BAB II

LANDASAN TEORI

2.1 Pengertian Anak Usia Prasekolah

Pengertian anak usia prasekolah (Sillahi : 2005) adalah “anak usia 4 sampai 6 tahun dimana pada masa ini anak telah mencapai kematangan dalam berbagai macam fungsi motorik dan diikuti dengan perkembangan intelektual dan sosioemosional”. Selain itu, imajinasi intelektual dan keinginan anak untuk mencari tahu dan bereksplorasi terhadap lingkungan juga merupakan cirri utama anak pada usia ini.

Anak usia dini adalah masa-masa yang butuh perhatian dan kasih sayang total dari kedua orngtuanya. Apabila anak diasuh dengan pola asuh demokratis maka tumbuh kembang anak akan lebih baik. Karena jika pola asuh yang diterpkan orangtua kepada anaknya demokratis yang selalu memberikannya kebebasan beraktivitas tetapi tetap diarahkan orangtuanya, anak akan cenderung bebas melakukan aktivitas pembelajaran dalam dirinya tetapi bertanggungjawab akan akibat yang akan diterma kelak, pemberani, mempunyai rasa percaya diri yang tinggi, tidak tergantung orangtuanya, dan riang gembira.

Sebaliknya jika pola asuh orangtuanya kepada anaknya otoriter anak akan cenderung takut untuk melakukan sesuatu untuk perkembangannya yang lebih baik karena apapun aktivitas anak selalu dikekang dan orang tua terlalu takut

membebaskan anaknya beraktivitas. Anaka akan cenderung penakut, tidak percaya diri, tergantung kepada orangtua, cenderung pendiam, pemurung, tidak mudah tersenyum, dan tidak gembira.

2.2 Pengertian Aplikasi

Aplikasi adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel.
Menurut Dhanta (2009:32),

Sedangkan menurut aplikasi adalah penerapan, penggunaan atau penambahan Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan software yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.
Anisyah (2000:30)

2.3 Pengertian Tutorial

Tutorial (*tutoring*) adalah bantuan atau bimbingan belajar yang bersifat akademik oleh *tutor* kepada anak (*tutee)* untuk membantu kelancaran proses belajar madiri anak secara perorangan atau kelompok berkaitan dengan materi ajar. Tutorial dilaksanakan secara tatap muka atau jarak jauh berdasarkan konsep belajar mandiri.

Konsep belajar mandiri dalam tutorial mengandung pengertian, bahwa tutorial merupakan bantuan belajar dalam upaya memicu dan memacu

kemandirian, disiplin, dan inisiatif diri anak dalam belajar dengan minimalisasi intervensi dari pihak pembelajar/tutor. Tutorial tidak ada, jika kemandirian tidak ada. Jika anak tidak, dan datang ke tutorial dengan 'kepala kosong', maka yang terjadi adalah "pembelajaran" biasa, bukan tutorial.

Peran utama tutor dalam tutorial adalah:

1. "***pemicu***" dan "***pemacu***" kemandirian belajar individu, berpikir dan berdiskusi; dan
2. "***pembimbing, fasilitator***, dan ***mediator***" individu dalam membangun pengetahuan, nilai, sikap dan keterampilan akademik dan profesional secara mandiri, dan/atau dalam menghadapi atau memecahkan masalah-masalah dalam belajar mandirinya; memberikan bimbingan dan panduan agar anak secara mandiri memahami materi; memberikan umpan balik kepada anak-anak secara tatap muka atau melalui alat komunikasi; memberikan dukungan dan bimbingan, termasuk memotivasi dan membantu anak mengembangkan keterampilan belajarnya.

