

BAB II LANDASAN TEORI

2.1 Pendahuluan

Teknologi *mobile* terus berkembang dari waktu ke waktu. Dahulu konektivitas antar *mobile phone* atau koneksi dengan PC hanya bisa dilakukan secara langsung dengan kabel, tetapi sekarang koneksi yang didukung sudah semakin berkembang antara lain dengan *infrared*, *Bluetooth*, GPRS dan masih banyak lagi.

Sekarang ini banyak muncul aplikasi-aplikasi *chatting* yang berjalan pada *mobile phone* dengan menggunakan berbagai macam koneksi. Mulai dari Yahoo Messenger yang dahulu berbasis desktop, kini sudah mulai membuat aplikasi *chatting* yang dapat berjalan pada *mobile phone* dan masih banyak perangkat-perangkat lunak serupa.

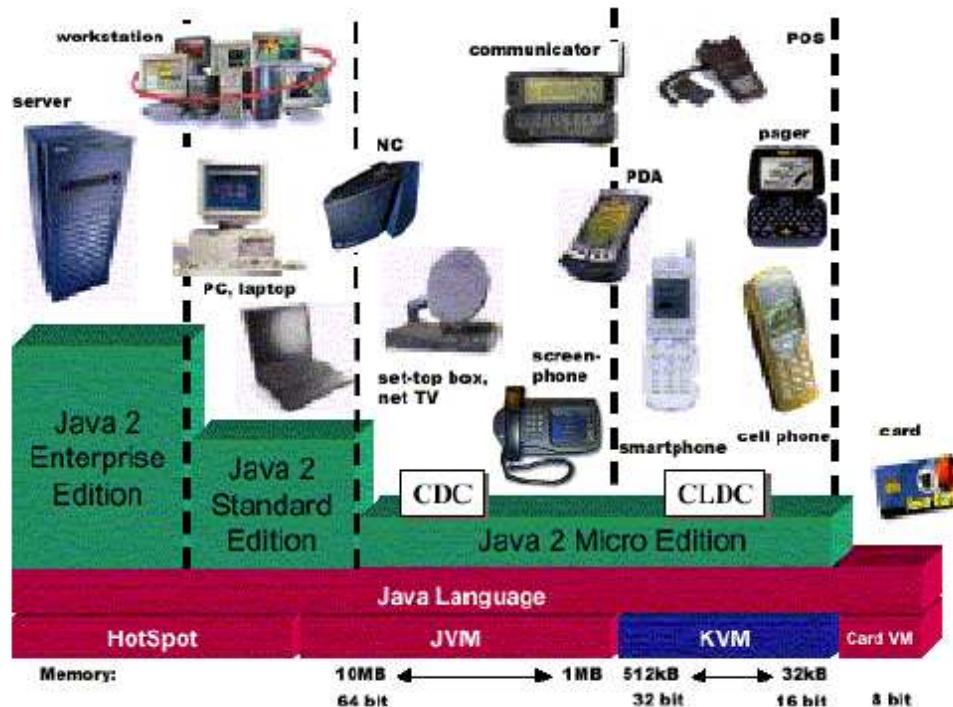
Java merupakan bahasa pemrograman yang dikembangkan oleh Sun Microsystems dan dirancang sedemikian rupa agar bersifat *portable* atau dapat berjalan pada semua *platform*. Konsep *write once run everywhere* (tuliskan sekali, jalan dimanapun) membuat banyak pengembang aplikasi *mobile* mengadopsi Java. Sun Java menyusun bingkai kerja pengembangan aplikasi *mobile* ke dalam J2ME (*Java 2 Micro Edition*).

