

**E – DAGANG MELALUI TELEFON MUDAH ALIH MENGGUNAKAN
JAVA 2 MICRO EDITION (J2ME)**

**Oleh
Siti Juliana Bte Safiei
Disertasi ini dikemukakan kepada
UNIVERSITI SAINS MALAYSIA**

**Sebagai memenuhi sebahagian daripada syarat keperluan
Untuk ijazah kepujian
SARJANA MUDA KEJURUTERAAN (KEJURUTERAAN ELEKTRONIK)**

**Pusat Pengajian Kejuruteraan
Elektri dan Elektronik
Universiti Sains Malaysia**

Mac 2005

ABSTRAK

Tidak dapat dinafikan bahawa telefon mudah alih menjadi salah satu keperluan manusia. Jika dahulu harga sesebuah telefon mudah alih ini menjangkau ribuan ringgit, namun pada masa kini pengguna boleh memperolehnya dengan harga ratusan sahaja. Bayangkan sekiranya kemudahan yang ditawarkan oleh telefon mudah alih ini dapat ditingkatkan. Salah satu aspek yang boleh diterokai ialah e-dagang. Seperti yang diketahui, e-dagang bermaksud melakukan urusaniaga dalam talian menerusi kemudahan Internet. Projek ini megimplementasikan e-dagang untuk pembelian tiket sistem perletakan kereta. Pengaturcaraan dilakukan menggunakan J2ME (Java 2 Micro Edition).

ABSTRACT

Handphone is now becoming one of the necessities of human life. Unlike years ago, the cost for handphone has significantly less as customer can get it for a mere few hundred dollars. What if the functionalities of the handphone can be expanded, for example, to support e-commerce. This project implements e-commerce support for buying parking ticket. Concerning implementation the project utilizes the JAVA 2 MICRO EDITION platform.

PENGHARGAAN

Alhamdulillah, segala kesyukuran kepada Ilahi, berkat kehendaknya, akhirnya saya dapat menghasilkan disertasi sebagai memenuhi keperluan pengijazahan dalam bidang Kejuruteraan Elektrik & Elektronik, Universiti Sains Malaysia. Ucapan jutaan terima kasih saya kepada Penyelia Dr. Kamal Zuhairi B Zamli, yang telah banyak memberi pertolongan, dorongan dan ilham dalam menyiapkan projek tahun akhir ini. Tidak lupa juga kepada pemeriksa kedua iaitu Dr. Harsa Amylia bte Mat Sakim yang telah memeriksa tesis ini.

Selain itu ucapan terima kasih ini juga ditujukan kepada saudari Noreen yang telah banyak membantu dan memberikan ilham yang bernas dalam menghadapi masalah kerja yang dilakukan, juga kepada saudari Faizah bte Idris yang telah banyak menolong dalam projek ini. Tidak lupa juga kepada saudari Faizah Hanim bte Che Seman yang telah membantu saya dalam menyiapkan tesis ini.

Tidak dilupakan kepada semua yang terlibat mahupun secara langsung atau tidak langsung dalam tempoh penghasilan projek ini. Sesungguhnya tanpa pertolongan dari anda semua, mustahil saya dapat menyiapkan projek ini dengan jaya sekali.

Sekian, Wassalam.

KANDUNGAN

Muka Surat

ABSTRAK.....	ii
ABSTRACT.....	iii
PENGHARGAAN.....	iv
JADUAL ISI KANDUNGAN.....	v

BAB 1: PENGENALAN

1.1 Pendahuluan.....	1
1.2 Tujuan.....	2
1.3 Matlamat Projek.....	2
1.4 Objektif Projek.....	2
1.5 Langkah-langkah Penyediaan Tesis.....	3
1.6 Struktur Laporan.....	5

BAB 2: KAJIAN ILMIAH

2.1 Pendahuluan.....	6
2.2 Penerangan mengenai <i>Java 2 Platform</i>	6
2.2.1 Penerangan mengenai <i>J2ME (Java 2 Micro Edition)</i>	7

2.3 Pembangunan Persekitaran Perkakasan tanpa Wayar (<i>Wireless Toolkit</i>).....	7
2.3.1 Pembangunan MIDlet – Perkakasan tanpa Wayar.....	8
2.3.2 Kitar Hayat MIDlet.....	12
2.3.3 Paparan <i>Stuff</i>	13
2.3.4 Bekerja dengan Skrin.....	15
2.3.5 <i>Form</i>	16
2.3.6 Menu dan Arahan.....	18
2.3.7 Mencipta skrin <i>Alert</i>	21
2.3.8 Kelas <i>Alert</i>	22

