# BAB 2
# TINJAUAN PUSTAKA

Terdapat beberapa landasan yang menjadi dasar pembuatan tugas akhir ini. Berikut ini diuraikan tinjauan kepustakaan pendukung penelitian.

## 2.1 Sistem Operasi

Secara umum Sistem operasi atau dalam bahasa inggris: *operation system* atau OS adalah perangkat lunak sistem yang bertugas untuk melakukan kontrol dan manajemen perangkat lunak sistem yang bertugas untuk melakukan kontrol manajemen perangkat keras serta operasi-operasi dasar sistem, termasuk menjalankan software aplikasi seperti program pengolahan kata dan *browser web* [2].

### 2.1.1 Sistem Operasi Mobile BlackBerry

*Mobile Operation System* atau Sistem Operasi Genggam (disingkat *Mobile OS*) adalah suatu sistem operasi yang mengontrol sistem dan kinerja pada barang elektronik yang *mobile,* mirip dengan fungsi Windows, Mac OS X, dan Linux pada Dekstop PC atau Laptop atau Notebook tetapi lebih sederhana. Biasanya penggunanya hadir di *smartphone*, PDA, dan *tablet computer* [3].

Sedangkan Blackberry OS adalah sistem operasi *mobile* yang dikembangkan oleh BlackBerry (*company*) untuk produk–produk BlackBerrynya. OS ini mendukung *multitasking* dan dukungan khusus bagi perangkat–perangkat inputnya seperti *trackwheel*, *trackball*, *trackpad* dan *touchscreen*. *Platfrom* BlackBerry terkenal dengan dukungannya dengan email korporat melalui MIDP

1.0 dan MIDP 2.0. Dimana *platfrom* ini juga mendukung sinkronisasi dengan *Microsoft Exchange, Lotus Domino,* dan *Novell Group Wise*. BlackBerry mendukung WAP 1.2 untuk koneksi ke jaringan internet. Selain itu, *update* OS-nya juga dapat dilakukan melalui layanan BlackBerry OTASL (*Over The Air Software Landing*) sesuai dengan carrienya. BlackBerry OS juga merupakan OS terbuka dimana pengembang pihak ketiga dapat mengembangkan aplikasi–aplikasi untuk Blackberry menggunakan Blackberry API (*Application Programing Interface*) [4].

### 2.1.2 Struktur Sistem Operasi BlackBerry

Dalam sistem operasi BlackBerry terdapat beberapa file yang mendukung struktur sistem operasi yang berjalan nantinya di dalam *device :*

#### 2.1.2.1 Kernel

Kernel adalah bagian dari sistem operasi yang menjembatani antara *hardware* dan *software.* Kernel adalah jantung dari sistem operasi. Tugas utama dari kernel adalah menyediakan antar muka kepada *user* dan membuat lapisan pemisah antara *hardware, software* aplikasi dan *user.* Kernel juga menyediakan berbagai dukungan *hardware,* file *system* dan jaringan serta beberapa fasilitas dasar lainnya.

Karena akses terhadap perangkat keras terbatas, sedangkan ada lebih dari satu program yang harus dilayani dalam waktu bersamaan, maka kernel juga bertugas untuk mengatur kapan dan berapa lama suatu program dapat

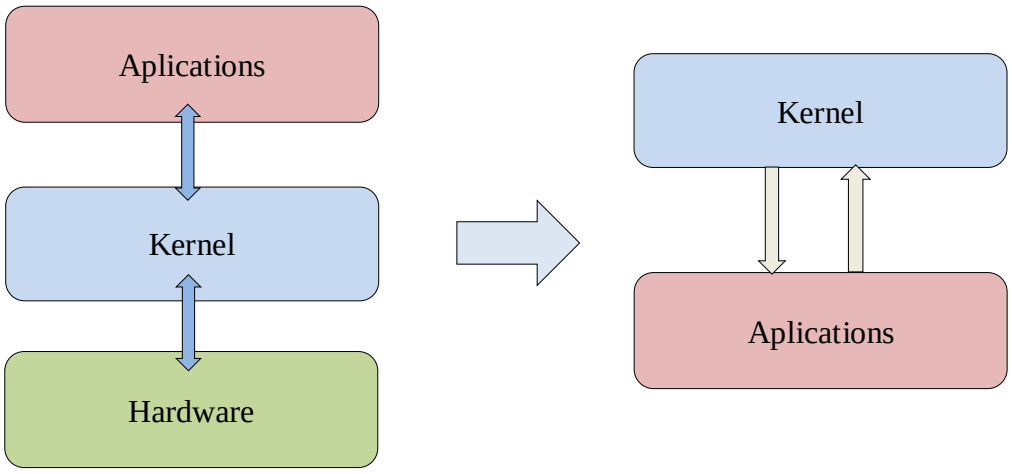
menggunakan satu bagian perangkat tersebut. Hal tersebut dinamakan *multiplexing.* Akses kepada perangkat keras secara langsung merupakan masalah yang kompleks, oleh karena itu kernel bisasanya mengimplementasikan sekumpulan abstraksi *hardware.* Abstraksi-abstraksi tersebut merupakan sebuah cara untuk menyembunyikan kompleksitas, dan memungkinkan akses kepada perangkat keras menjadi lebih mudah dan seragam. Sehingga abstraksi pada akhirnya memudahkan pekerjaan *programmer.*