2.2 Java

Bahasa Java dikembangkan oleh Sun Microsystems sejak tahun 1995 dengan versi pertamanya *Java 1.0* dan mulai diluncurkan ke publik pada tahun 1996 bersamaan dengan peluncuran *Netscape Navigator 2.0* yang mendukung *Java applet*. Setelah beberapa perbaikan akhirnya Sun meluncurkan *Java 1.02* yang oleh masyarakat luas dikenal

dengan Java 1.0. Pada Desember 1998 Sun meluncurkan *Java 2 Platform* yang lebih dikenal dengan *Java Development Kit* (JDK) 1.2 dan saat ini telah diluncurkan versi terbaru yaitu versi 1.6.0.06.

Selama pengembangannya, *Java 2 Platform* dibagi menjadi 4 komponen yang dapat digambarkan seperti pada gambar 2.0.



Gambar 2.0 Diagram Platform Java (Sun Microsystems)

Berikut penjelasan dari masing-masing komponen tersebut (Agung, 2005):

1. J2EE (*Java 2 Enterprise Edition*), platform ini berupa paket yang berisi *classes* dan *interface* yang digunakan untuk menjalankan dan mengembangkan aplikasi Java berbasis *web*, seperti *classes Servlet*, *Java Server Pages (JSP)* dan *Enterprise Java Beans (EJB)* serta *Java CORBA*.

2. J2SE (*Java 2 Standard Edition*), platform digunakan untuk menjalankan dan mengembangkan aplikasi *Java* pada level *Personal Computer* (PC). Platform ini berisi *classes* inti pada *Java* dan *Graphical User Interface* (GUI).
3. J2ME (*Java 2 Micro Edition*), platform ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi *Java* (Midlet) pada *handheld devices* atau perangkat-perangkat semacam *handphone*, *Personal Digital Assistance* (PDA) dan *Pocket PC*.

JavaCard, merupakan perangkat keras untuk membangun aplikasi pada sebuah *card electronic* seperti *SIM Card* pada telepon genggam, aplikasi kartu telepon CHIP, kartu VISA, dan aplikasi *mobile banking*.

2.3 J2ME

Java 2 Micro Edition (J2ME) digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi *Java* pada perangkat semacam telepon genggam, PDA (*Personal Digital Assistance*), *Palm*, dan *Pocket PC*. Karena adanya J2ME, memungkinkan bagi para pengembang untuk bisa membuat aplikasi *wireless* yang *multiplatform*, yang dapat diimplementasikan pada berbagai merek telepon genggam, yang mendukung aplikasi *Java*. Saat ini terdapat dua jenis aplikasi dari J2ME (Hartanto, 2003), yaitu:

1. **Walled garden application**, yaitu aplikasi yang berdiri sendiri atau *stand-alone* yang berjalan pada *handphone* tanpa perlu mengakses sumber data eksternal melalui jaringan pembawa atau *carrier*

network. Contoh dari aplikasi ini adalah kalkulator atau *single player games*.

2. **Network aware application** atau aplikasi yang berinteraksi dengan jaringan. Tidak seperti aplikasi yang pertama, aplikasi ini memiliki kemampuan untuk mengakses sumber data eksternal. Contoh dari aplikasi jenis ini adalah aplikasi *e-mail* yang berada di dalam handphone.

Komponen-komponen J2ME terdiri dari *Java Virtual Machine* (JVM) yang digunakan untuk menjalankan aplikasi *Java* pada emulator atau *handheld device*, *Java API* (*Application Programming Interface*) dan tools lain untuk pengembangan aplikasi *Java* semacam emulator *Java Phone*, emulator *Motorolla* dari *J2ME Wireless Toolkit*. Dalam pengembangan aplikasi *wireless* dengan *Java*, J2ME dibagi menjadi dua buah bagian, yaitu:

1. **Lapisan Konfigurasi (*Configuration Layer*)**

Virtual machine yang berupa konfigurasi yang menyediakan fungsi dasar dengan karakteristik yang sama (Mardiono, 2006). Contohnya: fungsi koneksi jaringan dan manajemen memori. J2ME mempunyai dua konfigurasi utama (Agung, 2005) yaitu:

- a. **Connected Limited Device Configuration (CLDC)**

CLDC diperlukan untuk pengembangan aplikasi *wireless* dengan MIDP. Implementasinya CLDC digunakan untuk program *Java* pada perangkat keras dengan ukuran memori yang terbatas, pada 160 sampai dengan 512 *Kilobyte*.

