BAB II

LANDASAN TEORI

2.1. Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *Open Office.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah. Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain :

1. Perangkat lunak perusahaan *(enterprise)*

2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk

2.2. Mobile

Mobile adalah perangkat elektronik portabel yang berfungsi sebagaimana pesawat telepon normal, yang dapat bergerak pada suatu area yang luas. (bandingkan dengan *cordless phone*).

Aplikasi bergerak merupakan istilah yang digunakan untuk suatu aplikasi yang dijalankan pada perangkat *mobile* yang cenderung memiliki ukuran kecil, *portable* dan dapat terkoneksi melalui jaringan nirkabel. Perangkat *mobile* memiliki banyak jenis dalam hal ukuran, desain dan *layout*, tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari sistem desktop.

2.2.1. Aplikasi Mobile

Aplikasi *mobile* adalah aplikasi yang melibatkan piranti bergerak dan melibatkan media komunikasi nirkabel. Oleh karena itu, peminatan ini menekuni sistem tersebar, keamanan jaringan, pemrograman jaringan, aplikasi *mobile* untuk bisnis, analisis dan perancangan jaringan, pemrograman web lanjut, sistem layanan berbasis lokasi, infrastruktur layanan *mobile*,

teknologi nirkabel dan aplikasinya, sistem tertanam, serta hal-hal yang berkaitan dengan itu, termasuk di dalamnya aspek perangkat keras komputer.

2.3. Barang

Barang atau komoditas dalam pengertian ekonomi adalah suatu objek atau jasa yang memiliki nilai. Nilai suatu barang akan ditentukan karena barang itu mempunyai kemampuan untuk dapat memenuhi kebutuhan.

Dalam makroekonomi dan akuntansi, suatu barang sering dilawankan dengan suatu jasa. Barang didefinisikan sebagai suatu produk fisik (berwujud, *tangible*) yang dapat diberikan pada seorang pembeli dan melibatkan perpindahan kepemilikan dari penjual ke pelanggan, kebalikan dengan suatu jasa (tak berwujud, *intangible*). Istilah "komoditas" sering digunakan dalam mikroekonomi untuk membedakan barang dan jasa.

2.4. Inventaris

Inventaris adalah benda milik perusahaan, atau organisasi, atau yayasan, atau lembaga apa saja sesuai dengan konteksnya yang bsa digunakan oleh anggotanya. Inventarisasi bertujuan untuk memberi tanda pengenal bagi semua fasilitas di industri. Inventaris yang dibuat harus mengandung informasi yang jelas dan mudah dimengerti dengan cepat, sehingga dapat membantu kelancaran pekerjaan. Dengan demikian pekerjaan perawatan akan lebih mudah.

2.5. Android

Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, *T-Mobile*, dan *Nvidia*.

2.5.1. Sejarah Android

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, *California Amerika Serikat*. Para pendiri Android Inc. bekerja pada Google, di antaranya *Andy Rubin, Rich Miner, Nick Sears,* dan *Chris White*. Saat itu banyak yang menganggap fungsi Android Inc. Hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. versi android terbaru yaitu versi 3.0. Android juga sudah bergabung dengan beberapa smart Mobile seperti Nokia, Sony Ericsson, dan lainnya. Sekitar September 2007 sebuah studi melaporkan bahwa Google

mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan

Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010). Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM *Holdings, Atheros Communications,* diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*Mobile*) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru. Telepon pertama yang memakai sistem operasi Android adalah *HTC Dream*, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

1. Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

3. Android versi 1.6 (*Donut*)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet VPN*. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus pada kamera, camcorder dan galeri yang dintegrasikan pada CDMA / EVDO, 802.1x, VPN, Gestures, dan Textto- speech engine. Kemampuan dial kontak teknologi *text to change speech* (tidak tersedia pada semua ponsel).

4. Android versi 2.0/2.1 (*Eclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah

pengoptimalan hardware, peningkatan Google *Maps* 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan Bluetooth 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi *Mobile* terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah *Shazam, Backgrounds, dan WeatherBug*. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya *oleh MySpace* dan *Facebook*.

