BAB II

LANDASAN TEORI

2.1 Aplikasi

Sumber Wikipedia (01:06:2013) 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.

2.2 Tenses

Menurut A.Faidlal Rahman Ali (2007:246) "*Tenses* dapat diartikan sebagai suatu gambaran atau penjelasan kapan suatu peristiwa, kejadian, atau tindakan terjadi di dalam kalimat yang sesuai dengan keadaan, antara lain : *sekarang, lampau* atau *akan datang*".

Berdasarkan *waktu kejadian*, *tenses* dalam bahasa Inggris dapat dibagi menjadi 4(empat) macam, yang dikenal dengan "*Four Fundamental Tenses*", yaitu :

a. Present Tenses : Untuk waktu yang sedang terjadi.
b. Past Tenses : Untuk waktu yang sudah terjadi.
c. Future Tenses : Untuk waktu yang akan terjadi
d. Past Future Tenses : Untuk waktu yang akan terjadi di masa lampau.

Sedangkan berdasarkan *sifat kejadian*, dapat dikelompokkan menjadi 4 (empat) bagian, yaitu:

a. *Simple Tenses* : Untuk kejadian yang tidak sedang dalam proses.
b. *Continous Tenses* : Untuk kejadian yang sedang dalam proses.
c. *Perfect Tenses* : Untuk kejadian yang sudah selesai.
d. *Perfect Continous Tenses* : Untuk kejadian yang dimulai sebelum suatu saat tertentu, dan pada saat tertentu kejadian masih berlangsung.

Kalau digabungkan kedua macam *tenses* di atas, yaitu yang berdasarkan *waktu kejadian* dan *sifat kegiatan*, maka akan menjadi enam belas waktu (tenses) itu antara lain : (A.Faidlal Rahman Ali, SE. Par, Fundamentals of English Grammar, Yogyakarta, Pustaka Widyatama, 2007)

1. Present Tense, dibagi menjadi 4 (empat) macam yaitu :
    a. Simple Present Tense
    b. Present Continous Tense
    c. Present Perfect Tense
    d. Present Perfect Continous Tense
2. Past Tense, dibagi menjadi 4 (empat) macam yaitu :
    a. Simple Past Tense
    b. Past Continous Tense
    c. Past Perfect Tense
    d. Past Perfect Continous Tense
3. Future Tense, dibagi menjadi 4 (empat) macam yaitu :
    a. Simple Future Tense
    b. Future Continous Tense

c. Future Perfect Tense

d. Future Perfect Continous Tense

4. Past Future Tense, dibagi menjadi 4 (empat) macam yaitu :

a. Past Future Tense

b. Past Future Continous Tense

c. Past Future Perfect Tense

d. Past Future Perfect Continous Tense

2.3 Interactive Learning

Interaktif itu sendiri bersifat saling melakukan aksi, antar-hubungan, saling aksi. Learning dalam bahasa Indonesia adalah pembelajaran. Pembelajaran adalah proses interaktif peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. Pembelajaran merupakan bantuan yang diberikan pendidik agar dapat terjadi proses perolehan ilmu dan pengetahuan, penguasaan kemahiran dan tabiat, serta pembentukan sikap dan kepercayaan pada peserta didik. dengan kata lain, pembelajaran adalah proses untuk membantu peserta didik agar dapat belajar dengan baik.

interaktif learning atau pembelajaran interaktif adalah suatu cara atau teknik pembelajaran yang digunakan guru pada saat menyampaikan bahan pelajaran dimana guru adalah pemeran utama dalam menciptakan situasi interaktif yang edukatif, yakni interaktif guru dengan siswa, siswa dengan siswa, dan dengan sumber pembelajaran dalam menunjang tercapainya tujuan belajar

2.4 Android

Menurut Nazruddin Safaat H (2012:1) “Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka”.

2.4.1 Sejarah Android

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuat lisensi perangkat lunak dan *open platform* perangkat selular.

Sekitar September 2007 Google mengenalkan *Nexus One*, salah satu jenis *smartphone* yang menggunakan Android sebagai system operasinya. Telepon selular ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 selular ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010.

Pada 9 Desember 2008, diumumkan anggota bbaru yang bergabung dalam program kerja Android ARM Holdingd, Atheros Communicatons, 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 monile uang merupakan modifikasi kernel Linux 2.6. sejak Android dirilis telah dilakukan berbagau pembaruan berupa perbaikan bug dan penambahan fitur baru.

Pada masa saat ini sebagian besar *vendor-vendor smartphone* sudah memproduksi *smartphone* berbasis android, *vendor-vendor* ini antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, Acer, Philips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi *vendor smartphone* di dunia yang memproduksi android. Hal ini, karena android itu adalah system operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh *vendor* manapun.