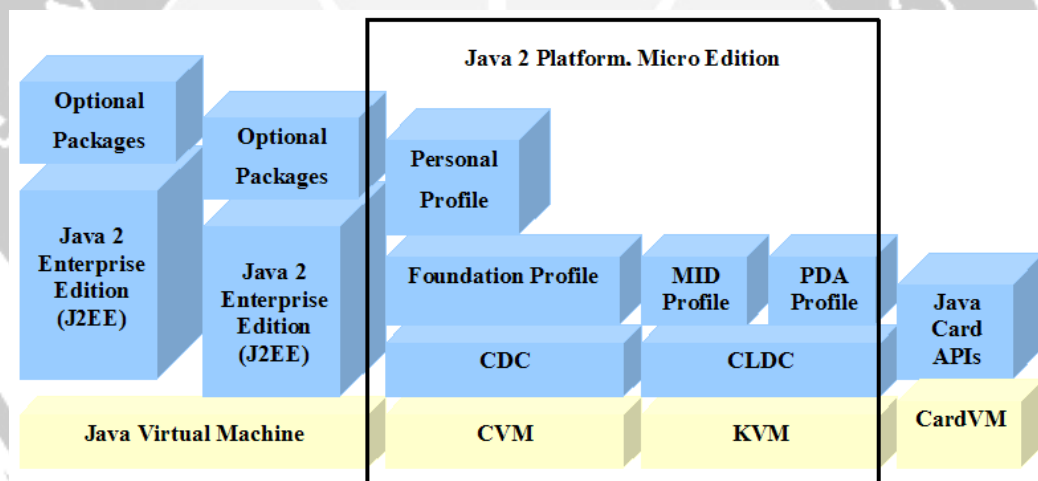
b. **Connected Device Configuration (CDC)**

CDC mampu menggunakan seluruh fitur Java 2 *Virtual Machine*. CDC juga mampu menggunakan hampir seluruh fitur J2SE. Besarnya permintaan CDC akan sumber daya, membuat CDC tidak bisa digunakan di banyak piranti *mobile*.

2. Lapisan Profile (Profile Layer)

Menyediakan lingkungan pustaka-pustaka API untuk membangun aplikasi *mobile* (Mardiono, 2006). J2ME mempunyai beberapa profil (Agung, 2005) antara lain:

- a. *Mobile Information Device Profile (MIDP)*
- b. *Foundation Profile (FP)*
- c. *Personal Profile*
- d. *Personal Digital Assistance (PDA) Profile*



Gambar 2.1 Java 2 Micro Edition (Sun Microsystem)

Dilihat dari Gambar 2.1 di atas, maka J2ME (Java 2 *Micro Edition*) mempunyai lapisan konfigurasi dan *profile* yang didukung oleh Java *Virtual Machine* (*CVirtual Machine* dan *K-Virtual Machine*).

2.4 Aplikasi J2ME

Aplikasi yang dibuat di dalam *handphone* dengan menggunakan profil MIDP disebut dengan MIDlet. Sun Microsystems telah menyediakan J2ME *Wireless Toolkit* (J2ME WTK) untuk mengembangkan aplikasi dalam *handphone*. J2ME WTK adalah sekumpulan *tool* yang digunakan untuk mengembangkan aplikasi-aplikasi dalam *handphone* dan *wireless device* lainnya.