5. Android versi 2.2 (*Froyo: Frozen Yoghurt*)

Pada 20 Mei 2010, Android versi 2.2 (*Froyo*) diluncurkan. Perubahanperubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan *Adobe Flash* 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser,

pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto *update* dalam aplikasi Android *Market*.

6. Android versi 2.3 (*Gingerbread*)

Pada 6 Desember 2010, Android versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru *(reverb, equalization, headphone virtualization, dan bass boost*), dukungan kemampuan *Near Field Communication* (*NFC*), dan dukungan jumlah kamera yang lebih dari satu.

7. Android versi 3.0 (*Honeycomb*)

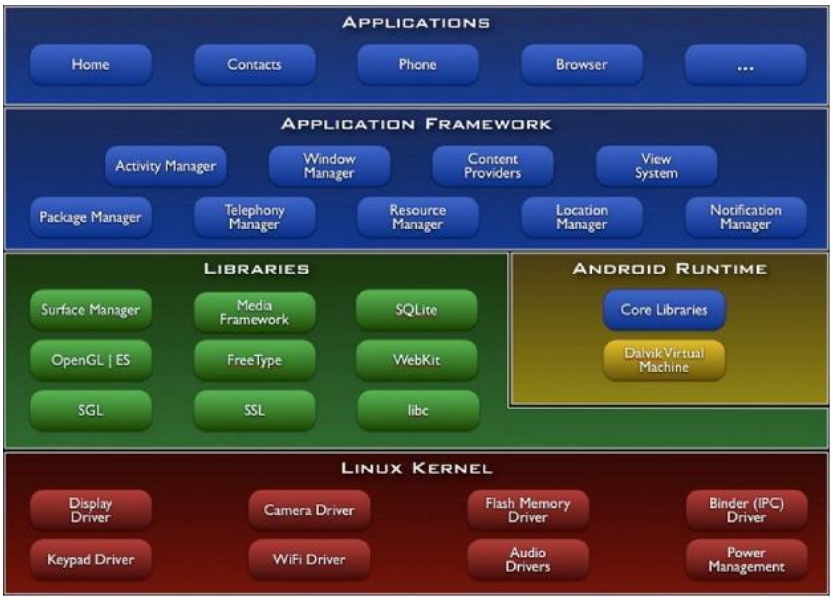
Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. *User Interfac*e pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardwar*e) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.

8. Android versi 4.0 (*ICS : Ice Cream Sandwich*)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

2.5.2. Anatomi Aplikasi Android

Dalam paket sistem operasi android tediri dari beberapa unsur seperti tampak pada gambar 2.1. Secara sederhana arsitektur android merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu Framework yang menyediakan dan mengatur alur proses aplikasi.



Gambar 2.1. Detail Anatomi Android

(Sumber : http://studyfuture.blogspot.com/2011/03/arsitektur-android.htm**)**

2.5.3. Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, *keypad, WiFi, Flash Memory, audio,* dan *IPC* (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

2.5.4. Libraries

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (*BSD*) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

1. Media *Library* untuk memutar dan merekam berbagai macam format audio dan video.
2. *Surface* Manager untuk mengatur hak akses layer dari berbagai aplikasi.
3. *Graphic Library* termasuk didalamnya SGL dan OpenGL, untuk tampilan 2D dan 3D.
4. *SQLite* untuk mengatur relasi database yang digunakan pada aplikasi.
5. *SSl* dan *WebKit* untuk browser dan keamanan internet.

Pustaka-pustaka tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan *Native Development Toolkit (NDK).*

2.5.5. Android *Runtime*

Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. Android Runtime merupakan mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam *Android Runtime* terdapat 2 bagian utama, diantaranya:

1. Pustaka Inti, android dikembangkan melalui bahasa pemrograman Java, tapi Android Runtime bukanlah mesin virtual Java. Pustaka inti android menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus android.
2. Mesin *Virtual Dalvik*, *Dalvik* merupakan sebuah mesin *virtual* yang dikembangkan oleh *Dan Bornstein* yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format *Dalvik Executable (*.dex).* Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan di

atas kernel Linux 2.6, dengan fungsi dasar seperti threading dan manajemen memori yang terbatas.