#### 2.1.2.2 Kernel Monolitik

Kernel yang menyediakan abstraksi perangkat keras yang kaya dan tangguh. Kernel monolitik mengintegrasikan banyak fungsi di dalam kernel dan menyediakan lapisan abstraksi perangkat keras secara penuh terhadap perangkat keras yang berada dibawah sistem operasi. Pendekatan kernel monolitik didefinisikan sebagai sebuah antar muka virtual yang berada pada tingkat tinggi perangkat keras, dengan sekumpulan primitif atau *system call* untuk mengimplementasikan layananlayanan sistem operasi, seperti halnya manajemen proses, konkurensi, dan manajemen memori pada modul-modul kernel yang berjalan di dalam *mode supervisor*. Meskipun jika setiap modul memiliki layanan operasioperasi tersebut terpisah dari modul utama, integrasi kode yang terjadi di dalam monolitik kernel sangatlah kuat, dan karena semua modul berjalam dalam *address space* yang sama, sebuah *bug* dalam salah satu modul dapat merusak keseluruhan sistem. Akan tetapi, ketika implementasi dilakukan secara benar, integrasi komponen internal yang sangat kuat tersebut justru menginjinkan fitur-fitur yang dimiliki oleh sistem yang berada dibawahnya dieksploitasi secara

efektif, sehingga membuat sistem operasi dengan kernel monolitik sangatlah efisien, meskipun sangat sulit membuatnya.

Pada sistem operasi modern yang menggunkan kernel monolitik, seperti halnya GNU/Linux, FreeBSD, Open Solaris, dan Microsoft Windows, dapat memuat modul-modul yang dapat dieksekusi pada saat kernel tersebut dijalankan sehingga mengijinkan eksistensi terhadap kemampuan kernel sesuai kebutuhan, dan tentu saja dapat membantu menjaga agar kode yang berjalan pada ruangan kernel seminim mungkin.




**Gambar 2.1** Diagram Monolitik Kernel

#### 2.1.2.3 Modul Kernel BlackBerry

Modul kernel BlackBerry adalah bagian dari kernel BlackBerry yang dapat dikompilasi, dipanggil, dan dihapus secara terpisah dari bagian kernel lainnya saat dibutuhkan. Modul kernel dapat menambahkan fungsionalitas kernel tanpa perlu me-*reboot* sistem. Secara teori tidak ada yang dapat membatasi apa yang dapat dilakukan oleh modul kernel. Kernel modul dapat mengimplementasikan antara *device driver,* sistem berkas, dan protokol jaringan.

#### 2.1.2.4 Manajemen Modul Kernel BlackBerry

Manajemen modul akan mengatur pemanggil modul ke dalam memori dan berkomunikasi dengan bagian lainnya dari kernel. Memanggil sebuah modul tidak hanya memasukkan isi binarinya ke dalam memori kernel, namun juga harus dipastikan bahwa setiap rujukan yang dibuat oleh modul ke simbol kernel atau titik masukan diperbaharui untuk menunjukan ke lokasi yang benar di alamat kernel. BlackBerry membuat file ALX di kernel, dimana file ALX ini diperlukan untuk penanganan rujukan yaang dilakukan oleh modul terhadap file-file COD yang berada pada folder JAVA.

#### 2.1.2.5 Registrasi Driver

Untuk membuat modul kernel baru dipanggil berfungsi, bagian dari kernel yang lain harus mengetahui keberadaan dan fungsi baru tersebut. Kernel membuat table dinamis yang berisi semua *driver* yang telah diketahuinya dan menyediakan serangkaian routines untuk menambahkan dan menghapus *driver* dari table tersebut. *Routines* yang bertanggung jawab untuk mendaftarkan fungsi modul baru tersebut.

Hal-hal yang masuk kedalam tabel registrasi BlackBerry adalah:

1. Version
2. Code
3. Data
4. Literal Data
5. Classes
6. Size

#### 2.1.2.6 File COD

COD (*Completed Java Code*) merupakan file yang melalui proses kompilasi dan memiliki dependensi kode yang tinggi. File COD merupakan inti program atau aplikasi yang akan diload ke dalam BlackBerry dan dibangun menggunakan sekumpulan file Java dalam satu *workspace.*

File COD bisa dieksekusi bersamaan dengan file ALX sehingga pada saat proses *load* menjadi bersamaan. File-file COD terdiri dari sistem inti (*Core System Application*) yang merupakan bawaan dari BlackBerry (*company*) dan aplikasi pihak ketiga (*3rb party*) yang dibuat oleh *developer* untuk BlackBerry. Aplikasi yang sudah terverifikasi (*signed application*) oleh BlackBerry (*company*) akan bisa diunduh ke dalam *device* melalui *BlackBerry Application World* bisa didapatkan *free* maupun berbayar.