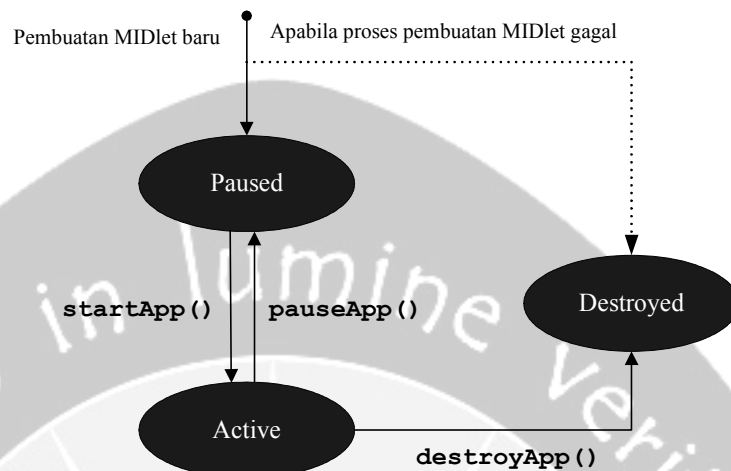
Developer dimungkinkan untuk membangun aplikasi *mobile* yang dapat terkoneksi dengan *internet* (layanan GPRS) maupun dengan *mobile device file system*.

Dalam membangun suatu MIDlet, J2ME memiliki *packages* `javax.microedition.io`, yaitu *class* untuk mendefinisikan koneksi-koneksi yang bersifat umum. *Class* ini terbagi atas tiga bagian, yaitu *Interfaces*, *Classes*, dan *Exception*. Jenis koneksi yang digunakan dalam tugas akhir ini adalah *HttpConnection* yang merupakan bagian dari *Interfaces*. *HttpConnection* berfungsi untuk memanfaatkan teknologi jaringan ponsel.

1. Siklus Hidup Aplikasi J2ME

AMS (*Application Management Software*) atau JAM (*Java Application Manager*) merupakan lingkungan tempat sebuah MIDlet dapat di-*install*, dijalankan, dihentikan, maupun di-*uninstall*. AMS akan membuat setiap *instance* baru dari MIDlet yang dapat mengontrol keadaannya, yaitu dengan cara menjalankannya (*start*), mengistirahatkan (*pause*), atau menghentikannya (*destroy*) secara langsung oleh dirinya sendiri. Terdapat tiga buah *method* yang harus diimplementasi oleh setiap MIDlet. Dengan kata lain, setiap MIDlet harus memiliki ketiga buah

method tersebut, yaitu: *startApp()*, *pauseApp()*, dan *destroyApp()*. Setiap MIDlet dapat berada dalam salah satu keadaan (*state*) berikut: *Paused*, *Active*, atau *Destroyed* (Raharjo, Hermantyo, Haryono, 2007).



Gambar 2.2. Siklus hidup MIDlet

Tampak pada gambar di atas bahwa pada saat pembuatan MIDlet baru, mula-mula MIDlet akan berada dalam keadaan *Paused*. Apabila proses pembuatan MIDlet gagal atau mengakibatkan kesalahan (menimbulkan eksepsi), maka MIDlet akan langsung berada dalam keadaan *Destroyed*. Namun apabila proses pembuatan MIDlet berjalan dengan baik, maka setelah MIDlet dijalankan AMS secara otomatis akan mengeksekusi *method startApp()* dan hal ini akan mengubah MIDlet untuk berada dalam keadaan *Active*. MIDlet yang berada dalam keadaan *Active* dapat diubah kembali menjadi keadaan *Paused* melalui pemanggilan *method pauseApp()* atau diubah menjadi keadaan *Destroyed* melalui pemanggilan *method destroyApp()*.

2.5 `HttpConnection`

`HttpConnection` merupakan salah satu spesifikasi J2ME yang digunakan pada aplikasi untuk mengakses *internet* dengan menggunakan teknologi jaringan yang tersedia pada ponsel.