2.4.2 Arsitektur Android

Secara garis besar arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

1. *Applications* dan *Widgets*

   Applications dan *Widgets* ini adalah *layer* dimana kita berhubungan dengan aplikasi saja dimana biasanya kita *download* aplikasi kemudian dengan kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semuanya aplikasi ditulis menggunakan bahasa pemrograman Java.

2. *Applications Frameworks*

   Android adalah "Open Development Platform" yaitu Android menawarkan kepada pengembangnya atau member kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur *alarm*, dan

menambahkan status *notifications*, dan sebagainya. Pengembang memliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*). Komponennya meliputi *views, Content Provider, Resource Manager, Notification Manager, Activity Manager.*

3. *Libraries*

   Libraries ini adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas kernel, *Layer* ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL, serta:

   a. *libraries* media untuk pemutaran media audio dan video,
   b. *libraries* untuk manajemen tampilan,
   c. *libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D,
   d. *libraries* SQLite untuk dukungan database,
   e. *libraries* LiveWebcore mencakup modern web browser dan *security*,
   f. *libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's.

4. *Android Run Time*

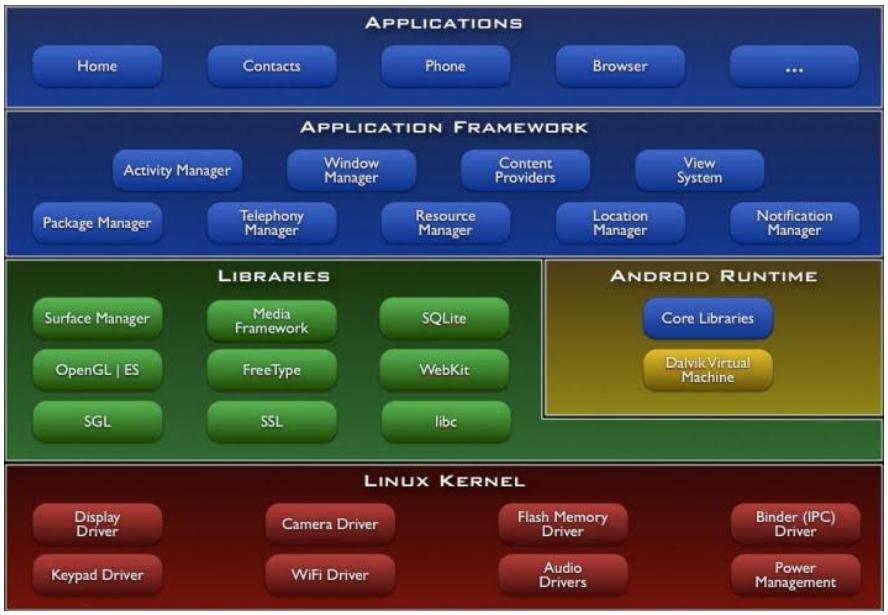
   Layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan Implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android *Run Time* dibagi menjadi dua bagian yaitu:

a. *Core Libraries* merupakan Aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan virtual Machine Java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menerjemahkan bahasa java/c yang ditangani oleh *Core Libraries*.

b. *Dalvik Virtual machine* merupakan Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

5. Linux *Kernel*

Linux Kernel adalah *layer* di mana inti dari *operating* sistem dari Android itu berada. Berisi *file-file system* yang mengatur sistem *processing, memory, resourse, drivers*, dan sistem-sistem operasi android lainnya. Linux *kernel* yang digunakan android adalah linux kernel *release* 2.6.



Gambar 2.1 Arsitektur Android

2.4.3 Komponen Aplikasi Android

Aplikasi Android ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi di mana prosesnya di package oleh *tools* yang dinamakan "apt tools" ke dalam paket Android sehingga menghasilkan *file* dengan ekstensi apk. *File* apk itulah yang kita sebut dengan aplikasi, dan nantinya dapat di install di perangkat *mobile.* Ada enam jenis komponen pada aplikasi Android yaitu:

a. *Activities*

   Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Banyaknya *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut.

b. *Service*

   Service tidak memiliki *Graphic User Interface (GUI)*, tetapi *service* berjalan secara *background*. *Service* dijalankan pada *thread* utama dari proses aplikasi.

c. *Broadcast Receiver*

   Broadcast receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi.

d. *Content Provider*

   Content provider membuat kumpulan suatu aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file system seperti *database* SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*.

2.4.4 Versi Android

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2010 diperkirakan hamper semua *vendor* seluler di dunia menggunakan Android sebagai *operating system*. Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut:

1. Android versi 1.1

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

2. Android versi 1.5 (*Cupcake*)

   Pada pertengahan Mei 2009, Google kembali merilis teleponselular dengan menggunakan Android dan SDK (*Software development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaharuan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, meng-*upload 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 system.