Agar tutorial tidak terjebak dalam situasi pembelajaran biasa, terbina hubungan bersetara, mampu memainkan peran-peran di atas, dan tutorial berjalan efektif, tutor perlu menyiapkan pertanyaan-pertanyaan yang berfungsi untuk:

1. membangkitkan minat anak terhadap materi yang sedang dibahas,
2. menguji pemahaman anak terhadap materi,
3. memancing anak agar berpartisipasi aktif dalam kegiatan tutorial,
4. mendiagnosis kelemahan-kelemahan anak, dan
5. menuntun anak untuk dapat menjawab masalah yang sedang dihadapi

Tutor juga menstimulasi anak untuk terlibat aktif dalam pembahasan masalah yang ditemukan anak dalam mempelajari materi. Untuk mendukung pelaksanaan peran dan fungsi-fungsi di atas, tutor perlu menguasai secara trampil sejumlah keterampilan dasar tutorial, yakni:

1. membuka dan menutup tutorial;
2. bertanya lanjut;
3. memberi penguatan;
4. mengadakan variasi;
5. menjelaskan;

2.4 Pengertian pembelajaran

Menurut Trianto (2010:17) "Pembelajaran merupakan aspek kegiatan manusia yang kompleks, yang tidak sepenuhnya dapat dijelaskan". Pembelajaran secara simpel dapat diartikan sebagai produk interaksi berkelanjutan antara pengembangan dan pengalaman hidup. Pembelajaran dalam makna kompleks adalah usaha sadar dari seorang guru untuk membelajarkan siswanya (mengarhkan interaksi siswa dengan sumber belajar lainnya) dalam rangkan mencapai tujuan yang diharapkan.

2.5 Pengertian Huruf Alfabeth

Istilah alphabet sebetulnya berasal dari bahasa Semit. Istilah ini terdiri dari dua kata, yaitu aleph yang berarti 'lembu jantan' dan kata beth yang berarti 'rumah'. Konotasi pictografis dari pengertian kedua kata ini menjadi sebutan untuk menunjukkan huruf pertama a (aleph) dan b (beth) dalam urutan huruf-huruf semit (Mario Pei,2008:176)

Macam- Macam huruf Alfabeth :

1. Huruf A / a dibaca a
2. Huruf B / b dibaca be
3. Huruf C / c dibaca ce
4. Huruf D / d dibaca de
5. Huruf E / e dibaca e
6. Huruf F / f dibaca ef
7. Huruf G / g dibaca ge
8. Huruf H / h dibaca ha
9. Huruf I / I dibaca i
10. Huruf J / j dibaca je
11. Huruf K / k dibaca ka
12. Huruf L / l dibaca el
13. Huruf M / m dibaca em
14. Huruf N / n dibaca en
15. Huruf O / o dibaca o

16. Huruf P / p dibaca pe
17. Huruf Q / q dibaca qi
18. Huruf R / r dibaca er
19. Huruf S / s dibaca es
20. Huruf T / t dibaca te
21. Huruf U / u dibaca u
22. Huruf V / v dibaca ve
23. Huruf W / w dibaca we
24. Huruf X / x dibaca eks
25. Huruf Y / y dibaca ye
26. Huruf Z / z dibaca zet

2.6 Pengertian Angka

Angka adalah suatu lambang yang digunakan untuk menjelaskan bilangan, angka juga bisa di definisikan gagasan abstrak yang digunakan untuk menghitung dan mengukur. Angka adalah pelambang jumlah yang telah diakui manusia untuk dipergunakan untuk menghitung dan mengukur.

Cara pembacaan angka :

1. Angka 1 dibaca satu
2. Angka 2 dibaca dua
3. Angka 3 dibaca tiga
4. Angka 4 dibaca empat

5. Angka 5 dibaca lima
6. Angka 6 dibaca enam
7. Angka 7 dibaca tujuh
8. Angka 8 dibaca delapan
9. Angka 9 dibaca Sembilan
10. Angka 10 dibaca sepuluh

2.7 Pengertian Bangun Datar

Bentuk adalah seluruh informasi geometris yang akan tidak berubah ketika parameter lokasi, skala, dan rotasinya dirubah. Bentuk sederhana dapat diterangkan oleh teori benda geometri dasar (dua dimensi) misalnya titik, garis, kurva, bidang (misal, persegi atau lingkaran), atau bisa pula diterangkan oleh benda padat (tiga dimensi) seperti kubus, atau bola.