HTTP merupakan salah satu cara untuk melakukan koneksi antara ponsel dan server yang disediakan oleh J2ME, disamping koneksi lainnya seperti *socket*, *SMS*, dsb. Koneksi jenis ini menjadi salah satu pilihan yang sering digunakan karena area penggunaannya luas dan tidak sedikit ponsel yang mendukung koneksi HTTP ini.

Menurut *J2ME Foundation Specification*, HTTP adalah sebuah protokol *request/response*, dimana parameter *request* harus diset terlebih dahulu sebelum *request* tersebut terkirim. Dalam metode ini, *client* mengirimkan *request* kepada *server* dengan alamat yang dispesifikasikan pada URL (*Uniform Resource Locator*), kemudian *server* akan memberikan *response* kepada *client*. Metode *request* itu sendiri ada tiga jenis, yaitu *GET*, *POST*, dan *HEAD*. Dengan metode *GET* data dikirimkan sebagai bagian dari URL. Sedangkan dengan metode *POST* data dikirim pada *stream* terpisah. Kemudian pada metode *HEAD* data yang dikirim adalah *meta information*.

1. **Permission**

Berikut ini merupakan *permission* untuk mendefinisikan `HttpConnection`, yaitu:

```
javax.microedition.io.HttpConnection
```

2. **Status Koneksi**

Koneksi HTTP ini berada dalam satu dari dua status yang ada, yaitu :

- a. *Setup*, dimana koneksi ke server gagal diciptakan.
- b. *Connected*, dimana koneksi telah tercipta, *request* sudah terkirim, dan *response* sedang ditunggu.

Berikut ini merupakan *method* yang mungkin terlibat dalam status *Setup* :

- a. `setRequestMethod`
- b. `setRequestProperty`

Terjadinya transisi dari status *Setup* ke *Connected* disebabkan oleh beberapa *method* yang membutuhkan data untuk dikirim atau diterima dari server. *Method-method* yang dapat menyebabkan terjadinya transisi ke status *Connected* tersebut adalah :

- | | |
|----------------------------------|------------------------------------|
| a. <code>openInputStream</code> | f. <code>getHeaderField</code> |
| b. <code>openOutputStream</code> | g. <code>getResponseCode</code> |
| c. <code>getLength</code> | h. <code>getResponseMessage</code> |
| d. <code>getType</code> | i. <code>getHeaderFieldInt</code> |
| e. <code>getEncoding</code> | j. <code>getHeaderFieldDate</code> |

Selain semua *method* di atas, masih banyak *method* `HttpConnection` lainnya yang dapat digunakan kapanpun dalam membangun aplikasi.

3. Penggunaan `HttpConnection`

Hal-hal utama yang perlu diperhatikan dalam mengimplementasikan `HttpConnection` adalah :

- a. Memasukkan *permission* `HttpConnection` :

```
import javax.microedition.io.HttpConnection;
```

b. Membuat objek koneksi Http :

```
HttpConnection c = null;
```

c. Membuka URL dengan menggunakan method :

```
c = (HttpConnection)Connector.open(URL);
```

d. Menutup koneksi Http :

```
c.close();
```

2.6 General Packet Radio Service (GPRS)

General Packet Radio Service (teknologi 2,5G dengan dukungan paket data). Teknologi GPRS mengizinkan *mobile phones* untuk digunakan mengirimkan dan menerima data melalui jaringan telepon dengan kecepatan transfer data sampai 144 kbps. GPRS adalah pembawa data yang memungkinkan untuk mengakses data yang ada pada jaringan seperti internet tanpa kabel atau *wireless access*. Aplikasi yang menggunakan GPRS antara lain *WAP services*, *MMS* dan *SMS*.

