BAB II

LANDASAN TEORI

2.1. Pengertian Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Menurut Widodo, Priyono (Kamus Istilah Internet dan Komputer 2002:1). Secara istilah aplikasi merupakan program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju.

2.2. Jenis – Jenis Aplikasi

Berikut penjelasan mengenai jenis – jenis dari Aplikasi :

1. *Software* Berbayar

Software berbayar ialah aplikasi yang didistribusikan untuk tujuan komersial, dimana setiap *user* yang ingin menggunakan software tersebut harus membeli atau membayar kepada pihak yang mendistribusikannya. *user* yang menggunakan software berbayar umumnya tidak dijinkan untuk menyebarluaskan software tersebut tanpa ada ijin secara bebas dari penerbitnya.

2. *Freeware*

Freeware (perangkat lunak gratis) ialah aplikasi komputer berhak cipta yang gratis digunakan tanpa batasan waktu. Para pengembang

freeware seringkali membuat perangkat gratis freeware untuk disumbangkan kepada komunitas.

3. Shareware

Shareware adalah aplikasi yang didapatkan secara gratis, tapi mempunyai batas waktu. Shareware merupakan program berbatas waktu yang didistribusikan baik sebagai demonstrasi ataupun versi evaluasi dengan fitur terbatas atau dengan menggunakan batas waktu yang telah ditetapkan. Dengan demikian, pengguna diberi kesempatan untuk menguji program terlebih dulu sebelum membeli versi lengkap dari software tersebut.

2.3. Pengertian Mobile Learning

Mobile learning as taking place when the learner is not at a fixed, predetermined location, or when the learner takes advantage of the learning opportunities offered by mobile technologies. Menurut Agnes Kukulska dan John Traxler (2003:6). dari definisi tersebut dapat disimpulkan bahwa, *m-learning* adalah metode pembelajaran yang memanfaatkan teknologi informasi dan komunikasi. Pada konsep pembelajaran, *m-learning* membawa manfaat ketersediaan materi ajar yang dapat di akses setiap saat dan visualisasi materi yang menarik. *Mobile learning* juga dapat didefinisikan sebagai suatu fasilitas atau layanan yang dapat memberikan informasi elektronik secara umum kepada pembelajar dan *content* yang edukasional yang membantu pencapaian pengetahuan tanpa mempermasalahkan lokasi dan waktu. Sistem *m-learning* ini memanfaatkan

mobilitas dari perangkat *handheld*/*mobile*, seperti *handphone* dan PDA, untuk memberikan suatu fungsi pembelajaran yang dapat dilakukan dimanapun dan kapanpun.

2.4. IPS (Ilmu Pengetahuan Sosial)

Numan Sumantri (2001: 93) menyatakan bahwa "Pendidikan IPS adalah penyederhanaan atau adaptasi dari disiplin ilmu-ilmu sosial dan humaniora, serta kegiatan dasar manusia yang diorganisasikan dan dikaji secara ilmiah dan pedagogis atau psikologis untuk tujuan pendidikan."

Pendapat tersebut sejalan dengan yang diutarakan oleh National Council for Social Studies(NCSS), yang mendefinisikan IPS adalah sebagai berikut: "Social studies are the integrated study of the social sciences and humanities to promote civic competence. Within the school program, social studies provides coordinated, systematic study drawing upon such disciplines as anthropology, archaeology, economics, geography, histori, law, philosophy, political science, psychology, religion, and sociology, as well as appropriate content from the humanities, mathematics, and the natural sciences" (Savage and Armstrong, 1996).

Pendapat serupa dikemukakan oleh Trianto (2010: 171) menyatakan bahwa:

"Ilmu Pengetahuan Sosial (IPS) merupakan integrasi dari berbagai cabang ilmu-ilmu sosial. Ilmu-ilmu sosial yang dimaksud seperti sosiologi, sejarah,

geografi, ekonomi, politik, hukum, dan budaya. Ilmu Pengetahuan Sosial dirumuskan atas dasar realitas dan fenomena sosial masyarakat yang diwujudkan dalam satu pendekatan interdispliner dari aspek dan cabang-cabang ilmu social tersebut".