3. Android versi 1.6 (*Donut*)

   Donut (versi 1.6) dirilis pada September dengan menapilkan proses pencarian yang lebih baik dibandu=ingkan sebelumnya, penggunaan

baterai indicator dan control applet VPN. Fitur lainnya adalah galeri yang memungkinan pengguna untuk memilih foto yang akan dihapus; kamera,, *camcorder*, dan *text-to-speech engine*, kemampuan *dial* kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel, pengadaan resolusi VWGA.

4. Android versi 2.0/2.1 (*Éclair*)

   Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Éclair), 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 manera 3,2MP, *digital Zoom*, dan *Bluetooth* 2.1.

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

   Pada bulan Mei 2010 Android versi 2.2 Rev 1 diluncurkan. Android inilah yang sekarang sangat banyak beredar di pasaran, salah satunya inilah dipakai di Samsung FX tab yang sudah dip saran. Fitur yang tersedia di Android versi inii sudah kompleks di antaranya adalah:

   a. Kerangka aplikasi memungkinan pengguna dan penghapusan komponen yang tersedia.
   b. *Dalvik Virtual Machine* dioptimalkan untuk perangkat mobile.
   c. Grafik, grafik di 2D dan grafis 3D berdasarkan *libraries* OpenGL
   d. SQLite untuk penyimpanan data.
   e. Mendukung media : *audio, video*, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
   f. GSM, *Bluetooth*, EDGE, 3G, dan WiFi *(hardware independent*)

g. Kamera, *Global Positioning system* (GPS), kompas, dan *accelerometer* (tergantung *hardware*).

6. Android versi 2.3 (*Gingerbread*)

   Android versi 2.3 diluncurkan pada Desember 2010, hal-hal yang direvisi dari versi sebelumnya adalah kemampuan adalah SIP-*based VoIp, Near Field Communications* (NFC), *Gyroscope* dan sensor, *Multiple cameras support, mixable audio effect, Downnload manager*.

7. Android versi 3.0 (*Honeycomb*)

   Dirilis Februari 2011 sebagai amdroid 3.0 revisi 1 serta android versi 3.0 revision 2 telah dirilis pada juli 2011.

8. Android versi 3.1

   Dirilis Mei 2011 sedangkan android 3.1 versi 2 juga dirilis mei 2011, serta android 3.1 revision 3 dirilis pada Juli 2011.

9. Android versi 3.2

   Dirilis Juli 2011

10. Android versi 4.0

    Dirilis November 2011

2.5 Konsep Perancangan Berorientasi Objek

Untuk memahami konsep-konsep dalam perancangan berorientasi objek dapat dilihat pada subjudul berikut ini:

2.5.1 Object / Objek

Menurut Yasmi Afrizal dan Wahyuni (2013:165) Objek adalah orang, tempat, benda, kejadian atau konsep-konsep yang ada di dunia nyata

dan penting bagi suatu aplikasi. Sebuah objek adalah Entitas yang memiliki Identitas, States (keadaan sesaat) dan Behavior (perilaku).

State sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam **Atribut** atau property. Behavior sebuah objek mendefinisikan bagaimana sebuah objek bertindak/bereaksi yang dinyatakan dlam **Operation.** Satu objek dapat diturunkan menjadi objek dalam bentuk lain, kemudian saling mengkait menyusun sesuatu yang lebih rumit.

2.5.2 Class / Kelas

Class adalah kumpulan atau himpunan objek-objek yang sejenis, memiliki kesamaan atribut/property, perilaku, serta relasi dengan objek lain yang mirip. Notasi kelas digambarkan dengan kotak, dengan nama kelas didalamnya ditulis menggunakan huruf besar di awal kata. Bila sebuah kelas memiliki 2 suku kata atau lebih, maka penulisannya disatukan tanpa spasi dengan huruf awal tiap suku menggunakan huruf besar.

2.5.3 Attribute / Atribut

Attribute adalah data yang dimiliki suatu objek atau property dari sebuah Class yang menggambarkan batas nilai yang mungkin ada pada objek dari kelas. Sebuah bisa memiliki nol atau lebih atribut. Notasi atribut digambarkan dengan kotak dibawah class, dengan nama atribut didalamnya ditulis menggunakan huruf kecil. Jika sebuah atribut memiliki 2 atau lebih suku kata, maka semua suku kata ditulis disatukan tanpa spasi, awal suku kata pertama dengan huruf kecil dan awal suku kata berikutnya dengan huruf besar.

Notasi atribut dapat ditambahkan informasi dengan tipe-tipe atribut tersebut. Penulisan tipe pada atribut dipisahkan denngan tanda titik dua (:), tipe ditambahkan berupa String, Floating-Point number, Integer dan Boolean.

