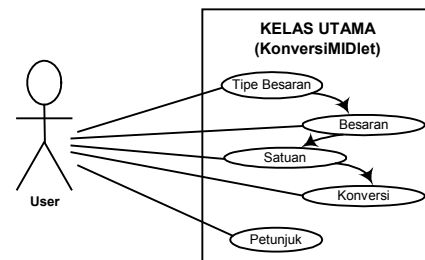
Berupa halaman *Form* yang berisi petunjuk menggunakan aplikasi konversi satuan. Terdapat *CommandListener Ok* untuk kembali ke Menu Utama.

3.2 Tahap Analisis

Tahap analisis bertujuan untuk mendapatkan definisi yang jelas mengenai fungsi dari sebuah perangkat lunak.

3.2.1 Use Case

Use case terdiri atas aktor, *use-case* serta interaksi antara aktor dan *use-case*.



Gambar 3.1 Diagram Use Case Aplikasi Konversi Satuan.

Dari diagram use case diatas dapat dijelaskan sebagai berikut :

User menjalankan aplikasi, akan ditampilkan menu-menu pilihan menggunakan `List.IMPLICIT` yaitu dalam menu utama, menu besaran, dan menu satuan serta ditampilkan menu konversi, dan menu petunjuk. Dalam tiap-tiap menu juga terdapat `commandListener`.

3.2.2 Objek-objek

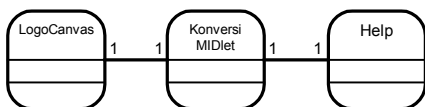
Dalam sistem aplikasi ini terdapat kelas `KonversiMIDlet`, yang akan mengatur inialisasi dan kondisi awal dari objek-objek di bawahnya dan sebagai pengontrol utama dari objek-objek yang ada.

Selain kelas utama terdapat pula objek-objek lain yang berfungsi untuk menangani suatu perintah tertentu dari pengguna, yaitu :

1. Objek `LogoCanvas`
Sebagai tempat logo digambar dan ditampilkan serta terdapat `commandListener` untuk mengetahui aksi *user* terhadap perangkat *mobile*.
2. Objek `Help`
Objek `Help` ini berfungsi untuk menampilkan menu/halaman petunjuk menggunakan aplikasi konversi satuan dalam format *text string*.

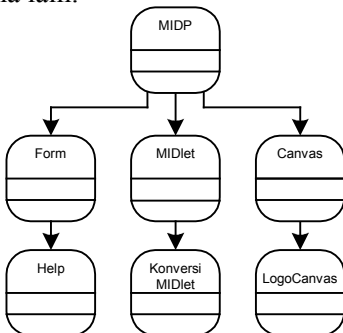
3.2.3 Diagram Kelas

Setelah menentukan objek-objek yang digunakan, maka dapat dibuat suatu diagram hubungan dari kelas-kelas yang digunakan dari MIDlet yang telah dibuat.



Gambar 3.2 Diagram Hubungan dari Kelas-kelas yang Digunakan

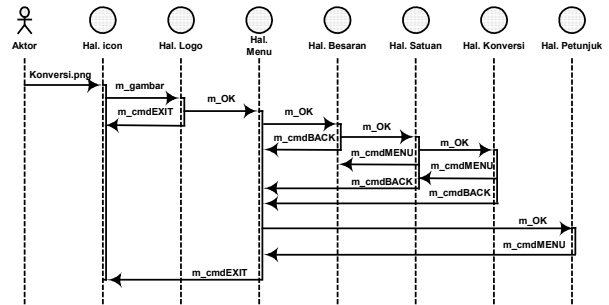
Diagram kelas menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain.



Gambar 3.3 Diagram Kelas (Hierarki).

3.2.4 Diagram Sequence

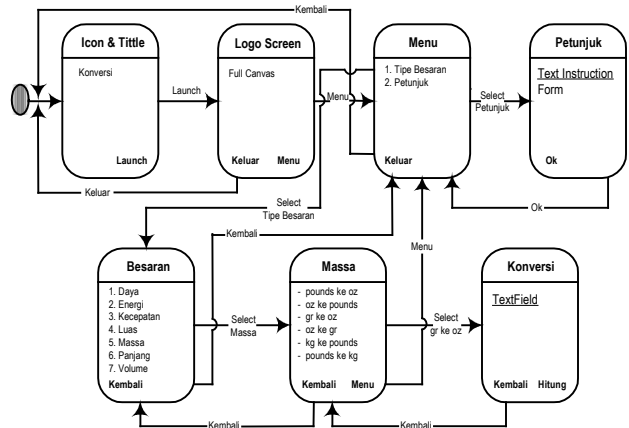
Diagram *sequence* menjelaskan tentang interaksi objek yang disusun dalam urutan waktu dan berkolaborasi dengan diagram *use case*.