BAB 3: PERLAKSANAAN PROJEK

3.1 Pendahuluan.....	25
3.2 Penghasilan aturcara.....	25
3.3 Konsep pelaksanaan aturcara.....	26
3.4 Simulasi.....	27
3.5 Keputusan.....	28
3.6 Kesimpulan.....	38

BAB 4: PENUTUP

4.1 Pendahuluan.....	39
4.2 Masalah yang dihadapi.....	40

4.3 Cadangan untuk masa hadapan.....	41
4.4 Kesimpulan.....	42
RUJUKAN.....	44
LAMPIRAN.....	45

BAB 1

PENGENALAN

1.1 Pendahuluan

Pada masa kini, telefon mudah alih merupakan asas keperluan hidup manusia. Dari sehari ke sehari peningkatan penggunaan telefon mudah alih berkembang dengan pesatnya. Ia boleh didapati dengan pelbagai saiz dan rekabentuk yang menarik. Pelbagai jenama boleh didapati di pasaran. Antaranya seperti Nokia, Samsung , Motorola, Alcatel, Ericson dan sebagainya. Setiap jenama yang ada di pasaran mempunyai daya tarikannya yang tersendiri. Selaras dengan kemajuan dan kecanggihannya, setiap orang berebut-rebut untuk memilikinya. Dengan faktor harga yang murah dan dilengkapi dengan pelbagai fungsi-fungsi yang menarik menjadikan telefon mudah alih ini satu alat telekomunikasi yang meluas digunakan. Telefon mudah alih ini memberi seribu satu manfaat kepada pengguna yang memilikinya.

Walaupun alat telekomunikasi yang dikatakan selari dengan kemajuan kehidupan ini dapat memberi kemudahan yang membanggakan kepada pemiliknya namun kegunaannya masih terhad. Selaras itu, peningkatan terhadap fungsi telefon mudah alih ini harus ditingkatkan. Projek yang dibangunkan ini mengimplementasikan perkhidmatan e-dagang. Secara umumnya e-dagang membawa maksud melakukan urusanniaga dalam talian menerusi kemudahan Internet. Umpamanya penggunaan kaedah EDI (Electronic Data Interchange) membolehkan pekerja sesebuah syarikat membuat tempahan dan pembelian barangan menerusi komputer syarikat pembekal.

1.2 Tujuan Projek

Tujuan projek ini adalah untuk melaksanakan peningkatan terhadap kemudahan penggunaan telefon mudah alih menggunakan perkhidmatan e-dagang di samping memperkenalkan fungsi-fungsi yang baru. Perkhidmatan e-dagang menawarkan pelbagai kemudahan pembayaran dan pembelian kepada pengguna. Antaranya pengguna boleh membuat pembayaran bil-bil menerusi kemudahan internet yang disediakan.

1.3 Matlamat Projek

- ✚ Matlamat projek ialah untuk membangunkan aplikasi untuk telefon mudah alih menggunakan bahasa JAVA.

1.4 Objektif Projek

- ✚ Mempelajari subset bahasa JAVA iaitu *Java 2 Micro Edition (J2ME)*.
- ✚ Mengimplementasikan sistem pembelian tiket menggunakan *Java 2 Micro Edition (J2ME)*.
- ✚ Mempelajari pelbagai kaedah dan cara untuk melaksanakan sesebuah projek seperti mengumpul data, membuat keputusan, penulisan tesis dan sebagainya.