2.8 Pengertian Warna

Bagian sinar dalam spektrum yang tergantung pada gelombang cahayanya. Dalam dunia desain, warna bisa berarti pantulan tertentu dari cahaya yang dipengaruhi oleh pigmen yang terdapat di permukaan benda. Misalnya pencampuran pigmen magenta dan cahaya dengan proporsi tepat dan disinari cahaya putih sempurna akan menghasilkan sensasi mirip warna merah

Menurut (Wirania Swasty: 2010; 6) Secara objektif atau fisik, warna adalah sifat cahaya yang dipancarkan. Sementara secara subjektif atau psikologis, warna adalah sebagian dari pengalaman indra penglihatan

2.9 Pengertian Aplikasi Mobile

Kata mobile mempunyai arti bergerak atau berpindah. Sehingga diperoleh pengertian bahwa aplikasi bergerak merupakan aplikasi yang dapat dijalankan walaupun pengguna berpindah atau karena pengguna berpindah.

Pemrograman aplikasi bergerak tidak banyak berbeda dengan pemrograman aplikasi bergerak tidak banyak berbeda dengan pemrograman konvensional pada PC. Aspek karakteristik dari perangkat bergerak sering mempengaruhi arsitektur dan implementasi dari aplikasi tersebut. Dalam pemrograman aplikasi bergerak berbagai aspek teknis perangkat lebih menonjol karena memiliki banyak keterbatasan dibandingkan computer konvensional atau PC.
(Budi Darytamo dkk, 2007:1)

2.10 Metode Pendekatan Sistem

Metode pendekatan sistem yang digunakan oleh penulis adalah pendekatan dengan *Object Oriented* yang menggunakan OOAD (*Object Oriented Analisis and Design*).

2.10.1 Pengertian OOAD

OOAD adalah metode analisis yang memeriksa requirements dari sudut pandang kelas kelas dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek system atau subsistem. OOAD merupakan cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas.

2.10.2 Objek (Object)

Objek adalah benda secara fisik dan konseptual yang ada di sekitar kita. Sebuah objek memiliki keadaan sesaat yang disebut state. State dari sebuah objek adalah kondisi dari objek atau himpunan keadaan yang menggambarkan objek tersebut. State dinyatakan dengan nilai dari atribut objeknya. Atribut adalah nilai internal suatu objek yang mencerminkan karakteristik objek, kondisi sesaat, koneksi dengan objek lain dan identitas. Behaviour (perilaku objek) mendefinisikan bagaimana sebuah objek bertindak dan memberi reaksi. Behaviour ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan oleh objek tersebut, yang dicerminkan oleh interface, service, dan method dari objek tersebut. Interface adalah pintu untuk mengakses service dari objek. Service adalah fungsi

yang dapat dikerjakan oleh sebuah objek. Method adalah mekanisme internal objek yang mencerminkan perilaku objek tersebut.

2.10.3 Kelas (Class)

Class adalah himpunan objek yang sejenis yaitu mempunyai sifat (*atribut*), perilaku umum (operasi), relasi umum dengan objek lain dan semantik umum. *Class* adalah abstraksi dari *objek* dalam dunia nyata. *Class* menetapkan spesifikasi perilaku dan *atribut* dari *objek* tersebut.

2.10.4 Kotak Hitam (*Black Box*)

Sebuah objek adalah kotak hitam. Konsep ini menjadi dasar implementasi objek. Dalam operasi OO hanya developer yang dapat memahami detail proses yang ada didalam kotak tersebut, sedangkan user tidak perlu mengetahui apa yang dilakukan yang penting mereka dapat menggunakan objek untuk memproses kebutuhan mereka. Kotak hitam berisi kode dan data.

Encapsulation, yaitu proses menyembunyikan detail implementasi sebuah objek. Untuk mengakses data objek tersebut adalah melalui interface. Untuk berkomunikasi dengan objek digunakan message.