2.5.6. Aplication Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan *User Interface* dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut:

1. *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan ”*Backstack*“ untuk navigasi penggunaan.
2. *Content Providers*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.
3. *Resuource Manager*, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan file layout.
4. *Location Manager*, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.
5. *Notification Manager*, mencakup berbagai macam peringatan seperti pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada status bar.

2.5.7. *Aplication Layer*

Puncak dari diagram arsitektur android adalah lapisan aplikasi dan *widget.* Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam Android *runtime* dengan menggunakan kelas dan *service* yang tersedia pada Framework aplikasi. Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada android semua aplikasi, baik aplikasi inti *(native)* maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API *(Application Programming Interface)* yang sama.

2.6. Komponen Aplikasi Android

Fitur penting android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur scroller dan aplikasi lain telah mengembangkan fitur scroller yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup menggunakan scroller yang telah ada. Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi main() pada aplikasi lain. Sebaliknya, aplikasi

memiliki komponen penting yang memungkinkan sistem untuk memanggil dan menjalankan ketika dibutuhkan.

2.6.1. *Activity*

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *Activity* menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap *Activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antar muka pengguna yang terdiri dari *Views* dan respon terhadap *Event.* Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi dijalankan. Perpindahan antara activity dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan activity stack. Keadaan suatu *activity* ditentukan oleh posisinya dalam tumpukan *acitivity*, *LIFO (Last In First Out)* dari semua aplikasi yang sedang berjalan. Bila suatu activity baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *Back*, atau menutup *activity* yang sedang digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory Manager* android menggunakan tumpukkan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut. Ketika *activity* diambil dan disimpan dalam tumpukkan *activity* terdapat 4 kemungkinan kondisi transisi yang akan terjadi :

1. *Active*, setiap *activity* yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukkan dari pengguna. Android akan berusaha untuk membuat *activity* aplikasi ini untuk untuk tetap hidup dengan segala cara, bahkan akan menghentikan *activity* yang berada dibawah tumpukkannya jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.
2. *Paused*, dalam beberapa kasus *activity* akan terlihat tapi tidak terfokus pada kondisi inilah disebut *paused*. Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan paused, dia terlihat *active* namun tidak dapat menerima masukkan dari pengguna. Dalam kasus ekstrim, android akan menghentikan *activity* dalam keadaan paused ini, untuk menunjang sumber daya bagi *activity* yang sedang aktif.
3. *Stopped*, ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped. Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh karenanya ketika suatu *activity* dalam kondisi *stopped* maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika *activity* telah keluar atau ditutup, maka dia akan menjadi inactive.
4. *Inactive*, kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. Inactive *activity* telah ditiadakan dari tumpukan *activity* sehingga perlu restart ulang agar dapat tampil dan digunakan kembali.

5. Kondisi transisi ini sepenuhnya ditangani oleh manajer memori android. Android akan memulai menutup aplikasi yang mengandung *activity* inactive, kemudian *stopped activity*, dan dalam kasus luar biasa *paused activity* juga akan di tutup.

2.6.2. *Service*

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.

2.6.3. Intens

Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman web, dan lain sebagainya.

Intents tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi suatu hal, misal pesan masuk. Intents dapat eksplisit atau implisit, contohnya jika suatu aplikasi ingin menampilkan URL, sistem akan menentukan komponen apa yang dibutuhkan oleh Intents tersebut.

2.6.4. *Broadcast Receiver*

Broadcast Receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar Broadcast berasal dari sistem misalnya, Batre sudah hampir habis, informasi zona waktu telah berubah, atau pengguna telah merubah bahasa default pada perangkat. Sama halnya dengan service, *Broadcast Receivers* tidak menampilkan antarmuka pengguna. Namun, *Broadcast Receivers* dapat menggunakan Notification Manager untuk memberitahukan sesuatu kepada pengguna.