2.7 J2ME Polish

Menurut sumber J2MEPolish.org, *J2ME Polish* adalah seperangkat *tool* dan teknologi yang ditujukan kepada *mobile developer* ataupun instansi yang bergerak dalam teknologi *mobile*. Fitur-fitur utama dari *J2ME Polish* adalah :

- a. *Lush* : UI *toolkit* yang sangat fleksibel dan yang dapat didesain di luar *source code* aplikasi.
- b. *Janus* : perangkat untuk menghubungkan aplikasi *mobile* ke *handset* dan *platform* yang berbeda-beda.
- c. *Touch* : teknologi untuk mengakses *server* dan berkomunikasi secara *remote*.

- d. *Trunk* : solusi untuk me-load dan menyimpan data yang kompleks hanya dengan satu baris *code*.
- e. *Marjory* : komunitas untuk mengatur perangkat *database*.

1. Lush

Dari sekian fitur yang disediakan J2ME *Polish*, fitur yang akan digunakan pada Tugas Akhir ini adalah *Lush*. Fitur ini memungkinkan *developer* untuk mengkustomisasi aplikasi tanpa mengubah *source code* dari aplikasi itu sendiri. Desain dengan berbagai animasi dan efek untuk aplikasi yang dibangun ini dispesifikasikan ke dalam file *CSS*. Proses kustomisasi ini tidak hanya memisahkan desain dengan pengembangan *business logic* aplikasi, tetapi juga memungkinkan untuk mengkustomisasinya dengan sangat mudah. Melalui fitur *Lush* J2ME *Polish* ini, *developer* dapat mendesain *background* untuk *Form* atau *List* tanpa harus menuliskan *code* yang rumit. Serta dapat menggunakan format *font* yang tidak biasa/standar, menggunakan berbagai *font effect*, berbagai animasi seperti *screen transition*, dan tampilan menu dengan menggunakan berbagai macam *view-type*.

Beberapa keuntungan penggunaan fitur *Lush* J2ME *Polish* ini diantaranya adalah :

- a. Proses desain yang serupa dengan mendesain halaman *web*.
- b. Desainer dan *developer* dapat mengubah desain kapanpun secara bebas.

- c. Kustomisasi desain untuk *customer* atau bahkan *user* yang berlainan.
- d. Adaptasi desain ke resolusi layar yang berbeda-beda.
- e. Pengembangannya yang sangat mudah, semenjak *J2ME Polish compatible* dengan standar MIDP.

Konsep dasar penggunaan GUI pada *J2ME Polish* adalah :

- a. Mengembangkan GUI dengan menggunakan *class* seperti biasa, yaitu `javax.microedition.lcdui` seperti *Form*, *List* atau *TextField*.
- b. Menggunakan penambahan *custom component* dari *J2ME Polish* seperti *TabbedForm*, *TreeItem*, *ChoiceTextField*, dan sebagainya.
- c. Jika membutuhkan *control* lebih lanjut dari komponen `javax.microedition.lcdui` seperti menyeting mode input dari *TextField*, gunakan `de.enough.polish.UIAccess`.
- d. Menambahkan direktif `#style` pada *source code* aplikasi untuk mengkoneksikan *code* tersebut dengan desain yang dibuat.
- e. Merancang dan mengkustomisasi aplikasi dengan menggunakan *textfile* CSS yang sederhana.
- f. Mengkonfigurasi penambahan GUI pada *script build.xml*.

2.8 SERVER SIDE PROGRAMMING

Server Side Programming (Scripting) merupakan sebuah teknologi *scripting* atau pemrograman web dimana

script (program) dikompilasi atau diterjemahkan di *server* dan dikirimkan ke *browser* dalam bentuk HTML. Pemrograman ini biasanya digunakan untuk menghasilkan halaman web yang dinamis dan interaktif, yang dapat berhubungan dengan *database* atau jenis penyimpanan data lainnya. Keuntungan utama dari penggunaan *server side programming* ini adalah kemampuannya dalam mengkustomisasi respon dari *server* yang didasari oleh kebutuhan dan hak akses user, atau pun *query-query* dalam *database* (Wikipedia, 2008).

