BAB II

LANDASAN TEORI

2.1 *Game*

Game atau permainan merupakan suatu aktivitas dengan tujuan untuk mengisi waktu luang dan memiliki beberapa aturan tertentu. Permainan biasanya dilakukan secara sendiri ataupun berkelompok. *Game* atau permainan juga merupakan simulasi dari bentuk nyata dalam kehidupan manusia [1]. Perkembangan *game* saat ini sangat pesat terdapat banyak sekali jenis *game* yang beredar, salah satu jenis yang populer saat ini adalah *casual rhythm game*. Selain karena menyenangkan aktifitas bermain permainan banyak dilakukan karena dukungan banyak *platform* untuk memainkannya, salah satu *platform* yang sangat diminati adalah android.

2.1.1 *Game Rhythm*

Game rhythm termasuk jenis *game casual* yang merupakan permainan musik yang berorientasi pada interaksi pemain dengan skor musik atau lagu individu. *Game* ini dapat dikelompokkan dengan permainan *puzzle* karena penggunaan umum tentang teka – teki berirama (*beat*) yang dihasilkan oleh permainan ini [4] *Game* ini biasanya berfokus pada kinerja simulasi alat musik, dan membutuhkan pemain untuk menekan tombol dalam urutan yang ditentukan di layar pada waktu yang tepat, layar menunjukkan tombol mana yang akan

ditekan oleh pemain dan kemudian *game* akan memberikan kepada pemain poin/skor sesuai dengan tingkat akurasi dan sinkronisasi dengan *beat*.

Salah satu *game rhythm* yang terkenal adalah VOS (*Virtual Orchestra Studio*). VOS sendiri adalah *game* seperti kita bermain Piano di Keyboard kita dengan 7 jari. Saat memainkan VOS kita dapat berlatih tangan , jari dan pikiran secara bersamaan. *Game* ini dibuat oleh Hanseul Soft, Korea Utara. Grafik *game* ini pun sangat simpel, dibuat dengan *sprites-sprites* 2D dan 3D tuts yang ada di keyboard.

Pada *game* ini akan memilih lagu dan karakter yang disukai.Karakter yg dipilih akan mempengaruhi mata ketika bermain sebagai karakter masing-masing akan menggunakan warna latar belakang yang berbeda. Karakter yang tersedia antara lain Rock(ungu), Bit-robot (cyan), Flanka-sebuah raksasa lender(hijau lender), astri(abu-abu), PLO (biru-abu2), Franken (merah muda) dan Rucy (biru).*game* ini terdiri dari 7 *beat* berbeda yang akan jatuh ke bawah dan mencapai garis biru, dan pemain harus menekan tombol yang sesuai pada saat yang tepat ketika *beat* masih dalam garis biru. Setiap kali pemain menekan tombol dimana *beat* jatuh ke garis biru maka pemain akan mendapatkan skor. Skor tersebut akan bervariasi tergantung kecepatan jari menekan tombol, jika penekanan tombol 2 kali maka akan terjadi kesalahan dan jika pemain menekan tombol terlalu cepat sebelum *beat* muncul maka pemain tidak mendapatkan skor, dan apabila pemain menekan tombol setelah *beat* terlewati maka akan dinyatakan kehilangan (*miss*). Tingkat skor apabila pemain dengan tepat menekan tombol saat *beat* muncul adalah baik.

Permainan ini membutuhkan musik sebagai pengantar pada saat permainan berlangsung. Musik dalam permainan ini menggunakan format kustom VOS yang pada dasarnya sebuah file *midi* dengan pengaturan *beat* yang kemudian di konversi ke format VOS. *Game* ini juga mendukung Be-Musik Script file (.BMS) yang dikembangkan Urao Yane.

Be-musik script (BMS) adalah format file untuk permainan *rhythm*. BMS ini menjelaskan bagimana *beat* tersebut diatur dan bagaimana permainan merespon ketika pemain mengambil tindakan tertentu. Ini adalah file teks biasa dengan garis – garis 'direktif' dimulai dengan #karakter. File BMS terdistribusi normal dengan gambar yang sesuai dengan file suara. Format file BMS pada awalnya digunakan untuk simulator *beatmania*, tetapi karena *game* ini telah diadopsi dan didukung oleh banyak *game action* maka irama pun termasuk ke dalam stepmania. File BMS juga telah diproduksi untuk musik asli, bukan lagu – lagu *game* komersial.