Contoh file COD :

```
net_rim_bb_qm_api.cod
net_rim_bb_qm_bbm_internal_api.cod
net_rim_bb_qm_lib_barcode.cod
net_rim_bb_qm_peer.cod
net_rim_bb_qm_peer_images.cod
```

**2.1.2.7 File ALX**

ALX (*Application Loader XML*) yaitu sebuah script perantara untuk memancing file COD java *module* untuk *load* ke BlackBerry. File ALX merupakan turunan file .xml pada pemrograman berbasis web. Intinya, semua file OS yang akan di-*load* ke BlackBerry berdasarkan perintah dari file .alx tersebut melalui *loader Java Virtual Machine (JVM).*

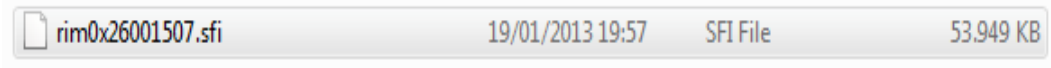
Contoh file ALX :

```
      <loader version="1.0">
<application id="net.rim.java.blackberrymessenger">
      <name>BlackBerry Messenger</name>
      <description>Kukurubow.unikom</description>
      <version>7.0.1.23</version>
      <vendor>Research In Motion Limited</vendor>
      <copyright>Copyright  (c) 1998-2011 Research In Motion
Limited</copyright>
      <language langid="0x0003">
            <name>BlackBerry Messenger</name>
```

```
            <description>BlackBerry Messenger</description>
      </language>
      <language langid="0x000c">
            <name>BlackBerry Messenger</name>
            <description>BlackBerry Messenger</description>
      </language>
      <language langid="0x0013">
            <name>BlackBerry Messenger</name>
            <description>BlackBerry Messenger</description>
</language>
```

#### 2.1.2.8 File SFI

File SFI atau juga bisa disebut file Radio (firmware) berfungsi sebagai integrasi dengan file.cod dan mempunyai *depedencies* yang kuat. Intinya file-file SFI sangatlah utama yang menentukan kualitas *Lock Signal* sehingga indikator sinyal pada BlackBerry tidak cepat berubah. File SFI merupakan file *close source* yang dibuat oleh BlackBerry (company), file tidak dapat dikembangkan karena file tersebut sudah menyatu menjadi satu dengan *handheld firmware* itu sendiri. File SFI juga sebagai mekanisme *boot*, yaitu meletakkan kernel ke dalam memory kernel, kernel dapat dikatakan sebagai inti dari sistem operasi BlackBerry.




**Gambar 2.2** File SFI

#### 2.1.2.9 JVM (*Java Virtual Machine*)

JVM *(Java Virtual Machine)* berfungsi sebagai *loader*, dimana jvm berpusat sebagai alur pertama dalam hal *booting.* JVM akan membaca semua isi folder java, di dalam folder java tersebut terdapat kumpulan file-file COD, dimana file-file COD itu sendiri berfungsi sebagai isi dari *OS* yang akan di instalkan kedalam *device BlackBerry*, file COD yang terdapat di dalam folder java tersebut yang akan mempengaruhi performa dari *booting* di saat penginstalan selesai nantinya.

## 2.2 Landasan Teori

### 2.2.1 Java Pada Perangkat Mobile

Ada banyak *platfrom* Java pada suatu sistem. Berikut ini penjelasan-penjelasan *platfrom* Java yang berjalan pada sistem yang banyak dijumpai [5].

#### 2.2.1.1 Platfrom Java

Java dibuat pada tahun 1991 oleh James Gosling. Pada awalnya diberi nama *Oak*, untuk menghormati pohon yang ada di luar jendela Gosling. Kemudian namanya dibuah ke Java karena telah ada sebuah bahasa yang diberi nama *Oak.*

Motivasi sesungguhnya dari Java adalah kebutuhan akan sebuah bahasa yang bisa digunakan pada berbagai *platfrom* yang bisa dimasukkan ke dalam berbagai produk elektronik seperti pemanggang roti dan lemari es. Salah satu dari proyek pertama yang dikembangkan menggunakan JAVA adalah sebuah *remote* kontrol yang diberi nama Star 7. Pada saat yang sama, *World Wide Wibe* dan

*internet* berkembang sangat cepat. Gosling menyadari bahwa Java dapat digunakan untuk pemrograman Internet. Dengan keluarnya versi 1.2, *paltfrom* Java telah dipilah-pilah menjadi beberapa edisi: The Standar Edition (J2SE), Entrprise Edition (J2EE), Mobile Edition (J2ME), dan JavaCard API.

**Tabel 2.1** Platfrom Java

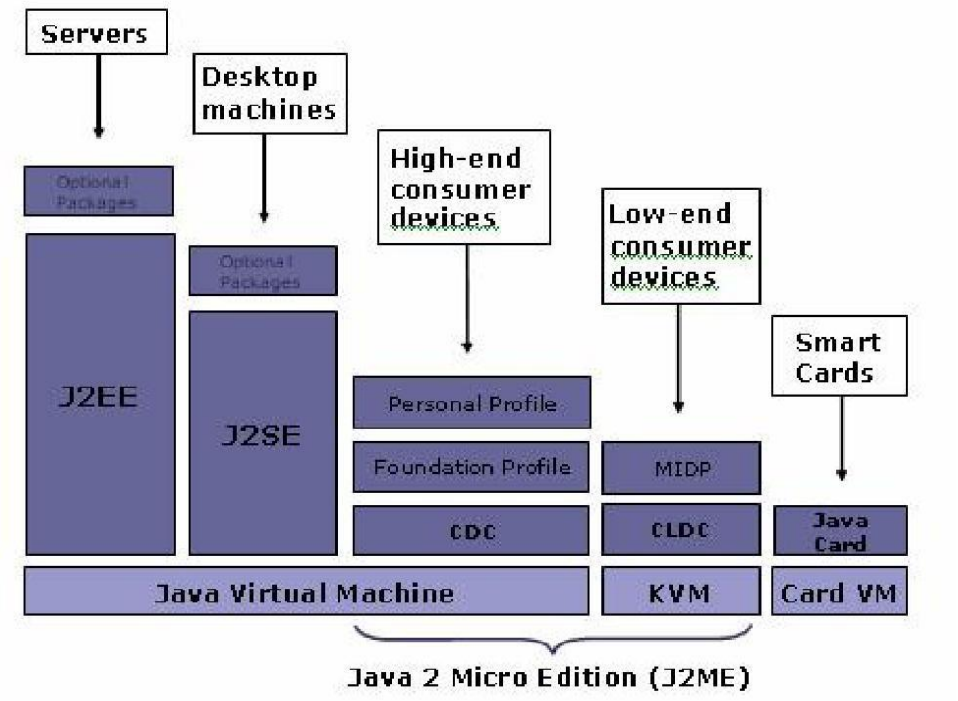
| | |
|---|---|
| J2SE – Java 2 Platform, Standart Edition | Aplikasi Desktop |
| J2EE – Java 2 Platform, Enterprise Edition | Aplikasi Enterprise dengan 18ocus pada pengembangan sisi web server, termasuk servlet, JSP, EJB, dan XML |
| J2ME – Java 2 Micro Edition | Perangkat Mobile |
| Java Card | Smart Cards |