2.6.5. *Content Providers*

Content Providers digunakan untuk mengelola dan berbagi *database.* Data dapat disimpan dalam file sistem, dalam database *SQLite,* atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *Content Provider* memungkinkan antar aplikasi untuk saling berbagi data. Komponen ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain, sehingga mudah dalam penerapannya.

2.7. Tipe Aplikasi Android

Terdapat tiga kategori aplikasi pada android :

1. ***Foreground Activity***

   Aplikasi yang hanya dapat dijalankan jika tampil pada layar dan tetap efektif walaupun tidak terlihat. Aplikasi dengan tipe ini pasti

mempertimbangkan siklus hidup *activity*, sehingga perpindahan antar *activity* dapat berlangsung dengan lancar.

2. Background Service

Aplikasi yang memiliki interaksi terbatas dengan user, selain dari pengaturan konfigurasi, semua dari prosesnya tidak tidak tampak pada layar. Contohnya aplikasi penyaringan panggilan atau sms auto respon.

3. Intermittent Activity

Aplikasi yang masih membutuhkan beberapa masukkan dari pengguna, namun sebagian sangat efektif jika dijalankan di background dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu.

Contohnya pemutar musik. Untuk aplikasi yang kompleks akan sulit untuk menentukan kategori aplikasi tersebut apalagi aplikasi memiliki ciri-ciri dari semua kategori. Oleh karenanya perlu pertimbangan bagaimana aplikasi tersebut digunakan dan menentukan kategori aplikasi yang sesuai.

2.7.1. Siklus Hidup Android

Siklus hidup aplikasi android dikelola oleh sistem, berdasarkan kebutuhan pengguna, sumberdaya yang tersedia, dan sebagainya. Misalnya Pengguna ingin menjalankan *browser web*, pada akhirnya sistem yang akan menentukan menjalankan aplikasi. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika pengguna ketika itu sedang menjalankan sebuah *Activity*, maka sistem akan memberikan perioritas utama untuk aplikasi yang tersebut.

Sebaliknya, jika suatu *Activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *Activity* yang prioritas rendah akan ditutup.

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin virtual pengolah sendiri, dengan ini melindungi penggunaan memori pada aplikasi. Selain itu juga android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karenanya android sangat sensitive dengan siklus hidup aplikasi dan komponen-komponennya. Perlu adanyapenanganan terhadap setiap kondisi agar aplikasi menjadi stabil.

2.7.2. Kelebihan Android

Sudah banyak platform untuk perangkat selular saat ini, termasuk didalamnya *Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM)*, dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut :

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan Linux dan open source. Pembuat perangkat menyukai hal ini karena dapat membangun platform yang sesuai yang diinginkan tanpa harus membayar royality. Sementara pengembang software menyukai karena android dapat digunakan diperangkat manapun dan tanpa terikat oleh vendor manapun.

2. Arsitektur komponen dasar android terinspirasi dari teknologi internet Mashup. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan service, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, *database SQL,* browser dan penggunaan peta. Semua itu sudah tertanam pada android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suarat terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh Flash menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun game yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin *virtual Dalvik*, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard,* layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.8. MySQL

MySQLadalah sebuah aplikasi Relational Database Management Server (RDBMS) yang menggunakan bahasa SQL (Structured Query Language) yaitu bahasa standar yang digunakan untuk pemrograman database.Atau disebut juga sebagai perangkat lunak database server.

Dengan menggunakan database ini, data semakin aman dan berdaya guna. Database ini juga banyak dipakai pada web database sehingga data semakin terintegrasi antara database dekstop dengan database web. Serta Server data dapat diakses secara cepat oleh banyak pemakai secara bersamaan.

Keunggulan dari MySQL adalah:

Bersifat open source.