2.4.1. Sejarah IPS

IPS merupakan terjemahan dari studi sosial (social studies) yang mulai diterapkan dalam dunia pendidikan dasar dan menengah di Amerika Serikat sejak tahun 1915 setelah perang dunia pertama. Para ahli pendidikan di Amerika Serikat pada waktu itu berkesimpulan bahwa pengajaran Ilmu-ilmu sosial yang diajarkan secara sendiri-sendiri dalam bentuk disiplin ilmu, seperti: Sejarah, geografi, ekonomi, dan lain-lain tidak akan mampu membekali para subyek didik untuk dapat mengenal dan mengerti masalah sosial yang ada disekitarnya. Dengan demikian diintroduksikannya social studies yang diharapkan dapat mengatasi kekurangan.

Kelahiran Bidang Studi IPS dalam Kurikulum sekolah di Indonesia, banyak-banyak di ilhami oleh pengajaran social studies di Amerika Serikat. Bahkan istilah Ilmu pengetahuan sosial (IPS), adalah terjemahan dari apa yang dinamakan Social studies dalam dunia pendidikan dasar dan menengah di Amerika Serikat (N. Daljuni 1981). Pengajaran IPS di Indonesia pertama kali diperkenalkan oleh pakar IPS pada tahun 1969 yaitu oleh Ibu Prof Dr. Soepartina Pakasi pada SD PPSP IKIP Malang.

Pada tahun 1971 IPS dimasukkan dalam buku induk Depdikbud. Pada tahun 1972 sudah ramai diperbincangkan dalam rencana pembaharuan Kurikulum sekolah di Indonesia. Bidang studi IPS resmi di cantumkan dalam kurikulum pada tahun 1974. Pada tahun 1975 nama bidang studi IPS sudah tercantum dalam kurikulim SD, SMP, SMU. Pelaksanaannya dilaksanakan secara bertahap dimulai pada tahun 1976. Jadi Ilmu Pengetahuan Sosial (IPS) di Indonesia kelahirannya bersamaan dengan lahirnya kurikulum tahun 1975.

Perkembangan ilmu pengetahuan dan teknologi dewasa ini, dimana dunia pengajaran sekolah pada umumnya selalu tertinggal, maka IPS diperlukan sebagai wadah pengetahuan yang mengharmoniskan laju perkembangan ilmu dan kehidupan dalam dunia pengajaran sekolah. Sebab IPS mampu melakukan lompatan-lompatan ilmu secara konsepsional untuk kepentingan praktis kehidupan baru yang sesuai dengan keadaan dan zaman. Maka melihat jenis dan susunan konsep/topik dalam IPS sungguh sangat banyak bervariasi dari berbagai ilmu sosial serta dari tuntutan-tuntutan persoalan kehidupan praktis.

2.5. Pengertian Android

Menurut Nasruddin Safaat H (Pemrograman aplikasi mobile smartphone dan tablet PC berbasis android 2012:1) android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada

sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android bersifat opensource yang berdampak pada meningkatnya jumlah pengguna maupun pengembang aplikasi secara continue dan signifikan.

2.5.1. Sejarah Sistem Operasi Android

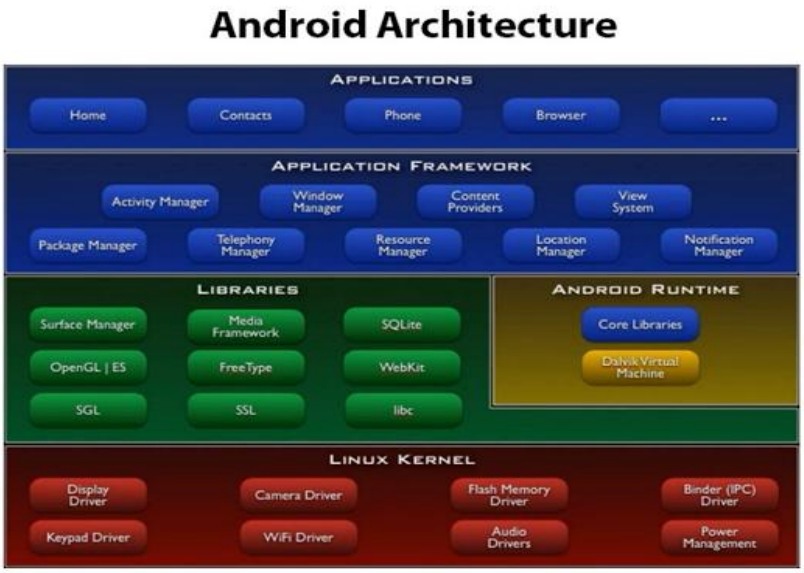
Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak 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.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode–kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar–

benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

2.5.2. Arsitektur dan Komponen Android

Android akan bekerja dengan serangkaian aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.



Gambar 2.5.2 Arsitektur Android

Sumber: www.dhunju.com/2012/01/android-architecture.html

2.5.3. The Dalvik Virtual Machine (DVM)

Salah satu elemen kunci dari Android adalah *Dalvik Virtual Machine* (DVM). Android berjalan di dalam *Dalvik Virtual Machine* (DVM) bukan di *Java Virtual Machine* (JVM). *Java Virtual Machine* memiliki banyak persamaan dengan DVM seperti Java ME (*Java Mobile Edition*), tetapi Android menggunakan *Virtual Machine* sendiri yang dikustomisasi dan dirancang untuk memastikan bahwa beberapa *feature* berjalan lebih efisien pada perangkat *mobile*.

Semua *hardware* yang berbasis Android dijalankan dengan menggunakan *Virtual Machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *Executeble File* diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan *tools* yang disediakan dalam Android.

2.5.4. Android SDK (Software Development Kit)

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Saat ini disediakan Androiod SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android dan menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi netral,

Android memberikan kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone/Smartphone*.

2.5.5. ADT (Android Development Tools)

Android Development Tools (ADT) adalah plugin yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi Android. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, selain itu kita dapat melakukan *running* aplikasi menggunakan Android SDK melalui Eclipse.

Dengan ADT kita dapat melakukan pembuatan *package* Android (.apk) yang digunakan untuk distribusi aplikasi Android yang dirancang. Berikut adalah versi ADT untuk Eclipse yang telah dirilis:

1. ADT 12.0.0 (Juli 2011)
2. ADT 11.0.0 (Juni 2011)
3. ADT 10.0.1 (Maret 2011)
4. ADT 10.0.0 (Februari 2011)
5. ADT 9.0.0 (Januari 2011)
6. ADT 8.0.1 (Desember 2010)
7. ADT 8.0.0 (Desember 2010)
8. ADT 0.9.9 (September 2010)

9. ADT 0.9.8 (September 2010)
10. ADT 0.9.7 (Mei 2010)
11. ADT 0.9.6 (Maret 2010)
12. ADT 0.9.5 (Desember 2009)
13. ADT 0.9.4 (Oktober 2009)

Semakin tinggi *platform* Android yang kita gunakan, dianjurkan menggunakan ADT yang lebih terbaru, karena biasanya muncul *platform* yang baru diikuti dengan munculnya versi ADT yang terbaru. Untuk melakukan instalasi ADT di Eclipse dapat dilakukan dengan *online* maupun *offline*.

2.5.6. Versi Android

Sejak Android diakuisisi pada tahun 2005 oleh google, telah lahir beberapa versi *upgrade* dari sistem operasi Android tersebut. Berikut adalah versi Android yang telah diperkenalkan:

a. Versi 1.1 Bender

Versi ini dirilis pada tanggal 9 februari 2009. *User interface* yang ada telah mendukung aplikasi standart, alarm, olah pesan *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

b. Versi 1.5 Cupcake

Cupcake merupakan pengembangan dari versi 1.1, *interface* Android lebih dipercantik dan penambahan efek-efek animasi.