### 2.2.1.2 Java Micro Edition

*Java Micro Edition* (J2ME) adalah salah satu set spesifikasi dan teknologi yang fokus kepada perangkat konsumen. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterei, layar yang kecil, dan *bandwith* jaringan yang rendah. *Java* menyediakan suatu lingkungan yang *portable* untuk mengembangkan dan menjalankan aplikasi pada perangkat ini. Program J2ME, seperti semua program *java* adalah diterjemahkan oleh *Virtual Machine* (VM). Program-program tersebut dikompile ke dalam *bytecode* dan diterjemahkan denga *Java Virtual Machine* (JVM). Ini berarti bahwa program-program tersebut tidak berhubungan langsung dengan perangkat. J2ME

menyediakan suatu *interface* yang sesuai dengan perangkat. Aplikasi-aplikasi tersebut tidak harus dikompile ulang supaya mampu dijalankan pada mesin yang berbeda. Inti dari J2ME terletak pada *configuration* dan *profile*. Suatu *configuration* menggambarkan lingkungan *runtime* dasar dari suatu sistem J2ME. Ia menggambarkan *core library*, *virtual machine*, fitur keamanan dan jaringan.

Java dikenal baik untuk kemampuannya dalam menjalankan *downloaded code* seperti *applets*. Aplikasi MIDP juga tidak akan pernah bisa menulis dalam *device memory* yang tidak ada JVM. Aplikasi MIDP tidak akan pernah mengacaukan aplikasi lain pada alat yang sama atau OS itu sendiri, inilah fitur yang hebat dalam MIDP yang membuat pengembang untuk melakukan pengembangan aplikasi di seluruh dunia karena tidak ada ketakutan programmer akan menulis aplikasi yang merusak *hand phone.*




**Gambar 2.3** Java Micro Edition (J2ME)

J2ME memiliki dua kategori *Configuration* yaitu : CDC (*Connected Device Configuration*) dan CLDC (*Connected Limited Device Configuration*). *Configuration* adalah spesifikasi minimal sebuah perangkat yang menjadi target Java ME. Configuration terdiri dari sebuah *virtual machine*, sekumpulan *API/library* minimal yang harus disediakan perangkat tersebut.

#### 2.2.1.3 Connnected Limited Device Configuration (CLDC)

CLDC atau *Connected Limited Device Configuration* adalah perangkat dasar dari J2ME, spesifikasi dasar yang berupa *library* dan API yang diimplementsikan pada J2ME seperti yang digunakan pada telepon selular, pager, dan PDA. Perangkat tersebut dibatasi oleh keterbatasan *memory* sumber daya, memiliki prosesor yang lambat sekitar 16 bit atau 32 bit, memori minimal sebesar 192Kb, dan memiliki koneksi terbatas, artinya user bisa online jika diperlukan. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal dari *package*, kelas dan sebagai fungsi *Java Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat-alat tersebut, JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*). Paket CLDC terdiri dari java.lang, java.io, dan java.util.

#### 2.2.1.4 Connnected Device Configuration (CDC)

CDC digunakan untuk perangkat yang memiliki prosesor dan memori yang lebih besar dibandingkan CLDC, misalnya *higher end* PDA, TV *set top boxes*. Implementasi CDC dapat penggunakan JVM standar (Java SE), dengan

perangkat yang memiliki memori sebesar 2MB atau lebih dan memiliki koneksi jaringan yang selalu online (*connected*). Paket CDC terdiri dari:

1. java.io
2. java.lang
3. java.lang.ref
4. java.lang.math
5. java.net
6. java.security
7. java.security.cert
8. java.text
9. java.util
10. java.util.jar
11. java.util.zip

#### 2.2.1.5 MIDP

*Mobile Information Device Profile* (MIDP) berada di atas dari CLDC. menulis aplikasi *mobile* tidak hanya dengan menggunakan CLDC API. tetapi harus tetap memanfaatkan MIDP yang mendefinisikan UI (*User Interface*). Spesifikasi MIDP, kebanyakan seperti CLDC dan API lainnya sudah digambarkan melalui *Java Community Process* (JCP). JCP melibatkan sebuah kelompok ahli berasal dari lebih dari 50 perusahaan, yang terdiri atas pabrik perangkat mobile, pengembang software. MIDP terus berkembang, dengan versi-versi masa depan yang telah lulus dari proses ketat JCP. Spesifikasi MIDP menggambarkan suatu perangkat MIDP yang memiliki karakteristik-karateristik ini sebagai batas minimum :