2.5.4 Operation / Operasi

Operation adalah sesuatu yang bisa dilakukan oleh sebuah class. Notasi penulisannya sama dengan atribut. Bagian operation ini juga bisa diberikan tambahan informasi, yaitu dengan menambahkan parameter yang akan dilakukan operation dalam tanda kurung. Contoh parameternya adalah function.

2.5.5 Inheritance / Pewarisan

Inheritance atau pewarisan memungkinkan dibuat class yang menyerupai class lain yang telah ada sebelumnya, tetapi masih memiliki beberapa sifat induknya. Seperti sebuah mobil biasa, anda dapat membuat mobil balap serta mobil angkutan umum. Prosesnya adalah mengubah sifat dari mobil biasa tersebut.

2.5.6 Polymorphisme / Kebanyakrupaan

Polymorphisme adalah object yang memiliki fungsi sama dengan object dasar tetapi memiliki satu atau lebih sifat berbeda atau dengan kata lain Polymorphisme adalah pemisah secara jelas diantara seubsistem yang berbeda. Sebagai contoh misalkan sebuah kelas memiliki operasi ‘OPEN’, operasi open ini bisa dipakai untuk membuka pintu, membuka buku, membuka baju dan lainnya. Meskipun ‘OPEN’ memiliki tujuan yang sama, tapi apa yang dilakukannya berbeda.

2.5.7 Encapsulation / Pembungkusan

Encapsulation sering disebut dengan penyembunyian informasi (*Hidding*), suatu konsep berdasarkan fakta di dunia nyata yang menyatakan bahwa segala sesuatu tidak perlu diperlihatkan. Misalnya kita perlu tahu apa yang dilakukan sistem ketika kita menekan remote untuk menghidupkan televisi.

2.5.8 Responsibilities / Tanggung Jawab

Responsibilities adalah model tambahan yang digambarkan pada bagian bawah suatu kelas setelah bagian operasi digunakan untuk menjelaskan pernyataan-pernyataan yang bisa dilakukan oleh kelas tersebut.

2.5.9 Unified Modelling Language (UML)

Menurut Yasmi Afrizal & Wahyuni (2013:183) "Merupakan pengganti metode analisis berorientasi object dan design berorientasi object (OOA & OOD) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an". (Rekayasa Perangkat Lunak, Yasmi Afrizal & Wahyuni, Unikom, Bandung, 2013) UML merupakan bahasa standar untuk penulisan Blueprint Software yang digunakan untuk Visualisasi (*Visualize*), Spesifikasi (*Specify*), Pembentukan (*Construct*) dan Pendokumentasian (*Documentation*) alat-alat dari sistem perangkat lunak.

Tujuan penggunaan UML adalah, sebagai berikut :

a. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.
b. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

Keunggulan menggunakan UML dibandingkan menggunakan metodologi terstruktur :

a. Uniformity

   Pengembang cukup menggunakan 1 metodologi dari tahap analisis hingga perancangan. Memungkinkan merancang komponen antarmuka secara integrasi bersama perancangan PL dan perancangan struktur data.

b. Understandability

   Kode yang dihasilkann dapat diorganisasi kedalam kelas-kelas yang berhubungan dengan masalah yang sesungguhnya sehingga lebih mudah dipahami.

c. Stability

   Kode program yang dihasilkan relative stabil sepanjang waktu, karena mendekati permaslahan yang sesungguhnya.

d. Reusability

   Dengan metodologi berorientasi objek, dimungkinkan pengguna ulang kode, sehingga pada akhirnya akan sangat mepercepat waktu pengembangan perangkat lunak (atau sistem informasi).

2.6 SQLite

Menurut Nazruddin Safaat H (2012:171) SQLite adalah salah satu *software* yang *embedded* yang sangat populer, kombinasi SQL *interface* dan penggunaan *memory* yang sangat sedikit dengan kecepatan yang sangat cepat. SQLite diandroid termasuk ke dalam Android *runtime*, sehingga setiap versi android dapat membuat *database* dengan SQLite.

Dalam sistem android memiliki beberapa teknik untuk melakukan penyimpanan data. Teknik yang umum digunakan adalah sebagai berikut :

1. **Shared Preferences** yaitu menyimpan data beberapa nilai (*value*) dalam bentuk *groups key* yang dikenal dengan *preferences*.
2. **Files** yaitu menyimpan data dalam file, dapat berupa menulis ke file atau membaca dari file.
3. **SQLite Databases**, yaitu menyimpan data dalam bentuk *Databases*.
4. **Content Providers**, yaitu menyimpan data dalam bentuk *content providers service*.