Inovasi yang lebih dilakukan pada versi ini lebih ke arah multimedia. Cupcake dirilis pada tanggal 30 April 2009 yang mampu merekam dan memutar video.

c. Versi 1.6 Donut

Pada Android Versi 1.6 ini, banyak terobosan baru di sistem jaringan dan konektivitas. Dirilis pada 15 September 2009 sudah ada pengintegrasian antara kamera, kamera video, dan galeri foto. Selain itu, peningkatan fitur voice search, voice dial, dan peningkatan teknologi jaringan CDMG (EVDO), wifi, gesture dan mesin text to speech.

d. Versi 2.0/2.1 Eclair

Di Eclair ini, terdapat perubahan total struktur dan tampilan *user interface* dan telah mendukung format HTML5.

e. Versi 2.2 Froyo (Frozen Yoghurt)

Terdapat 20 fitur baru pada Froyo, antara lain peningkatan kecepatan, fitur Wi-Fi hotspot tethering dan dukungan terhadap Adobe Flash.

f. Versi 2.3 Gingerbread

Perubahan utama di versi 2.3 ini termasuk update UI, peningkatan fitur soft keyboard dan copy/paste, power management, dan support Near Field Communication.

g. Versi 3.0/3.1/3.2 Honeycomb

Honeycomb merupakan versi Android yang ditujukan untuk gadget/device dengan layar besar seperti Tablet PC. Fitur baru Honeycomb yakni adalah dukungan terhadap posesor multicore dan grafis dengan hardware acceleration.

h. Versi 4.0 Ice Cream Sandwich

Android Ice Cream Sandwich diumumkan pada 10 Mei 2011 di ajang Google I/O Developer Conference (San Fransisco) dan resmi dirilis pada tanggal 19 Oktober 2011 di Hongkong. Android ini dapat digunakan baik di tablet maupun *smartphone*. Fitur utama versi Android ini adalah Face Unlock, Android Beam (NFC), perubahan mayor *User Interface*, dan ukuran layar standar (*native screen*) ke resolusi 720p (high definition).

i. Versi 4.1/4.2 Jelly Bean

Android Jelly Bean lebih memfokuskan fitur peningkatan *User Interface* yang lebih lancar dan responsif. Pada versi ini juga telah terdapat fitur Google Now yang memberikan saran dan rekomendasi berdasarkan data-data yang tersimpan (kontak,kalender,lokasi dll) di handphone.

j. Versi 4.4 KitKat

Android 4.4 KitKat yang merupakan pengganti Android 5.0 Key Lime Pie ini diprediksikan akan kompatibel untuk dapat dipakai pada smartphone Android mulai dari kelas *high-end* hingga *low-end*.

Hal ini mengemuka karena rumor yang beredar sebelumnya bahwa Key Lime Pie akan dapat dipakai pada smartphone / tablet dengan RAM berkapasitas mulai 512MB (low-end).

2.6. Perancangan Sistem

Secara umum perancangan sistem adalah suatu tahap dimana didalamnya terdapat identifikasi komponen-komponen sistem informasi yang akan dirancang secara rinci, yang bertujuan untuk memberikan gambaran kepada pengguna mengenai sistem yang baru.

2.6.1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem.

1. *Use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu.
2. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Use case diagram dapat digunakan untuk :

1. Menyusun *requirement* sebuah sistem,
2. Mengkomunikasikan rancangan dengan klien, dan
3. Merancang *test case* untuk semua *feature* yang ada pada sistem.

2.6.2. Activity Diagram

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

2.6.3. 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 antar 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.

2.6.4. 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 diantaranya :

1. Atribut/properti suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).
2. 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 adalah Nama (dan stereotype), Atribut, dan Metoda. 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.

2.6.5. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) di antaranya. Komponen piranti lunak adalah modul berisi code, baik berisi source code maupun binary code, library maupun executable yang muncul pada compile time, link time, maupun run time. Umumnya komponen terbentuk dari class dan package, tapi dapat juga dari komponen-komponen yang lebih kecil.

2.6.6. Deployment Diagram

Deployment Diagram menggambarkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.