**Tampilan:**

1. Ukuran Layar: 96x54.
2. kedalaman tampilan: 1-bit.
3. Ketajaman pixel: sekitar 1:1.

**Masukan:**

1. Satu atau lebih mekanisme *user-input*: satu *keyboard*, dua *keyboard*, atau *touch screen.*

**Memory:**

1. 256 *kilobytes of non-volatile memory* untuk implementasi MIDP.
2. 8 *kilobytes of non-volatile memory for application-created persistent Data.*
3. 128 kilobytes *of volatile memory for the Java runtime*.

**Jaringan:**

1. Dua jalur, *wireless*, *bandwidth* terbatas.

**Sound:**

1. Kemampuan untuk memainkan nada-nada MIDP menggambarkan model aplikasi, UI API, penyimpanan dan jaringan yang kuat, permainan dan media API, kebijakan keamanan, penyebaran aplikasi dan ketetapan *Over The Air* (OTA).

#### 2.2.1.6 RIMLET

RIMlet merupakan *library application packages*, pada aplikasi BlackBerry dapat dikembangkan sebagai MIDlet standar yang akan berjalan pada perangkat MIDP-enabled, atau sebagai RIMlet, aplikasi berbasis CLDC yang menggunakan BlackBerry spesifik API dan karenanya akan hanya berjalan pada

perangkat BlackBerry. Jika ingin mengembangkan aplikasi khusus untuk BlackBerry harus menggunakan model API CLDC karena RIM akan memberikan tampilan ”rasa” BlackBerry tanpa melarang pilihan untuk menggunakan API J2ME standar di daerah lain dari UI. Untuk ketekunan, dapat menggunakan API BlackBerry atau RMS MIDP API. Kelas utama dari RIMlet *ezntend*s net.rim.system Application, jika itu merupakan aplikasi *background* tanpa *user interface*, atau *extends* net.rim.system. UiApplication jika RIMlet membutuhkan *user interface*. Tidak seperti titik awal MIDlet, titik masuk ke sebuah RIMlet adalah *main()method*. Seperti dalam UI MIDP, layar RIMlet bukan *window* yang bergerak. Untuk menampilkannya hanya diperlukan *method pushScreen*().

#### 2.2.1.7 UiApplication

Semua aplikasi yang memiliki *user interface (screen, menus, dll)* harus menjadi subsclass net.rim.device.api.ui.UiApplication. Kita dapat melakukan dengan menambahkan import pada bagian paling atas program seperti *listing program* dibawah ini :

```
import net.rim.device.api.ui.UiApplication;
/** contoh sederhana Hello World pada BlackBerry.
*/

class HelloWorldApp extends UiApplication {
HelloWorldApp(){ }
}
```

#### 2.2.1.8 BlackBerry Sistem API

Sistem API meliputi berbagai fungsi sistem inti (*Core System*) Mereka menyediakan fungsi untuk manajemen *thread*, penanganan pesan, dan tugas *switching*. Mereka juga menyediakan fungsi untuk alokasi memori, *timer*, alarm dan sistem jam. *Threads* dapat memberikan informasi tentang status baterai, bahasa, status genggam dalam satu sistem operasi BlackBerry secara keseluruhan API Sistem pada BlackBerry *Java Library* seperti pada tabel dibawah [6].

**Tabel 2.2** RIM Java Library

| | |
|---|---|
| net.rim.device.api.bluetooth | Menyediakan dukungan komunikasi serial untuk koneksi Bluetooth |
| net.rim.device.api.browser | Menyediakan akses ke browser-komponen fungsi dan dapat memasukkan ke dalam aplikasi lain |
| net.rim.device.api.collection | Mendefinisikan fungsi dasar untuk koleksi data |
| net.rim.device.api.collection.util | Menyediakan berbagai implementasi pengumpulan dan utilitas |
| net.rim.device.api.compress | Menyediakan komponen untuk mengompresi dan decompressing data |
| net.rim.device.api.i18n | Menyediakan fungsi yang mendukung internasionalisasi aplikasi pada Blackberry |
| net.rim.device.api.io | Menyediakan komponen untuk mengelola data I / O |
| net.rim.device.api.itpolicy | Berisi kelas yang didefinisikan fungsi kebijakan IT |
| net.rim.device.api.lowmemory | Mendefiniskan memori Blackberry yang menipis |
| net.rim.device.api.memorycleaner | Mendefinisikan memori untuk dibersihkan |
| net.rim.device.api.notification | Menyediakan fungsi untuk memicu pemberitahuan acara melalui Konsekuensi (yaitu, nada, getaran atau LED) atau Notification Engine Listner yaitu dialog atau interaksi layar pada perangkat Blackberry |

| | |
|---|---|
| net.rim.device.api.servicebook | Mendefinisikan aplikasi service book |
| net.rim.device.api.smartcard | Menyediakan perpustakaan komponen untuk berinteraksi dengan kartu SD card |
| net.rim.device.api.synchronization | Sinkronisasi data pada Blackberry |
| net.rim.device.api.ui | Menyediakan fungsionalitas dasar untuk membangun user interface dari aplikasi perangkat Blackberry |
| net.rim.device.api.ui.autotext | Menyediakan layanan library auto text untuk *user* |
| net.rim.device.api.ui.text | Menyediakan kelas untuk menyaring string teks yang berisi berbagai macam data |
| net.rim.device.api.util | Menyediakan sejumlah metode utilitas dan interface yang berguna dalam konteks sistem Blackberry |
| net.rim.plazmic.mediaengine | Layanan Tema |
| net.rim.plazmic.mediaengine.io | Layanan tema I/O |

#### 2.2.1.9 Komunikasi Radio API

BB API Komunikasi Radio menyediakan akses mudah ke jaringan radio untuk mengirim dan menerima data.