Gambar 3.4 Diagram Sequence Aplikasi Konversi Satuan.

3.2.5 Diagram Statechart

Diagram *statechart* akan menggambarkan transisi dan perubahan keadaan dari satu state ke state lainnya suatu objek.



Gambar 3.5 Diagram Statechart Aplikasi Konversi Satuan.

3.3 Perancangan Lingkungan Pengembangan

`KonversiMIDlet` ini digunakan untuk mengkonversi satuan yang dijalankan pada telepon genggam yang telah mendukung Java serta *complaint* dengan MIDP 1.0.

3.3.1 Perangkat Masukan dan Keluaran

Perangkat masukan dan keluaran pada telepon genggam berupa *keypad* yang fungsinya berbeda-beda, seperti yang terlihat pada Gambar 3.6.



Gambar 3.6 Perangkat Masukan dan Keluaran Nokia 3650.

3.3.2 Perancangan Antarmuka

Perancangan antarmuka tersedia dalam paket `javax.microedition.lcdui`. *Class Canvas* digunakan untuk tempat menggambar logo. *Class Screen* terdiri atas beberapa *subclass*, diantaranya yaitu *List Implicit* dipergunakan untuk menampilkan Menu Utama, Menu Besaran dan Menu Satuan, *Text Field* dipergunakan sebagai masukan nilai dalam Menu Konversi, dan *StringItem* dipergunakan untuk menampilkan teks yang digunakan dalam Menu Petunjuk.

IV. PENGUJIAN SISTEM

Pada bab ini akan dijelaskan pengujian sistem aplikasi konversi satuan secara keseluruhan. Pengujian aplikasi konversi satuan dengan melakukan proses kompilasi, proses verifikasi serta proses *packaging*.

Apabila ketiga proses telah dilaksanakan maka tahap selanjutnya adalah proses instalasi aplikasi konversi satuan pada Nokia 3650, yaitu dengan mentransfer file `Konversi.JAR` dan file `Konversi.JAD` menggunakan konektivitas *bluetooth*. Apabila proses transfer selesai, kedua file tersebut dapat langsung instalasi ke dalam Nokia 3650 dalam *folder Application*.

4.1 Pengujian Aplikasi Menggunakan *Real Device*

Pengujian aplikasi konversi satuan menggunakan *real device* Nokia 3650.

4.1.1 Halaman *Icon* dan Judul

Halaman ini akan menampilkan *icon* konversi satuan yang dibuat berformat *png* (*Portable Network Graphics*) dan nama projek yang dibuat, yaitu `Konversi`.



Gambar 4.1 Halaman *Icon* dan Judul.

4.1.2 Halaman Logo

Halaman logo merupakan implementasi dari kelas `LogoCanvas.java` yang akan menampilkan logo aplikasi konversi satuan dari format `PNG` yang diambil dari file `logo.png`.



Gambar 4.2 Halaman Logo.

4.1.3 Halaman Menu Utama

Pada halaman menu utama Nokia 3650 akan menampilkan sebuah daftar (*list*) yang berisi pilihan (menggunakan `List.IMPLICIT`) dari aplikasi yang diperlukan oleh *user*, yaitu Tipe Besaran dan Petunjuk.



Gambar 4.3 Halaman Menu Utama.

Untuk *Command*, *left soft key* yaitu "*Option*" dan *right soft key* yaitu "*Exit*" untuk keluar dari aplikasi. Untuk memilih salah satu pilihan maka tekan tombol gulir (*scroll-key*).

4.1.4 Halaman Besaran

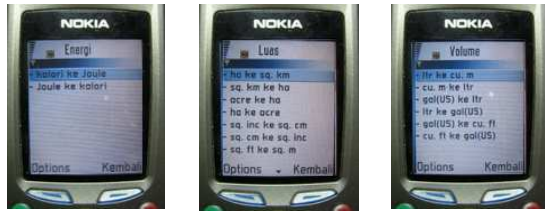
Pada halaman besaran, menampilkan sebuah daftar (*list*) yang berisi 7 (tujuh) macam besaran (menggunakan `List.IMPLICIT`), yaitu Daya, Energi, Kecepatan, Luas, Massa, Panjang dan Volume.