1.5 Langkah-langkah Penyelidikan Tesis

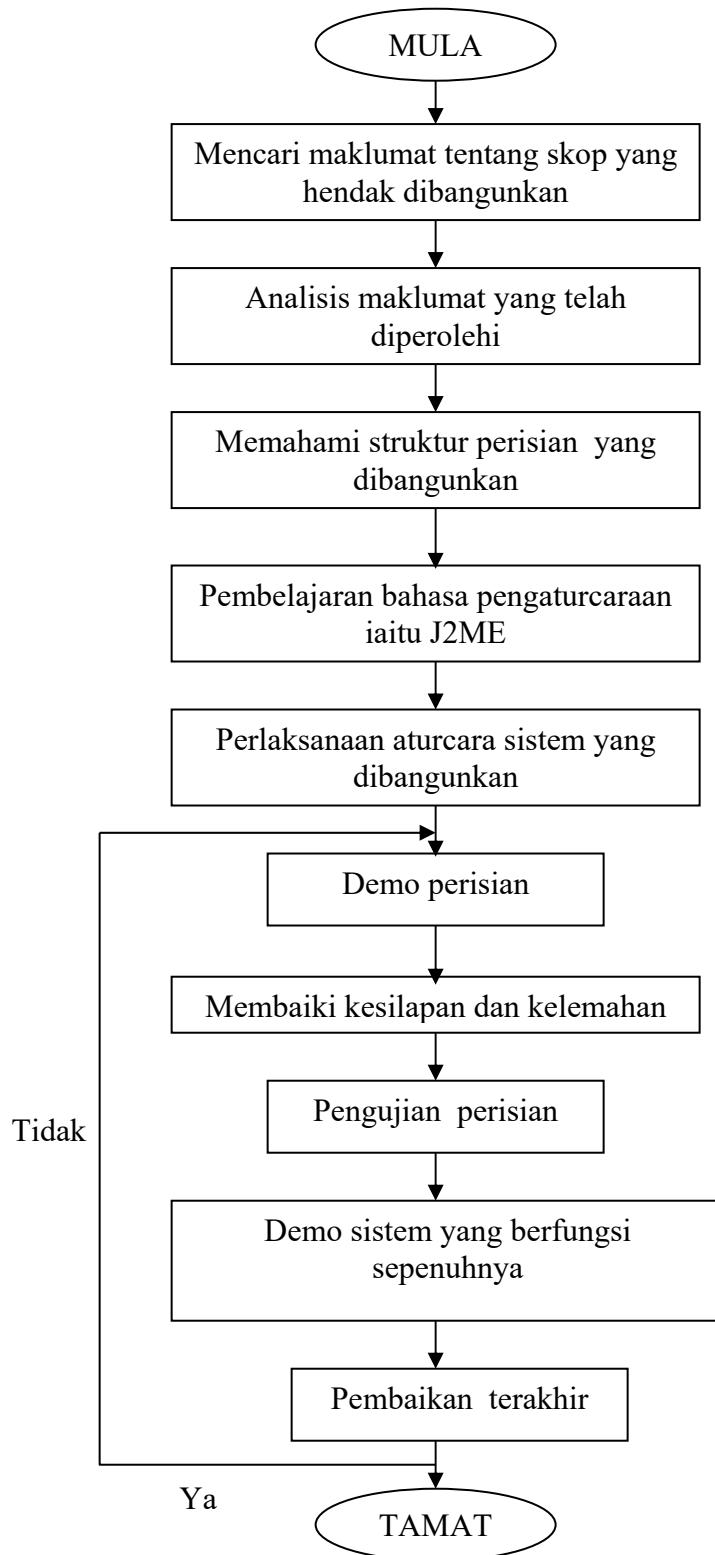
Untuk mencapai objektif projek tahun akhir ini, satu rangka kerja telah dibentuk untuk memudahkan proses penulisan tesis. Carta alir **Rajah 1.1** di bawah menunjukkan aktiviti - aktiviti yang telah dijalankan untuk menjayakan tesis ini.

Sebelum sistem perisian ini dilaksanakan, pelbagai maklumat dikumpulkan untuk mengenalpasti kehendak dan keperluan pengguna akan peningkatan yang dilakukan terhadap sistem tersebut. Sistem yang dibangunkan perlu memenuhi spesifikasi yang ditetapkan agar memudahkan pengguna untuk menggunakan perkhidmatan yang ditawarkan.

Bahasa pengaturcaraan yang digunakan iaitu *Java 2 Micro Edition (J2ME)* hendaklah dipelajari bagaimana ia berfungsi. Arahan-arahan penting bagi bahasa pengaturcaraan yang digunakan turut dipelajari dalam membangunkan sistem perisian.

Sistem perisian yang telah siap dibina akan diuji daripada segi pengoperasiannya samada setiap fungsi di dalam sistem tersebut berfungsi ataupun tidak.

Rajah 1.1 : Langkah-langkah Pelaksanaan Tesis.



1.6 Struktur Laporan

Secara keseluruhannya, **Bab 1** menerangkan objektif tesis serta langkah-langkah yang telah diambil untuk menjayakan tesis ini. Skop tesis ini adalah untuk membina perisian yang dapat meningkatkan fungsi penggunaan telefon mudah alih pada masa kini menggunakan bahasa pengaturcaraan *Java 2 Micro Edition (J2ME)*.

Bab 2 akan menerangkan mengenai bahasa pengaturcaraan yang digunakan iaitu *Java 2 Micro Edition (J2ME)*. Arahan-arahan bagi bahasa pengaturcaraan ini dipelajari dan difahami dengan jelas agar pelaksanaan projek dapat dilaksanakan.