Mengenai Komunikasi Radio, berikut event yang mungkin terjadi :

MESSAGE_RECEIVED : Pesan Diterima

MESSAGE_SENT : Pesan Terkirim

MESSAGE_NOT_SENT : Pesan Tidak Terkirim

SIGNAL_LEVEL : Tahapan Sinyal

NETWORK_STARTED : Jaringan Mulai

BASE_STATION_CHANGE : Stasiun Utama Ganti

RADIO_TURNED_OFF : Radio Mati

MESSAGE_STATUS : Status Pesan

#### 2.2.1.10 Sistem Event

Sistem event berikut ini yang mungkin terjadi :

SWITCH_BACKGROUND : Tukar Latar Belakang

SWITCH_FOREGROUND : Tukar Latar Depan

POWER_OFF/POWER_UP : Mesin Mati/Hidup

BATTERY_LOW : Baterai Hampir Habis

TASK_LIST_CHANGED : Ganti Daftar Tugas

TIMER DEVICE EVENTS : Peralatan Waktu Acara

#### 2.2.1.11 Keypad API

Tombol fungsi :

1. KeypadBeep : mengaktifkan nada tombol aktif atau tidak aktif.
2. KeypadRate : megkonfigurasi tombol untuk pengulangan tombol otomatis.
3. KeypadRegister : memungkinkan aplikasi untuk mencegah kunci panas global.

Event meliputi : THUMB_CLICK, THUMB_UNCLICK, THUMB_ROLL_UP, THUMB_ROLL_DOWN, KEY_REPEAT, KEY_STATUS.

#### 2.2.1.12 API Lainnya

Ada berbagai macam API *packages* untuk *BlackBerry OS* yang bisa *load* ke dalam *device* Blackberry. *Packages* tersebut adalah seperti pada tabel berikut :

**Tabel 2.3** Blackberry API *Packages*

| | |
|---|---|
| net.rim.blackberry.api.browser | Menyediakan akses ke aplikasi BlackBerry Browser untuk membuat dan menampilkan halaman HTML atau WML dan untuk mengatur opsi browser |
| net.rim.blackberry.api.homescreen | Menyediakan akses interface screen |
| net.rim.blackberry.api.invoke | Memungkinkan aplikasi pihak ketiga untuk memanggil aplikasi internal |
| net.rim.blackberry.api.mail | Menyediakan fungsi untuk mengirim, menerima dan mengakses pesan email dengan menggunakan BlackBerry |
| net.rim.blackberry.api.mail.event | Mendefinisikan email pesan kepada user |
| net.rim.blackberry.api.menuitem | Menyediakan fungsi untuk menambahkan item menu kustomisasi untuk aplikasi |
| net.rim.blackberry.api.options | Menyediakan fungsi untuk menambahkan item pengaturan |
| net.rim.blackberry.api.phone | Menyediakan akses ke aplikasi Phone BlackBerry untuk memulai panggilan, menerima pemberitahuan event, dan perubahan pengaturan |
| net.rim.blackberry.api.phone.phonelogs | Menyediakan akses ke file log phone |
| net.rim.blackberry.api.pim | Menyediakan akses ke kontak alamat, Kalender, dan aplikasi tugas/memo untuk melihat, menambah atau mengubah kontak pengguna |
| net.rim.blackberry.api.sms | Menyediakan layanan SMS |

### 2.2.2 Kebutuhan Perangkat Keras

Untuk menunjang semua *tools* yang ada maka perangkat keras yang dibutuhkan selama tahap pengembangan sistem ini adalah *Personal Computer (PC)* dengan spesifikasi minimum :

1. Intel Pentium 4 prosesor (2.0 GHz atau lebih).
2. 512 MB RAM atau lebih.
3. 64 MB video memori.
4. 2 GB *free harddisk space.*

#### 2.2.2.1 Kebutuhan Informasi Perangkat Keras (*device)*

Sebelum mengetahui seluk beluk *device* yang akan dikembangkan, walaupun sebenarnya pengujian bisa dilakukan melalui *simulator*, akan tetapi akan layak jika diaplikasikan pada perangkat yang sebenarnya. Dibawah ini menjelaskan spesifikasi *device* yang akan diaplikasikan :

**Tabel 2.4 Informasi Perangkat (*device)***

| Available Features | Carrier Networks |
|---|---|
| 1. BlackBerry 7 OS<br>2. 5 MP camera<br>3. Optical trackpad<br>4. GPS<br>5. Wi-Fi<br>6. Near Field Communication (NFC)<br>7. Email and text messaging<br>8. Social networking apps<br>9. Bluetooth | 1. UMTS: 2100/1900/850/800 MHz or 2100/1700/900 MHz<br>2. GSM/GPRS/EDGE: 850/900/1800/1900 MHz<br>3. HSUPA: 5.76 Mbps<br>4. HSDPA: 7.2 Mbps |

| Memory | Processor |
|---|---|
| 1. Memory: 768 MB RAM, 8 GB eMMC<br>2. Expandable memory: up to 32 GB with microSD card | 1. Marvel Tavor MG1 1 GHz |