Gambar 4.4 Halaman Besaran.

4.1.5 Halaman Satuan

Pada halaman satuan Nokia 3650 menampilkan sebuah daftar (*list*) yang berisi beberapa macam satuan (menggunakan `List.IMPLICIT`), misalkan besaran yang menjadi contoh adalah Energi maka akan tampil beberapa satuan Energi, yaitu kalori ke Joule dan Joule ke kalori. Demikian pula bila yang dipilih adalah Luas, akan ditampilkan ha ke km^2 , km^2 ke ha, acre ke ha, ha ke acre, inc^2 ke cm^2 , cm^2 ke inc^2 , ft^2 ke m^2 , m^2 ke ft^2 . Dan sebagai contoh disini akan ditampilkan satuan dari Energi, Luas dan Volume.



Gambar 4.5 Halaman Satuan dari Energi, Luas dan Volume.

4.1.6 Halaman Konversi

Halaman konversi akan menampilkan posisi *scroll* pada *textfield* satuan asal, di sini dapat diketikkan nilai yang akan diubah, untuk bilangan pecahan, maksimal 4 (empat) angka dibelakang koma dan ketik pada *textfield* Koma. Dan untuk mengetahui hasilnya dengan menekan “Option” pilih *command* Hitung, sehingga nilainya akan tampil pada *text field* satuan tujuan.



Gambar 4.6 Halaman Konversi.

Dalam halaman Konversi ini, yang perlu diingat adalah saat akan mengetikkan nilai pada *text field* koma, dimana terlebih dahulu harus meng-clear angka 0000 dan pastikan dalam *text field* ini terisi 4 digit.

4.1.7 Halaman Petunjuk

Halaman Petunjuk merupakan implementasi dari kelas `help.java` yang menampilkan instruksi bagaimana menggunakan aplikasi konversi satuan ini, menggunakan *Form* sebagai subkelas dari *screen* dengan judul “Petunjuk”.



Gambar 4.7 Halaman Petunjuk.

Dari Gambar 4.7, instruksi yang ditampilkan harus menggulung ke bawah (*scroll down*) dan menekan tombol panah bawah agar halaman dapat terlihat seluruhnya. Untuk tombol *Command* terdapat

“Option” yang berisi *command* Ok, untuk kembali ke halaman menu utama.

4.2 Pengujian Program

Bilangan pecahan (*floating point*) digunakan pada fungsi perhitungan yang membutuhkan ketelitian pecahan, namun dalam MIDP 1.0 tidak ada dukungan untuk *floating point* sehingga `java.lang.Float` dan `java.lang.Double` tidak ada dalam MIDP 1.0. Oleh karena itu, dalam aplikasi konversi satuan ini, untuk perhitungannya menggunakan tipe data *long*. *Long* merupakan tipe 64-bit bertanda. Rentang nilai untuk *long* cukup besar, sehingga untuk operasi perkalian dapat menghasilkan bilangan yang cukup besar pula. Penggalan *listing* program untuk perhitungan konversi oz ke *pounds* dapat dilihat di bawah ini.

```
public void oz_pounds() {
    long b = 625L;
    in1 = textField1.getString();
    in2 = textField2.getString();
    total = in1 + in2;
    long a = (Long.parseLong(total));
    long hasil = (b * a);
    long hslDpn = (b * a) / 100000000;
    textFied3.setString(Long.toString(hslDpn));
    .....}

```

Dari penggalan *listing* di atas, diketahui bahwa untuk perhitungan terlebih dahulu menentukan faktor konversinya, misalkan untuk konversi :

$$1 \text{ ons} = 0,0625 \text{ pounds}$$

maka untuk nilai 0,0625 disimpan sebagai bilangan integer 625, kemudian mengeset inputan yang terdiri atas 2 (dua) variabel integer yang masing-masing menyimpan nilai di sebelah kiri koma dan kanan koma menjadi sebuah string.

Untuk pengesetan awal di sebelah kiri koma (`in1`) = 1, dan di sebelah kanan koma (`in2`) = 0000 (empat digit angka 0). Empat digit ini yang menentukan banyaknya digit dibelakang koma yang harus dituliskan pada *teks field* koma.