Bab 3 pula menerangkan bagaimana keseluruhan projek ini dapat dibangunkan. Di dalam bab ini penerangan mengenai penghasilan aturcara program yang digunakan serta konsep pelaksanaan projek akan diterangkan di samping memperihalkan keputusan yang telah diperolehi. Di sini pengguna dapat melihat dan membuat pembelian tiket dengan menggunakan datagram sebagai medium penghantaran.

Akhir sekali di dalam **Bab 4** ini, penutup akan mengakhiri laporan ini. Dalam bab ini perbincangan mengenai masalah yang dihadapi semasa sistem ini disiapkan serta beberapa cadangan untuk meningkatkan perisian ini akan dicadangkan agar sistem ini dapat digunakan dengan lebih berkesan dan jayanya pada masa akan datang.

BAB 2

KAJIAN ILMIAH

2.1 Pendahuluan

Bab 2 ini menerangkan tentang bahasa pengaturcaraan dan arahan-arahan yang digunakan untuk melaksanakan projek ini. Bahasa pengaturcaraan yang digunakan ini mudah untuk difahami dan diaplikasikan. Pengguna dapat mengaplikasikan perisian ini dengan cepat dan mudah.

2.2 Penerangan mengenai *Java 2 Platform*

Java 2 Platform terbahagi kepada tiga edisi iaitu *Java 2 Standard Edition (J2SE)*, *Java 2 Enterprise Edition (J2EE)*, dan *Java 2 Micro Edition (J2ME)*. Setiap bahagian dilengkapi *platform* dengan persekitaran yang lengkap untuk melaksanakan aplikasi *Java* dan *Java Virtual Machine (VM)*.

Ketiga-tiga edisi ini mempunyai sasaran aplikasi yang berbeza berdasarkan kepada peranti yang berbeza. Aplikasi berdasarkan meja dibangunkan menggunakan edisi J2SE di mana ia mengandungi keperluan kelas antaramuka pengguna. Aplikasi berdasarkan pelanggan pula dibangunkan menggunakan edisi J2EE di mana ia menekankan perisian berdasarkan komponen dan pembahagian. Pegang tangan dan peranti terbenam adalah sasaran bagi edisi J2ME.

Perbezaan antara satu edisi ini dengan edisi yang lain ialah takrifan kandungan kelas koleksi setiap edisi ini. Seperti yang dijangkakan J2ME adalah subset kepada J2SE dan J2SE adalah subset kepada J2EE. Ini adalah mustahil untuk menjalankan *Java bytecode* yang sama pada setiap edisi asalkan kelas dirujuk kepada *bytecode* yang boleh digunakan

untuk ketiga-tiga edisi ini. Ini kerana peranti berdasarkan J2ME mempunyai beberapa kelas berbanding dengan kandungan J2SE dan J2EE terutamanya untuk peranti yang kecil. (<http://www.java.sun.com/products/j2mewtoolkit>)

2.2.1 Penerangan mengenai *Java 2 Micro Edition (J2ME)*

Dalam J2ME, persekitaran Java adalah pepadanan untuk menghadkan peranti di mana setiap peranti mempunyai had-had yang boleh dilakukannya apabila ia dibandingkan dengan *desktop* atau pelanggan komputer. Untuk peranti yang tamat rendah penghad adalah lebih ketara, memori terlalu kecil, size skrin kecil, pilihan kaedah input, dan proses yang perlahan. Bagi peranti yang tamat tinggi, ia mempunyai beberapa penghad tetapi ia masih memberi kebaikan daripada persekitaran dan juga antaramuka aturcara yang baru mengikut takrifan J2ME.

2.3 Pembangunan Persekitaran – Perkakasan tanpa Wayar (*Wireless Toolkit*)

Wireless Toolkit yang dibangunkan oleh SUN adalah percuma, lengkap dan mudah digunakan dan boleh didapati untuk *Windows*, *Linux*, dan *Solaris*. Ianya juga mempunyai beberapa kod sumber (*source code*) termasuk contoh-contoh permainan. Selain itu, ia juga mengandungi *emulator* yang membolehkan pengguna melakukan pengujian terhadap aplikasi yang dibangunkan pada telefon yang tidak berwarna, telefon berwarna dan sebagainya.