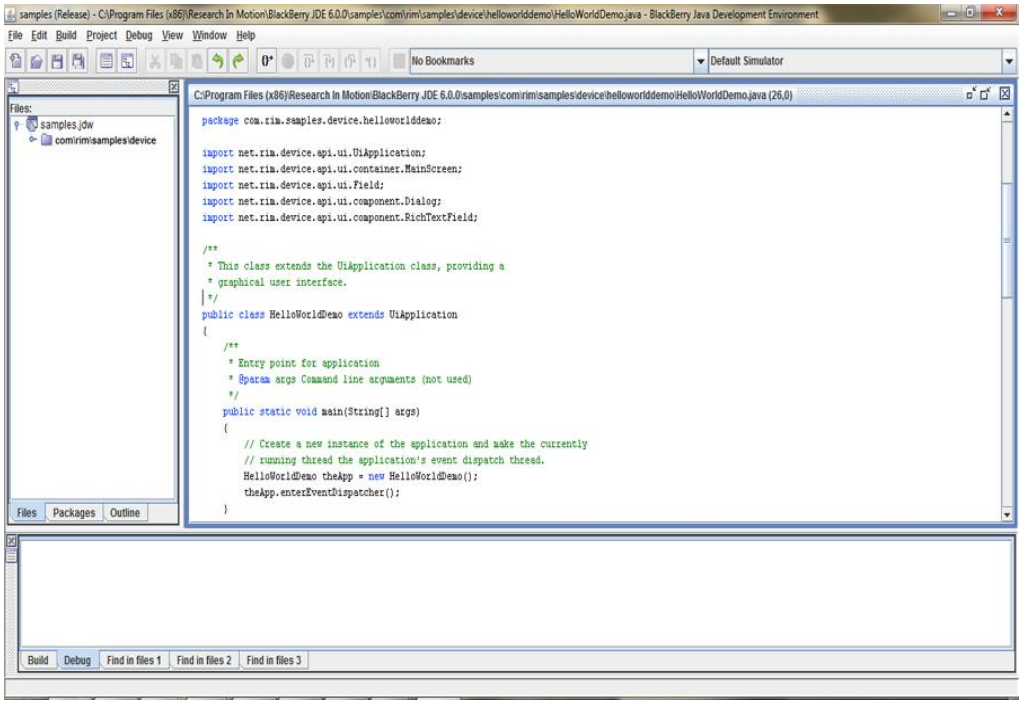
### 2.2.3 *Software* Pendukung

Proses pengembangan sistem ini menggunakan berbagai *software* pendukung yang disediakan oleh RIM untuk *developer* sehingga memudahkan dalam pengembangan seluruh sistem operasi, mulai dari *coding, inject file, installation, back-up data, wipe data, restore data,* dsb.

Berikut adalah *tools* yang digunakan penulis untuk proses pengembangan OS Blackberry sampai akhirnya mendapatkan hasil.

#### 2.2.3.1 BlackBerry *Java Development Environtment*

*BlackBerry Java Development Environtment* (JDE) adalah alat atau tools yang memungkinkan pengembang untuk membuat dan menguji aplikasi BlackBerry Java yang berjalan pada *BlackBerry OS* Java *smartphone.* BlackBerry JDE menyediakan lingkungan pengembang untuk *codeline* OS BlackBerry tertentu. BlackBerry JDE termasuk *simulator* BlackBerry, BlackBerry MDS simulator, dan alat untuk membangun dan menguji aplikasi BlackBerry. BlackBerry JDE juga menyediakan *library* dan RIMlet yang memungkinkan pengembang dilakukan dengan *workspace* yang sama sehingga memudahkan *developer* untuk membangun, melakukan *compile, running,* dan juga dokumentasi file Java yang dibuat [7].




**Gambar 2.4** Blackberry JDE *user interface*

### 2.2.3.2 BlackBerry *Simulator*

BlackBerry *Simulator* adalah sebuah aplikasi yang diinstal pada komputer dan meniru *interface* dari model dan memiliki tampilan dari sebuah *device* BlackBerry yang sebenarnya. Fungsi ini meliputi interaksi pengguna yang sama dari sebuah BlackBerry yang sebenarnya (termasuk *trackball, trackpad,* atau *trackwheel,* dan layar sentuh atau *keyboard),* aplikasi yang sama, dan fitur yang sama, seperti pesan email, telepon, dan browsing internet.

BlackBerry *Simulator* berfungsi sebagai *platfrom* dimana aplikasi dapat berjalan, termasuk kemampuan aplikasi untuk membuat koneksi jaringan, menyimpan data, dan menangani pesan email. Dan yang paling penting memberi kemampuan *developer* untuk menginstal dan menguji aplikasi yang sedang dibuat [8].