*Messag*e adalah permintaan agar objek menerima untuk membawa metode yang ditunjukkan oleh perilaku dan mengembalikan result dari aksi tersebut kepada objek pengirim (*sender*).

2.11 UML (*Unified Modeling Languages*)

Menurut Bambang Hariyanto. Ir.,MT. (2004:17) UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi. Tujuan UML adalah :

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independent bahasa pemprograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.

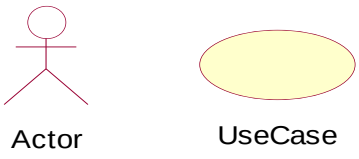
2.11.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem.

a. Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di- include akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal.
b. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common.
c. Sebuah use case juga dapat meng-extend use case lain dengan behavior- nya sendiri.
d. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain



Gambar 2.1. *Use Case Diagram*

(Munawar. 2005. *Pemodelan visual dengan UML*. Graha Ilmu. Yogyakarta.)

2.11.2 **Class Diagram**

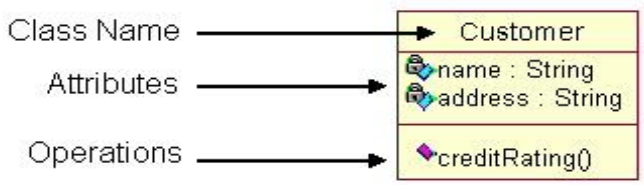
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Untuk Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. Private, tidak dapat dipanggil dari luar class yang bersangkutan
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
3. Public, dapat dipanggil oleh siapa saja



Gambar 2.2. Bagian-bagian dari *class*

2.11.3 Activity Diagram

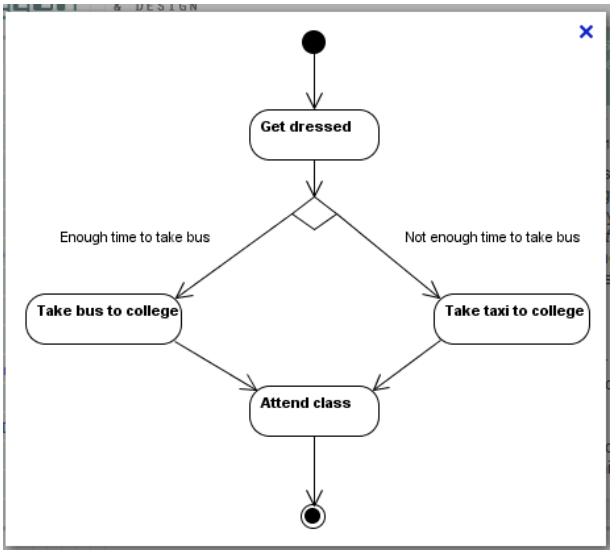
Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar 2.3. *Activity Diagram*

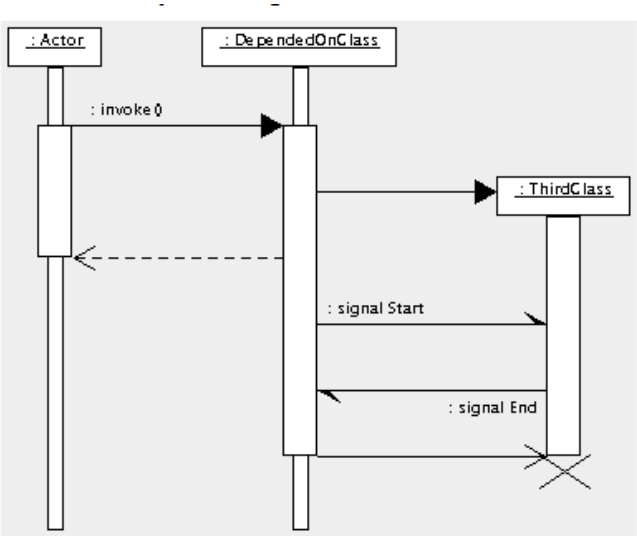
(Sumber : http://www.albertoaden.wordpress.com/ msf-fase-planning-planning-phase/ 04 juli 2012)

2.11.4 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

1. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal.
2. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class.
3. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah massage.
4. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek boundary, controller dan persistent entity.