1. Sistem yang digunakan oleh perangkat lunak ini tidak memberatkan kerja dari server, karena dapat bekerja di background.
2. Mempunyai koneksi yang stabil dan kecepatan yang tinggi.

2.8.1. Kelebihan MySQL

MySQL merupakan database server yang memiliki konsep database modernMySQL mempunyai beberapa kelebihan antara lain :

1. Portabilitas.

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga dll

2. Open source

MySQL didistribusikan secara geratis (open source) di bawah lisensi GPL sehingga dapat di gunakan tanpa di puzngut sepeserpun

3. Multiuser

   MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. Performance tuning

   MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Column types

   MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Command dan function\

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan where dalam Query

2.9. Phpmyadmin

PhpMyadmin adalah perangkat lunak yang di tulis dalam bahasa pemrograman php yang digunakan untuk menangani administrasi melalui jejaring jagat lembar (world wide web) PhpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (fields), relasi (relations), indeks, pengguna (users), perijinan (permissions), dan lain-lain

dan Pada dasarnya mengelola basis data yang ada di komputer. Untuk membukanya, buka browser lalu ketikka alamat *http://localhost/phpmyadmin*, maka akan muncul halaman phpMyAdmin. Di situ nantinya seseorang bisa membuat (create) basis data baru, dan mengelolanya.

2.10. Bahasa Pemrograman JAVA

Java merupakan perangkat lunak produksi Sun Microsistem Inc. untuk pemrograman beberapa tujuan (*multi purpose*), dapat berjalan di beberapa sistem operasi (*multiplatform*), mudah dipelajari dan powerful. Aplikasi-aplikasi yang dapat dibuat dengan Java, meliputi pemrograman web (*web programming*), Pemrograman Desktop (*Desktop programming*), Pemrograman mobile/*handphone* (*mobile programming*).

Sun Microsistem sendiri mendeskripsikan Java sebagai bahasa pemrograman yang sederhana, *garbage collected*, *robust*, dan dapat diperluas. Sifat yang dideskripsikan oleh Sun Microsistem dapat dipaparkan sebagai berikut:

1. Sederhana

   Pengembang Java banyak membuang fitur-fitur yang tidak diperlukan seperti yang dimiliki bahasa pemrograman tingkat tinggi lainnya.

2. *Garbage collected* (Pembuang Sampah)

   Program Java menyingkirkan sendiri „sampah-sampah‟ yang tidak berguna, artinya program tidak perlu menghapus objek-objek yang dialokasikannya di memori.

3. *Robust* (Tangguh)

   Karena interpretasi Java memeriksa seluruh akses sistem yang dilakukan program, maka program Java tidak akan membuat sistem menjadi *crash*.

4. Dapat diperluas

   Program Java mendukung metode *native* yakni fungsi-fungsi yang ditulis dalam bahasa latin, biasanya C++.

Java membagi versi programnya ke dalam tiga kelompok besar, yaitu :

a. Java 2 Standar Edition (J2SE) untuk konsentrasi pada PC

b. Java 2 Enterprise Edition (J2EE) untuk konsentrasi pada aplikasi server besar

c. Java 2 Micro Edition (J2ME) untuk konsentrasi pada mobile

2.11. Metode Analisis dan Perancangan Objek

Metode pendekatan sistem yang akan digunakan penyusun adalah metode analisis dan perancangan terstruktur. Pendekatan sistem yang penyusun ambil adalah berorientasi pada objek, dimana pada analisis dan perancangan terdapat :

2.11.1. Activity Diagram

Activity diagram menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti

2.11.2. Use Case Diagram

Use case diagram menggambarkan interaksi antara actor dengan proses atau sistem yang dibuat. Use case diagram mempunyai beberapa bagian penting seperti :

a. Actor adalah bagian dari use case yang bertindak sebagai subjek (pelaku) dalam suatu proses
b. Use case adalah proses-proses yang terjadi dalam suatu software. Use case juga menggambarkan apa yang sedang dilakukan oleh Actor
c. Relasi menggambarkan hubungan antara actor dan use case

2.11.3. Collaborasi Diagram

Collaboration Diagram hampir sama dengan sequence diagram tetapi berbeda pada objek yang di titik tekankan , collaboration lebih menekankan pada pemunculan objek itu sendiri sedangkan sequence diagram lebih pada penyampaian message dengan parameter waktu.