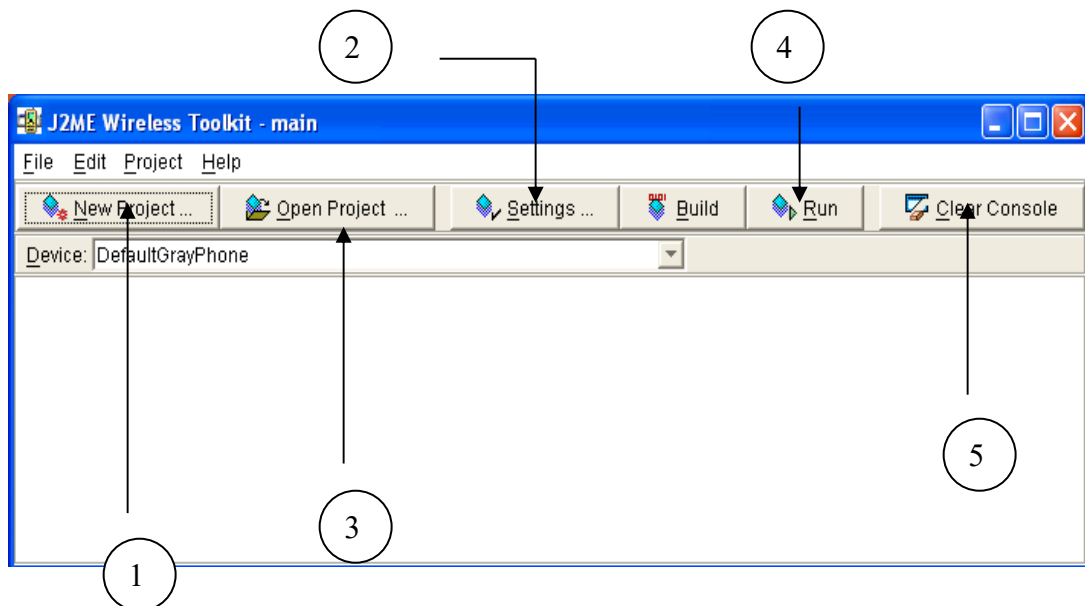
Wireless Toolkit juga boleh digabungkan ke dalam perisian *Borlands' J Builder*.



Rajah 2.1: jenis-jenis telefon mudah alih

2.3.1 Pembangunan MIDLET menggunakan Perkakasan tanpa Wayar

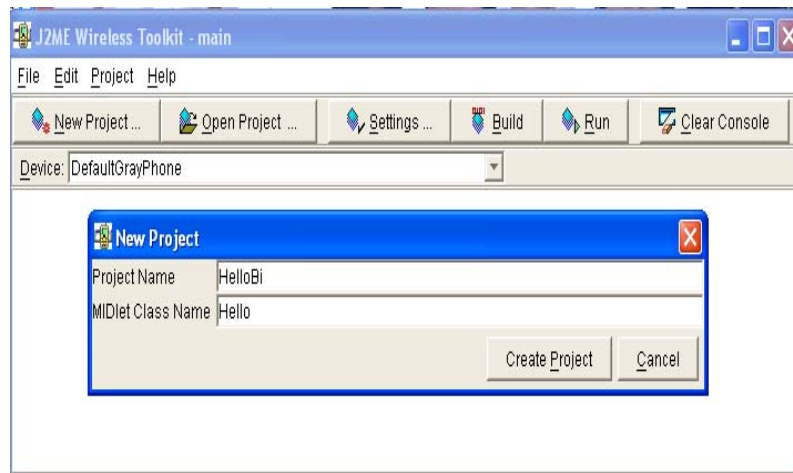
- ✚ Program Perkakasan tanpa Wayar (*Wireless Toolkit*) dijalankan untuk menghasilkan proksi langganan dan juga saiz longgokan (*heap size*) untuk dicontohi.
- ✚ Program Bar perkakasan K (*K Toolbar*) yang memuat naik satu menu grafik dengan segala pilihan pembangunan dijalankan.



Di bawah adalah senarai komponen utama yang terdapat dalam Bar Perkakasan:

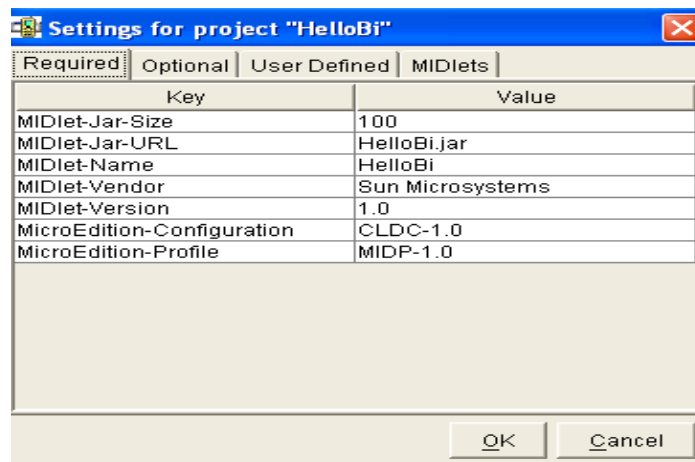
1. *New project* – butang ini akan digunakan untuk menghasilkan satu projek yang baru.
2. *Setting* - aturan akan memaparkan maklumat yang dihasilkan oleh sesuatu projek baru.
3. *Build* – Penyusunan, penggabungan dan pra pengesahan dilaksanakan terhadap aplikasi yang dibangunkan ke dalam satu langkah. Ini dilakukan dengan memilih butang *Build*.
4. *Run* - Untuk menjalankan *emulator*, butang *run* ini akan dipilih.
5. *Console* – Jika tiada kesalahan yang didapati, konsol akan memaparkan mesej *Build* lengkap (*Build Complete*). Jika terdapat kesalahan pada aplikasi, kesalahan akan dipaparkan. Kesalahan ini boleh dibaiki dengan kembali kepada fail java.

✚ Butang *new project* dipilih untuk menghasilkan projek yang baru. Nama yang ringkas tanpa ruangan diberikan kepada projek yang ingin dibangunkan dan berikan juga nama pada kelas MIDlet yang utama. Contohnya, jika anda ingin menghasilkan satu aplikasi HelloBi, berikan nama projek HelloBi dan Hello pada kelas MIDlet seperti **Rajah 2.2**.



Rajah 2.2

- ✚ Penghasilan langkah di atas ditunjukkan dalam **Rajah 2.3**. Pembentukan nama projek yang ingin dibangunkan boleh di ubah mengikut keperluan.



Rajah 2.3

Butang ok dipilih untuk menyimpan setiap aturan yang dibuat. Setiap aturan yang telah dibuat boleh diubah pada bila-bila masa dengan memilih butang aturan.

- ✚ Satu direktori baru akan dihasilkan di bawah direktori *C: \J2mewtk\apps*. Untuk kes ini, direktori yang terhasil adalah *C: \J2mewtk\apps\HelloBi*. Di bawah direktori tersebut terdapat 4 direktori lain yang terhasil iaitu *bin*, *lib*, *res* dan *src*.
- ✚ Jika aplikasi yang dibangunkan PNG imej atau sumber lain, masukkan semua ke dalam direktori *C: \J2mewtk\apps\HelloBi\res*.
- ✚ Penyusunan, penggabungan dan pra pengesahan dilaksanakan terhadap aplikasi yang dibangunkan ke dalam satu langkah. Ini dilakukan dengan memilih butang *Build*.
- ✚ Jika tiada kesalahan yang didapati, konsol akan memaparkan mesej *Build* lengkap (*Build Complete*). Jika terdapat kesalahan pada aplikasi, kesalahan akan dipaparkan. Kesalahan ini boleh dibaiki dengan kembali kepada fail java.
- ✚ Pengujian boleh dilakukan menggunakan *built-in emulator*. Pilih peranti yang hendak diimulasi dengan menggunakan menu tarik bawah (*pull down menu*).
- ✚ Untuk menjalankan *emulator*, butang *run* akan dipilih. semua MIDlet yang boleh didapati disenaraikan untuk dilancarkan akan ter papar. Pilih HelloBi dan butang *lauch* dipilih. Aplikasi ini akan dijalankan seperti **Rajah 2.4**.



Rajah 2.4: HelloBi MIDlet

2.3.2 Kitar Hayat MIDlet

Java terbahagi kepada dua bahagian utama iaitu *applet* dan *servelet*. J2ME (*Java 2 Micro Edition*) mempunyai satu aplikasi yang dikenali sebagai MIDlet.

MIDlet – sejenis *applet* yang menggunakan peranti MIDP untuk berfungsi. Kelas MIDlet mengandungi segala keperluan yang diperlukan oleh pengurusan aplikasi telefon untuk dimasukkan ke dalam aplikasi yang dibangunkan. Proses *start*, *pause*, dan *end* dimulakan.

✚ *public void startApp ()* – kaedah ini dipanggil sejurus selepas MIDlet menjadi aktif. Ketika aktif, MIDlet mungkin akan *run* untuk mencapai sumber telefon. Kaedah ini dipanggil apabila program dimulakan dan juga ketika program dilepaskan dari keadaan berhenti seketika (*pause state*).