Gambar 2.4. *Sequence Diagram*

(Sumber : http://argouml-stats.tigris.org/documentation/manual-0.32/ch19.html/ 04 juli 2012)

2.12 Android

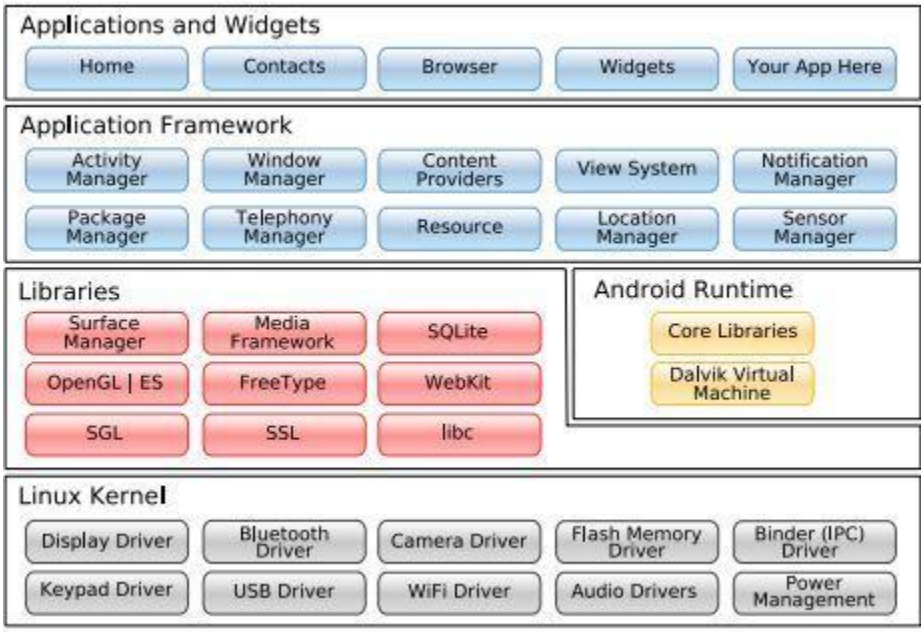
Android merupakan suatu software stack untuk *mobile device.* Di dalamnya terdapat sistem operasi, *middleware*, dan key application. Aplikasi pada platform ini dikembangkan dalam bahasa pemrograman java. Android sendiri memiliki banyak fitur diantaranya adalah:

1. Merupakan sebuah *Application Framework* sehingga programmer dapat menggunakan beberapa fungsi yang telah disediakan.
2. *Dalvik virtual machine*. Tiap aplikasi dalam Android memiliki instance virtual machine yang dapat bekerja secara efisien dalam lingkungan memoriyang terbatas.
3. *Integrated browser*. Web browser berbasis *WebKit engine* terdapat pada browser default Android atau pun dapat diintegrasikan dengan aplikasi lain.
4. *Optimized graphics*. *Library* grafis 2D yang kaya dan 3D berbasis *OpenGL* ES 1.0 yang mendukung akselerasi hardware.
5. *SQLite*. Basis data relasional yang ringan namun sangat powerful.
6. Media Support. Mendukung berbagai format audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
7. *GSM Telephony*. Mendukung fungsi komunikasi GSM.
8. *Bluetooth,EDGE,3G* dan WiFi. Mendukung komunikasi pada jaringan (tergantung hardware).

9. Kamera, GPS, kompas dan accelerometer. Mendukung berbagai fitur yang disediakan oleh hardware.
10. Kakas pengembangan yang lengkap. Termasuk *device emulator*, tools untuk debugging, profiling memori dan performa, plugin untuk *Eclipse IDE*.