2.11.4. Sequence Diagram

Sequence diagram merupakan interaksi antara objek-objek dalam suatu sistem dan terjadi komunikasi yang berupa pesan (message) serta parameter waktu.

2.11.5. Class Diagram

Class diagram menggambarkan interaksi antar class serta atribut-artibut yang melekat pada class tersebut.

2.11.6. Tabel Spesifikasi Kelas

Tabel spesifikasi kelas merupakan rincian dari diagram spesifikasi kelas. Tabel ini menjelaskan nama kelas, peran, atribut dan method yang dimiliki oleh kelas yang bersangkutan.

2.11.7. Package Diagram

Package Diagram adalah diagram yang menggambarkan bagaimana elemen model diorganisasikan atau dikelompokkan ke dalam packages.

2.11.8. Deployment Diagram

Deployment diagram menggambarkan secara fisik antara software dan hardware.

2.12. Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat dari Eclipse:

1. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Mulit-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, *test* perangkat lunak, pengembangan web, dan lain sebagainya. Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in.

2.12.1. Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US$ 40 juta[1] untuk pengembangannya. Semenjak itu konsursium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

2.12.2. Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP). Berikut ini adalah komponen yang membentuk RCP:

1. Core platform
2. OSGi
3. SWT (Standard Widget Toolkit)
4. Jface
5. Eclipse Workbench

Secara standar Eclipse selalu dilengkapi dengan JDT (Java Development Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru. Eclipse beserta plug in-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya.

2.12.3. Perkembangan Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous Release. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah *deployment* dan *maintenance* untuk sistem enterprise. Adapun versi eclipse yang telah diluncurkan yaitu :

1. Eclipse 3.0
2. Eclipse 3.1
3. Callisto
4. Europa
5. Ganymede
6. Galileo
7. Helios
8. Indigo
9. Juno

2.13. Software Development Kit (SDK)

Software Development Kit (SDK) adalah suatu kit atau *library* dari bahasa pemrograman untuk pengembangan atau pembangunan suatu perangkat lunak dan biasanya SDK terdiri dari kumpulan *tools* yang dibutuhkan. Misalnya bahasa

pemrograman java, mempunyai SDK yang berisi suatu *library* yang dapat digunakan untuk membuat suatu aplikasi berbasis java.

2.14. Java Development Kit (JDK)

Java Development Kit (JDK) adalah sekumpulan perangkat lunak yang dapat kamu gunakan untuk mengembangkan perangkat lunak yang berbasis Java, Sedangkan JRE adalah sebuah implementasi dari Java Virtual Machine yang benar-benar digunakan untuk menjalankan program java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber kompiler java, bundling, debuggers, development libraries dan lain sebagainya. Perbedaan JDK dengan SDK (Software Development Kit) yaitu JDK adalah sebuah SDK tetapi sebuah SDK tidak harus menjadi sebuah JDK.

2.15. ADT

Android Development Tools (ADT) adalah plugin untuk Eclipse Intergrated Development Environment (IDE) yang dirancang untuk memberikan lingkungan yang terpadu di mana untuk membangun aplikasi Android.

ADT memperluas kemampuan Eclipse untuk membiarkan para developer lebih cepat dalam membuat proyek baru Android, membuat aplikasi UI, menambahkan komponen berdasarkan Android Framework API, debug aplikasi dalam pengunaan Android SDK, dan membuat file APK untuk mendistribusikan aplikasi.

Mengembangkan aplikasi di Eclipse dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi android, karena banyak kemudahan-kemudahan sebagai tools yang terintegrasi seperti, custom XML editor, dan debug panel *ouput*. Selain itu ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.