✚ *public void pauseApp ()* – kebanyakan peranti MIDP membenarkan pengguna untuk berhenti seketika aplikasi yang sedang dijalankan. Contohnya, jika telefon pengguna berdering, ia akan memilih keadaan *pause* untuk menjawab panggilan. Apabila *pause* digunakan, kaedah ini dipanggil. Dalam keadaan *pause*, MIDlet akan melepaskan sumber yang dikongsi. Kaedah ini hanya dipanggil apabila MIDlet berada di dalam status aktif. Sesetengah telefon mudah alih seperti SIEMENS LS45i tidak menggunakan kaedah ini. Walaubagaimanapun kaedah ini perlu dimasukkan ke dalam perisian yang dibuat.

✚ *public void destroyApp ()* – apabila pengguna memilih untuk menghentikan servis membeli tiket perletakan kereta, telefon mudah alih tersebut akan secara automatik akan memanggil kaedah ini. Pada masa ini, MIDlet akan

menghapuskan dan memasuki keadaan *destroyed*. Di dalam keadaan ini, MIDlet perlu melepaskan semua sumber-sumber, menghentikan semua gegelung dan menyimpan dalam keadaan yang berterusan. Kaedah ini boleh dipanggil semasa MIDlet berada dalam keadaan *pause* atau aktif.

2.3.3 Paparan *Stuff*

Untuk melukis sesuatu skrin peranti mikro (*micro device screen*), pencapaian terhadap paparan perlu dicapai. Untuk memperolehi pencapaian ini, kelas *javax.microedition.lcdui.Display* yang terdapat dalam MIDlet boleh digunakan.

Paparan mewakili paparan grafik sistem dan input peranti. Ia termasuk kaedah pembetulan semula dan memaparkan objek pada peranti. Contohnya ia terdapat dalam kelas paparan per MIDlet. Untuk mendapatkan contoh rujukan, kaedah *getDisplay* dipanggil. Aplikasi boleh memanggil kaedah tersebut daripada mana-mana sumber di dalam kod pada bila-bila masa.

Setiap objek antaramuka pengguna yang boleh dihasilkan pada paparan mesti disambungkan dengan kelas *Displayable*. Objek *Displayable* mempunyai arahan yang berkaitan dengannya. Kandungan yang dipaparkan dan hubungan dengan pengguna adalah didefinisikan dalam sub kelas. Secara amnya, setiap skrin yang hendak dipaparkan perlulah didefinisikan sebagai objek *Displayable* yang berasingan.

Peranti hanya boleh menunjukkan satu objek *Displayable* pada satu masa. Objek tersebut dirujuk sebagai objek *currentDisplayable*. Terdapat beberapa kelas *Displayable* yang utama.

- ✚ *Is Color ()* – mengembalikan warna asal jika peranti tersebut mempunyai skrin yang berwarna dan sebaliknya.
- ✚ *numColors ()* – mengeluarkan bilangan warna yang boleh digambarkan pada peranti.
- ✚ *getCurrent ()*– mengembalikan objek *Displayable*. Semasa objek *Displayable* dikembalikan kemungkinan objek tidak dapat dilihat jika MIDlet dilaksanakan dalam latarbelakang yang berbeza atau jika objek *Displayable* dihalang oleh skrin sistem. Nilai yang dikembalikan *get Current* akan menjadi sifar jika kaedah ini tidak dipanggil.
- ✚ *setCurrent (Displayable nextDisplayable)* – kaedah ini akan menukarkan paparan semasa. Masukkan sahaja objek *Displayable* yang baru dan pertukaran tersebut tidak akan bertukar dengan serta-merta malah akan dilambatkan supaya ianya terjadi di antara kaedah penghantaran panggilan. Kerana kelewatan ini satu panggilan *getCurrent* seurus selepas *setCurrent* tidak akan dapat mengembalikan nilai yang disalurkan kepada *setCurrent*.

~ Apabila satu aplikasi MIDlet dimulakan, tiada objek *Displayable* semasa. Ianya adalah tanggungjawab aplikasi tersebut untuk memastikan objek *Displayable* dapat dilihat dan boleh berinteraksi dengan pengguna pada setiap masa. Oleh itu, satu aplikasi perlu sentiasa memanggil *setCurrent* sebagai bahagian permulaan.

- ✚ *setCurrent (Alert alert, Displayable next Displayable)*- ini ialah versi *setCurrent* yang dihasilkan khas untuk *alert*. *Alert* ialah suatu jenis objek *Displayable* yang harus ditunjukkan dan dihapuskan dengan serta merta. Contohnya, *alert* akan

memberi amaran kepada pengguna mengenai sesuatu masalah. Untuk menggunakan kaedah ini, aturkan *alert* sebagai objek *Displayable* dan skrin *nextDisplayable* dipilih. Sejurus selepas kotak *alert* dihapuskan, skrin *nextDisplayable* akan dipaparkan.