2.12.1 Arsitektur Android

Android terdiri dari beberapa stack software yang terdiri dari: *Applications, Application Framework, Libraries, Android Runtime* dan *Kernel Linux*. Arsitektur lengkap platform ini dapat dilihat pada Gambar 2.1.



Gambar 2.5 Arsitektur Android

2.12.2 Aplication

Application merupakan program yang langsung berhubungan dengan user. Baik program yang merupakan bawaan dari Android sendiri maupun program yang dibuat oleh developer menggunakan bahasa pemrograman java. Contoh program bawaan dari platform Android sendiri adalah *email client*, program SMS, calendar, maps, web browser, contact dan sebagainya.

2.12.3 Aplication Framework

Lapisan ini berisi sekumpulan API yang dapat digunakan oleh programmer maupun *core application* dari Android. Lapisan ini dirancang untuk memudahkan penggunaan komponen dari Android sendiri. Aplikasi manapun dalam Android dapat berbagi fungsi sehingga aplikasi lain dapat memanfaatkannya. Aplikasi pada Android disusun atas beberapa komponen:

1. Sekumpulan Views. Digunakan untuk mengatur tampilan pada aplikasi. Contohnya adalah lists, grids, text box, button, bahkan embeddable webbrowser.
2. *Content providers.* Komponen yang mengatur agar aplikasi dapat mengakses resources dari aplikasi lain (seperti Contacts), atau berbagi data dengan aplikasi lain.
3. *Resource Manager*. Menyediakan akses ke pada resource non-code seperti localized string, grafik dan file layout.

4. *Notification Manager*. Memungkinkan agar suatu aplikasi dapat menampilkan peringatan yang dapat di kostumasi pada status bar.

Manager. Mengatur siklus aplikasi dan navigasi antar aplikasi yang sedang berjalan.

2.12.4 Libraries

Android mendukung beberapa library C/C++ yang digunakan pada berbagai komponen Android. Kemampuan ini dapat diakses oleh developer melalui *Android application framework*. Beberapa library diantaranya adalah:

1. System C library. Implementasi library C standar (libc).
2. Media Libraries. Mendukung berbagai format multimedia (termasuk MPEG4,H.264, MP3, AAC, AMR, JPG, PNG).
3. *Surface Manager*. Mengatur akses ke subsistem display.
4. LibWebCore. Engine web browser modern.
5. SGL. Engine grafis 2D.
6. 3D Library. Implementasi OpenGL ES 1.0 yang mendukung akselerasi hardware.
7. *FreeType*. Rendering untuk bitmap dan vector font.
8. *SQLite*. Basis data relasional yang kecil namun sangat ampuh.

2.12.5 Android Runtime

Tiap aplikasi pada Android memiliki proses-nya masing-masing. Tiap aplikasi tersebut memiliki instans dari *Dalvik virtual machine* (VM) *Dalvik virtual machine* dirancang agar suatu device dapat menjalankan beberapa VM secara efisien. Dalvik VM mengeksekusi file dengan format *Dalvik Executable format* (.dex) yang dirancang untuk meminimalkan memory footprint . Dalvik VM berbasis register, dan dapat menjalankan kelas-kelas yang dikompilasi dengan bahasa pemrograman java dan ditransformasikan menjadi format .dex. Dalvik VM sendiri bergantung pada Kernel Linux untuk fungsi dasarnya, seperti threading dan manajemen memori secara low-level.

2.12.6 Anatomi Aplikasi Android

Terdapat empat *building blocks* pada aplikasi *Android* yaitu:

1. *Activity*
2. *Intent Receiver*
3. *Service*
4. *Content Provider*

Tidak semua aplikasi membutuhkan keempat blok ini, tetapi suatu aplikasi dibuat menggunakan kombinasi beberapa blok ini. Setelah memutuskan blok komponen yang akan digunakan, blok tersebut didaftarkan pada suatu file yang disebut dengan *AndroidManifest.xml*. File XML ini digunakan untuk menyatakan

komponen apa saja yang dibutuhkan oleh sebuah aplikasi dan kemampuan serta kebutuhan aplikasi tersebut.