**Gambar 2.5** Blackberry *Simulator*

### 2.2.3.3 BBSAK (*BlackBerry Swiss Army Knife*)

BBSAK merupakan *software* atau *tools* yang berfungsi sebagai pendukung *developer* untuk memodifikasi *device* secara mudah, *BBSAK* dibuat oleh software house yang mengatasnamakan *JMTLab*, dengan *BBSAK* kita dapat membackup data, merestore data, mengupgrade *firmware*, *downgrade firmware*, dan memudahkan *developer* untuk menginstall *(inject)* aplikasi yang dibuat kedalam BlackBerry sebenarnya setelah pengujian melalui *BlackBerry Simulator*.




**Gambar 2.6** BBSAK

### 2.2.4 *Base OS Official* BlackBerry

*Base OS Official* yang akan digunakan dalam *mixing* modul-modul untuk mengoptimalkan kinerja sistem operasi BlackBerry menggunakan 2 *base OS* :

1. 9790M_v7.1.0.907_P9.16.0.180

   *Base OS* 7.1.0.907 adalah *OS* pengembangan dari versi 7.0.0.711 yang dilakukan oleh *Blackberry*, namun menurut *observasi* yang dilakukan dalam *base os* 7.0.0.711 ada kekurangan yang ditemukan dalam hal, ketahanan baterai, lambatnya dalam *booting*, pencarian yang masih *error* di *maps* dan *error virtual keyboard*. Dalam versi 7.1.0.907 sudah tidak ditemukan *error* seperti yang dijelaskan. Berikut ini *listing module* yang digunakan dalam pembangunan *OS Repackaged :*

**Tabel 2.5 *Listing Module 7.1.0.907***

| *Module* | 9790M_v7.1.0.907_P9.16.0.180 |
|---|---|
| 1. net_rim_batterystretch<br>2. net_rim_bb_apps_framework<br>3. net_rim_bb_diagnostic<br>4. net_rim_bb_email<br>5. net_rim_bb_gps_ee<br>6. net_rim_bb_ldap_browser<br>7. net_rim_bb_medialibrary<br>8. net_rim_bb_messaging<br>9. net_rim_bb_options_app<br>10. net_rim_bb_ribbon_app<br>11. net_rim_bb_ribbon_skin_svg<br>12. net_rim_bb_ribbon_skin_svg_DelayedInitializer<br>13. net_rim_bb_videorecorderlibrary<br>14. net_rim_bb_virtual_keyboard_framework<br>15. net_rim_bb_virtual_keyboard_layout360x480<br>16. net_rim_bbapi_blackberrymessenger<br>17. net_rim_bbapi_browser<br>18. net_rim_bbapi_mail<br>19. net_rim_bbapi_maps | 7.1.0.907 |

2. 9790M_v7.1.0.912_P9.16.0.180

*Base OS* 7.1.0.912 adalah *OS* pengembangan dari versi 7.0.0.711 yang dilakukan oleh *Blackberry,* namun menurut *observasi* yang dilakukan dalam *base os* 7.0.0.711 masih ada kekurangan yang ditemukan dalam hal, kamera yang tidak bisa digunakan untuk foto, *recorder* yang tidak bisa berfungsi dengan baik, *safe mode* yang bermasalah, *notifications* yang telat muncul disaat sms, atau bbm berjalan. Untuk penyempurnaan dalam peningkatan kinerja sistem operasi, beberapa modul dari versi 7.1.0.912 digunakan.

Berikut ini *listing module* yang digunakan dalam pembangunan *OS Repackaged :*

**Tabel 2.6 *Listing Module 7.1.0.912***

| *Module* | 9790M_v7.1.0.912_P9.16.0.180 |
|---|---|
| 1. net_rim_bb_addressbook<br>2. net_rim_bb_BluetoothFileTransfer<br>3. net_rim_bb_camera<br>4. net_rim_bb_email_compressionl<br>5. net_rim_bb_mediarecorder<br>6. net_rim_bb_safe_mode<br>7. net_rim_bb_task_app<br>8. net_rim_wlan_notifications | 7.1.0.912 |

### 2.2.5 Teknik Pengolahan Data Angket

Skala likert adalah suatu skala psikometrik yang digunakan dalam kuesioner dan merupakan skala satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Rensis Likert telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932 yang sekarang

terkenal dengan nama skala *likert*. Skala *likert* digunakan untuk mengukur kesetujuan dan ketidaksetujuan seseorang terhadap sesuatu rencana program, pelaksanaan program, ataupun tingkat keberhasilan suatu program.

Skala Likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial [10].

Dalam skala likert terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif, dan pernyataan *negative* yang berfungsi untuk mengukur sikap *negative* objek.

**Tabel 2.7 Skala Penilaian Kuesioner**

| No. | Keterangan | Skor Positif | Skor Negatif |
|---|---|---|---|
| 1. | Sangat Setuju | 5 | 1 |
| 2. | Setuju | 4 | 2 |
| 3. | Ragu-Ragu | 3 | 3 |
| 4. | Tidak Setuju | 2 | 4 |
| 5. | Sangat Tidak Setuju | 1 | 5 |

Cara perhitungan :

Hasil jumlah skor masing-masing nilai jawaban kuesioner yang diujikan kepada tiap responden dengan diberikan 5 skala menggunakan skala likert. Untuk menghitung skor maksimum tiap jawaban. Skor yang diberikan untuk jawaban kuesioner :

1. Jawaban A diberi skor 5
2. Jawaban B diberi skor 4
3. Jawaban C diberi skor 3
4. Jawaban D diberi skor 2
5. Jawaban E diberi skor 1