Beberapa contoh *Server Side Programming* :

1. ASP (*Active Server Page*) dan ASP.NET
2. ColdFusion
3. JSP (*Java Server Pages*)
4. Perl
5. Python
6. PHP

1. Prinsip Kerja Web

Aplikasi web berjalan pada protokol HTTP, dan semua protokol di internet selalu melibatkan sisi server dan client. Ketika seseorang mengetikkan suatu alamat di *browser*, maka *browser* akan mengirimkan perintah tersebut ke *web server*. Jika yang diminta oleh *client* adalah file yang mengandung perintah *server side* maka *web server* akan menjalankan dahulu program tersebut lalu mengirimkannya kembali ke *browser* dalam bentuk HTML sehingga dapat diterjemahkan oleh *browser*. Sedangkan

jika yang diminta oleh *client* adalah file yang mengandung *client side* maka oleh *server* file tersebut akan langsung dikirimkan oleh *browser* (Caturstudio weblog, 2008).

2. Code Igniter

CodeIgniter adalah aplikasi open source yang berupa framework dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal (www.codeigniter.com).

CodeIgniter dirilis pertama kali pada 28 Februari 2006. Versi yang digunakan untuk membangun web administrator adalah versi 1.7.1.

2.9 Mobile Software Development

Dalam lingkungan modern saat ini, teknologi mobile menjadi hal yang sangat penting untuk setiap orang. Menurut sumber mengenai *Mobile Application for Business*, Teknik Informatika UAJY, teknologi ini dapat memberikan kemudahan komunikasi maupun akses internet bagi pengguna di seluruh dunia. Selain itu konsumen dan pebisnis dapat melakukan transaksi dan pertukaran informasi melalui perangkat mobile atau wireless handset seperti ponsel, PDA, atau SmartPhone. Kekurangan dari teknologi ini adalah ukuran layar dan resolusinya yang terbatas, minimalnya pemasukan informasi, dan terdapat situs web yang tidak kompatibel.

Definisi dari wireless handset itu sendiri adalah perangkat yang dapat menerima dan mengirim suara serta

data menggunakan sistem nirkabel. Komponen utama yang mencirikan wireless handset adalah :

- a. Antena : penguat / pengantara transmisi.
- b. Number pad : dialing (memasukkan angka).
- c. Speaker dan microphone : komunikasi suara.
- d. Operating system : operasi / antarmuka dengan pengguna.
- e. Computer chips : fungsi lainnya.

Pada tugas akhir ini, wireless handset yang digunakan dalam pengembangan aplikasi mobile adalah telepon selular atau ponsel. Ciri-ciri dari ponsel adalah :

- a. Transmisi sinyal digital dan analog.
- b. Adanya *phone books*.
- c. *Caller ID* dan *voicemail*.
- d. *Organic-electro-luminescent color display screen*.
- e. *Voice-activated technology* : menjalankan operasi melalui perintah suara.
- f. *Internet access*.

Di dalam wireless handset, *developer* dapat membangun berbagai macam aplikasi mobile dengan menggunakan teknologi mobile yang semakin berkembang. Dalam pengembangan aplikasi mobile tersebut, terdapat beberapa tool yang dapat digunakan yaitu Java Platform, .Net, Symbian, dsb. Pada tugas akhir ini tool yang digunakan adalah Java Platform (J2ME), yang ditujukan pada perangkat dengan RAM (*Random Access Memory*) sekitar 128 KB dan prosesor dengan kemampuan lebih rendah dari

komputer desktop. Berdasarkan arsitektur J2ME (secara khusus MIDP), J2ME ini digunakan pada perangkat mobile dengan karakteristik sebagai berikut :

- a. Minimum layar 96 x 54 pixels.
- b. 1-bit display *depth*.
- c. Memiliki input *device*, seperti keyboard/keypad, atau *touch screen*.
- d. Minimum 128 KB nonvolatile memory untuk komponen MIDP.

Memiliki dua arah konektivitas wireless.