2.12.7 Perkembangan Aplikasi Android

Android mengalami perkembangan yang baik berikut adalah tahapan perkembangan android :

1. Android Beta

   Pertama kali dirilis pada 5 November 2007, kemudian pada 12 November 2007 Software Development Kit (SDK) dirilis oleh Google.

2. Android 1.0 (Astro)

   Pertama kali dirilis pada 23 Spetember 2008. Sebenarnya Android versi pertama ini akan dinamai dengan nama "Astro" tapi karena alasan hak cipta dan trademark nama"Astro" tidak jadi disematkan pada versi pertama dari OS Android ini. HTC Dream adalah ponsel pertama yang menggunakan OS ini. Berikut penampakan HTC Dream.

   Pada android versi pertama ini sudah mulai diperkenalkan Android Market, tempat untuk mendownload berbagai macam aplikasi dan update untuk Android. Kemudian terdapat fitur syncronisasi dengan berbagai layanan Google lainnya seperti Google Contact, *Calender*, Maps, Talk, dan lainnya.

3. Android 1.1 (*Bender*)

   Pertama kali dirilis pada 9 Februari 2009. Versi Android kedua ini juga mengalami masalah penamaan yang sama dengan versi pertamanya. Pada awalnya Android ini akan diberi nama "Bender" akan tetapi karena alasan melanggar trademark, nama "Bender" tidak jadi disematkan pada versi Android ini. Awalnya versi OS Android ini dirilis untuk perangkat T-Mobile G1 saja. Versi ini merupakan update untuk memperbaiki beberapa bugs, mengganti API dan menambahkan beberapa fitur. Berikut penampakan home screen OS Android v.1.1.

4. Android 1.5 Cupcake

   Pertama kali dirilis pada 30 April 2009. Nah, mulai versi Android ini penamaan menggunakan nama makan pencuci mulut (dessert) mulai digunakan, karena ini merupakan versi yang ketiga maka penamaan diawali dengan huruf "C" dan jadilah "Cupcake" menjadi nama resmi dari versi OS Android ketiga ini. OS ini berbasiskan pada kernel Linux 2.6.27 dan menambahkan beberapa update serta UI baru dari versi Android sebelumnya. Mulai terdapat "widget" yang dapat dibesar kecilkan. Kemudian ditambah kemampuan untuk meng-upload video dan gambar ke Youtube dan Picasa.

5. Android 1.6 Donut

   Dirilis pertama kali pada 15 September 2009. Terdapat peningkatan pada fitur pencarian dan UI yang lebih user friendly. Pada versi ini juga sudah mendukung teknologi CDMA/EVDO, 802.1x, VPNs.

Kemudian support layar dengan resolusi WVGA. Berikut penampakan Android v1.6 Donut.

6. Android 2.0/2.1 Enclair

   Dirilis pertama kali pada 9 Desember 2009. Terjadi penambahan fitur untuk 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. Beberapa versi updatenya antara Android v.2.0 kemudian v2.0.2 dan terakhir v.2.1.

7. Android 2.2 Frozen Yoghurt (Froyo)

   Dirilis pertamakali pada 20 Mei 2010 pada smartphone Google Nexus One. Pada versi ini sudah support terhadap Adobe Flash Player 10.1. Peningkatan pada kecepatan membuka dan menutup aplikasi, serta penggunaan SD Card sebagai tempat penyimpanan aplikasi. Ketika Android Froyo hadir mulai muncul banyak diskusi yang membahas mengenai persaingan antara Android dengan iOS yang akan semakin ketat di masa yang akan datang. Beberapa versi update yang dirilis antara lain Android v.2.2.1 hingga v.2.2.3.