Dari perhitungan konversi satuan yang telah dilakukan, hasil operasi perkalian tergantung pada faktor konversi yang digunakan terutama faktor konversi yang menggunakan bilangan pecahan, dimana bilangan dibelakang koma yang digunakan merupakan hasil pembulatan.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Aplikasi Konversi Satuan yang dibuat menggunakan bahasa pemrograman *Java 2 Micro*

Edition (J2ME) dengan CLDC yang support MIDP 1.0.

2. Aplikasi Konversi Satuan ini diaplikasikan pada ponsel Nokia 3650, untuk transfer data menggunakan konektivitas *bluetooth*.
3. Dalam aplikasi Konversi Satuan ini hanya menggunakan 7 (tujuh) macam besaran yaitu Daya, Energi, Kecepatan, Luas, Massa, Panjang, dan Volume,
4. Hasil konversi hanya menampilkan ketelitian sampai 4 (empat) digit dibelakang koma dan pada *text field* koma, terlebih dahulu harus meng-clear angka 0000 dan dalam *text field* ini harus terisi empat digit dikarenakan awal mengeset inputan yang terdiri atas 2 (dua) variabel integer yang masing-masing menyimpan nilai di sebelah kiri koma dan kanan koma dijadikan sebuah string, dimana nilai di sebelah kanan koma hanya menggunakan empat digit.

5.2 Saran

1. Kemunculan versi *Mobile Information Device Profile* (MIDP) 2.0 meningkatkan perkembangan aplikasi karena kelebihan fitur-fiturnya yang semakin futuristic, oleh karena itu aplikasi ini dapat dibuat lagi dengan menggunakan MIDP 2.0 atau dengan penggunaan kelas *Record Management System* (RMS), karena dengan RMS akan mudah melakukan aksi tambah, edit, dan hapus.
2. Untuk pengembangan lebih lanjut, untuk faktor konversi yang menggunakan bilangan pecahan sebaiknya tidak dilakukan pembulatan dan untuk hasil perhitungan sebaiknya tidak hanya menggunakan empat angka dibelakang koma, karena semakin banyak angka penting yang ditulis menunjukkan semakin tinggi derajat ketelitian perhitungan yang dilakukan.

DAFTAR PUSTAKA

- [1] Dharwiyanti, Wahono, *Pengantar Unified Modelling Language*, <http://www.ilmukomputer.com>, 2003.
- [2] Hartanto, Antonius Aditya, *Tip dan Trik Java 2 Micro Edition Mobile Interface Device Programming*, Elex Media Komputindo, Jakarta, 2003.
- [3] Hartanto, Antonius Aditya, *Tip dan Trik Java 2 Micro Edition Tingkat Lanjut*, Elex Media Komputindo, Jakarta, 2003.
- [4] Kanginan, Marthen, *Buku Pelajaran FISIKA SMA Jilid 1A*, Erlangga, Jakarta, 1993.

- [5] Madison, Joshua F., *Convert Version 4.08*, <http://www.joshmadison.com/software>, 1998.
- [6] Naughton, Patrick, *Konsep Dasar Pemrograman Java*, Andi dan McGraw-Hill Book Co., Yogyakarta, 2002.
- [7] Pressman, Roger S., *Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku II)*, Andi dan McGraw-Hill Book Co., Yogyakarta, 2002.
- [8] Purbo, Onno W., *Buku Pintar Internet Java dan JavaScript*, Elex Media Komputindo, Jakarta, 2000.
- [9] Suyoto, *Membuat Sendiri Aplikasi Ponsel*, Gava Media, Yogyakarta, 2005.
- [10] Tapson, Frank, *A Dictionary of Measures, Units and Conversions – Part 1 and Part 2*, <http://www.cleavebook.co.uk/index/htm>, 2004.
- [11] <http://www.developer.motorola.com>
- [12] <http://www.forumnokia.com>
- [13] <http://www.ilmukomputer.com>
- [14] <http://www.java.sun.com>



Sintha Saptiwiningrum
(L2F303480)
Mahasiswa Jurusan Teknik
Elektro Fakultas Teknik
Universitas Diponegoro dengan
Konsentrasi Informatika dan
Komputer.

Menyetujui dan Mengesahkan

Pembimbing I

Agung Budi Prasetyo, S.T., MIT.
NIP. 132 137 932
Tanggal

Pembimbing II

Aghus Sofwan, S.T., M.T.
NIP. 132 163 757
Tanggal