2.3.4 Bekerja dengan Skrin

Kelas *javax.microedition.lcdui.Screen* ialah kelas yang lazimnya digunakan untuk semua objek antaramuka pada aras tinggi. Skrinnya adalah sangat mudah. Ianya membenarkan pengguna untuk menambah tajuk atau *scroll* penanda kepada kelas *Displayable*. Setiap skrin yang akan dilihat oleh pengguna adalah direka dari kelasnya sendiri. Paparan tersebut akan secara automatik diubah apabila kandungan skrin dikemaskinikan. Aplikasi tersebut tidak perlu memanggil kaedah lain untuk mengubah paparan tersebut.

Ia adalah satu latihan perisian yang baik untuk menukar kandungan sesuatu skrin apabila ianya tidak dilihat. Penukaran pada kandungan pada skrin yang tidak boleh dilihat, boleh mendatangkan masalah pada sesetengah peranti. Ia juga boleh mengelirukan pengguna, jika kandungan skrin bertukar pada masa yang sama dimana pengguna sedang berinteraksi dengannya.

Terdapat beberapa sub kelas di bawah kelas skrin iaitu:

 *Form*

 *Alert*

 *List*

 *TextBox*

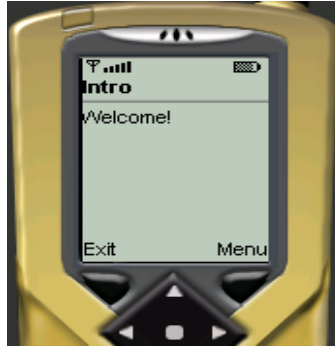
Terdapat 4 kaedah daripada skrin:

- ✚ *getTitle* – mendapatkan tajuk bagi skrin
- ✚ *getTitle (String title)* – menentukan tajuk bagi skrin. Jika skrin tersebut tidak boleh dilihat secara fizikal, kesan nyata perlu mengambil alih tanpa lengah daripada sejurus selepas panggilan balik.
- ✚ *getTicker* – mendapatkan *scroll* penanda objek yang digunakan oleh skrin.
- ✚ *setTicker(Ticker ticker)*- aturkan penanda *scroll* untuk digunakan pada skrin untuk menggantikan penanda yang sebelumnya. Beberapa objek skrin di dalam sesuatu aplikasi boleh berkongsi penanda yang sama. Ini dilakukan dengan memanggil *setTicker* pada skrin yang berbeza dengan penanda objek yang sama.

2.3.5 Form

Kelas *javax.microedition.lcdui.Form* ialah yang paling lazim digunakan oleh skrin. Ia mungkin mengandungi campuran *arbitrary* seperti imej, lapangan teks, lapangan tarikh dan pilihan kumpulan. Secara amnya sub kelas kepada kelas item boleh didapati dalam satu *form*.

Setiap telefon mudah alih akan mengawal aturan bahagian-bahagian mengikut perancangan. Item yang terdapat di dalam *form* adalah rujukan indeks iaitu *interger* berturutan yang bermula daripada sifar. Item–item ini boleh diubah dengan menggunakan kaedah *append*, *delete*, *insert*, dan *set*.



Rajah 2.5: Pengenalan Skrin

Form baru perlu disambungkan kepada kelas *form*. Pembina *form* mesti memanggil pembina utama yang menerima tajuk sebagai parameter. Untuk mencapai *MIDlet's exit* dan kaedah-kaedah yang lain, global boleh ubah yang merujuk kepada kelas utama dikhususkan.

`javax.microedition.lcdui.StringItem` ialah satu komponen antaramuka pengguna untuk mempersembahkan label teks baca sahaja yang berbeza. Setiap *form* mempunyai kaedah utama seperti berikut:

- + `append (Item item)` – menambah item antaramuka pengguna kepada *form*.
- + `Append (String str)` – menambah string kepada *form*
- + `Append (Image img)` – menambah imej kepada *form*
- + `Insert (int itemNum, Item item)` – menambah satu item kepada *form* sejurus sistem nombor item tersebut dikhususkan. Saiz *form* semakin meningkat. Parameter *itemNum* mestilah berada dalam kadar semasa item yang sebenar.
- + `Delete(int itemNUM)` - memadam item dalam *itemNum*.