8. Android 2.3 Gingerbread

   Pertama kali diperkenalkan pada 6 Desember 2010. Terjadi banyak peningkatan pada versi Android yang satu ini dibandingkan dengan versi sebelumnya. Dirancang untuk memaksimalakan kemampuan aplikasi dan game. Serta mulai digunakannya Near Field

Communication (NFC). Perbaikan terhadap dukungan layar resolusi WXGA dan diatasnya. Beberapa versi update yang dirilis antara lain v.2.3.3 hingga v.2.3.7. Sampai saat ini Android Gingerbread merupakan versi Android yang memiliki pengguna terbanyak dibandingkan dengan seri Android lainnya, yaitu mencapai 65% dari seluruh versi Android yang dirilis.

9. Android 3.0/3.1 Honeycomb

   Pertama kali diperkenalkan pada 22 Februari 2011 dan Motorola Xoom adalah yang pertama kali menggunakannya. Android versi ini merupakan OS yang didesain khusus untuk pengoptimalan pengunaan pada tablet PC.

10. Android 4.0 Ice Cream Sandwich

    Sampai tulisan ini ditulis ICS merupakan versi Android yang paling anyar. Pertama kali dirilis pada 19 Oktober 2011. Smartphone yang pertama kali mengunakan OS Android ini adalah Samsung Galaxy Nexus. Gingerbread, dapat di-update ke Android Ice Cream Sandwich. Namun sayangnnya sampai saat ini kebanyak smartphone yang menggunakan Android ICS merupakan smartphone kelas high-end yang dijual dengan harga cukup mahal. Mungkin karena alasan inilah distribusi versi Android satu ini tidak lebih dari 8% sampai pertangahn tahun 2012 ini.Berikut diagram distribusi OS Android berdasarkan versi-nya.

2.13 Perangkat Lunak Yang Digunakan

Untuk memulai pemrograman mobile, memerlukan suatu tool pengembangan khusus. Salah satu tool yang paling populer untuk pemrograman berbasis android adalah *eclips* Indigo IDE 3.7.1.

2.13.1 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:

- *Multi-platform:* Target sistem operasi Eclipse adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX* dan *Mac OS X*.
- *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.
- 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.

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:*

- *Core platform*
- *OSGi*
- *SWT (Standard Widget Toolkit)*
- *JFace*
- *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. Salah satu situs yang menawarkan plug-in secara gratis seperti *Eclipse downloads by project.*

2.13.2 Adobe Photoshop

Adobe Photoshop, atau biasa disebut Photoshop, adalah perangkat lunak editor citra buatan Adobe Systems yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (market leader) untuk perangkat lunak pengolah gambar/foto, dan, bersama Adobe Acrobat, dianggap sebagai produk terbaik yang pernah diproduksi oleh Adobe Systems. Versi kedelapan aplikasi ini disebut dengan nama Photoshop CS (Creative Suite), versi sembilan disebut Adobe Photoshop CS2, versi sepuluh disebut Adobe Photoshop CS3 , versi kesebelas adalah Adobe Photoshop CS4 , versi keduabelas adalah Adobe Photoshop CS5 , dan versi yang terakhir (ketigabelas) adalah Adobe Photoshop CS6.

Photoshop tersedia untuk Microsoft Windows, Mac OS X, dan Mac OS; versi 9 ke atas juga dapat digunakan oleh sistem operasi lain seperti Linux dengan bantuan perangkat lunak tertentu seperti CrossOver.

2.13.3 Microsoft Office Picture Manager

Microsoft Office Picture Manager adalah sebuah program perangkat lunak yang disertakan dengan Ms office versi Suite 2003 sampai 2010. Hal ini tidak lagi disertakan dengan Office 2013. Picture Manager adalah editing gambar dasar dan program manajemen gambar.

Program ini pergi melalui revisi beberapa nama sementara dalam versi beta, terutama mulai keluar sebagai Microsoft Office Picture Library 2003, kemudian kehilangan penunjukan 2003 (lebih dicadangkan untuk aplikasi Office yang sebenarnya) dalam versi beta 2, dan akhirnya menetap pada nama saat ini.