2.1.2 StepMania [15]

Stepmania merupakan *rhythm* dan *engine game* yang bersifat *cross-platform*, dan termasuk kedalam perangkat lunak *open source*. Pada awalnya stepmania dikembangkan sebagai simulator permainan konami yang merupakan seri dari permainan *Dance Dance Revolution*, dan kemudian berkembang menjadi sebuah *rhythm game engine* yang mampu mendukung berbagai jenis permainan berbasis *rhythm*.

Stepmania memiliki format berkas sendiri yaitu ".sm", dimana format ini dibuat untuk menjadi format berkas yang mendukung semua jenis *game* yang

dapat dimainkan oleh stepmania (*dance, pump, beat, guitar,* dll). Sintaks yang terdapat pada berkas ".sm" tidak berbeda jauh dengan ".dwi" dan ".ksf" hanya terdapat beberapa tags yang berbeda.

Pada berkas ini tersimpan informasi mengenai lagu, *beat* dan latar belakang atau *cover* dari lagu. Berikut beberapa *tag*(label) yang terdapat pada berkas stepmania:

Tabel 2-1 Label dalam berkas StepMania

Label	Keterangan
#TITLE	Berisi informasi mengenai judul lagu
#SUBTITLE	Isi pada label ini akan ditampilkan dibawah judul utama dari lagu pada layar *Select Music* di aplikasi stepmania
#ARTIST	Nama artis yang menyanyikan lagu
#CREDIT	Nama dari pembuat *beat*
#BANNER	Berisi nama berkas dari gambar *banner*
#BACKGROUND	Berisi nama dari berkas gambar latar
#MUSIC	Berisi nama dari berkas music
#OFFSET	Jumlah waktu dalam detik sebelum atau sesudah musik dimulai sehingga *beat* 0 terjadi.
#BPMS	Memiliki format nilai "*beat*=bpm", yang menunjukkan bahwa pada "*beat*", kecepatan panah akan berubah sesuai "bpm"
#NOTES	Berisi deskripsi mengenai *beat*/note

Pada #NOTES berisi sub-tag tentang informasi *beat*, antara lain:

a. NotesType, harus berisi salah satu jenis dari format yang didukung oleh stepmania
b. DifficultyClass, berisi nama dari tingkat kesulitan yang akan digunakan pada *beat*
c. DifficultyMeter, berisi peringkat dari tingkat kesulitan
d. NoteData, berisi 4 karakter yang menjelaskan *beat* yang akan keluar

1. Karakter 0, menandakan tidak ada *beat*
2. Karakter 1, menandakan *beat* biasa (tap note)
3. Karakter 2, menandakan mulainya “hold note”
4. Karakter 3, menandakan berakhirnya “hold note”

2.2 Android

Android adalah sistem operasi yang digunakan di smartphone dan juga tablet PC. Android juga merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan *key-application* yang di release oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API (*Application Programming* Interface) diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java.

2.2.1 Sejarah Android [12]

Android pertama kali dikembangkan oleh perusahaan bernama Android Inc., dan pada tahun 2005 di akuisisi oleh raksasa Internet Google. Android dibuat dengan basis kernel Linux yang telah dimodifikasi, dan untuk setiap release-nya diberi kode nama berdasarkan nama hidangan makanan. Keunggulan utama Android adalah gratis dan *open source*. Keuntungan open source, banyak pengembang software yang bisa melihat dan memanfaatkan kode itu serta bisa membuat aplikasi baru di dalamnya. Berbagai aplikasi android diwadahi dalam

sebuah portal, yaitu Android Market, sehingga pengguna tinggal meng-*install* aplikasi pilihannya.

a. Produk Awal

September 2007 Google mengajukan hak paten aplikasi telepon seluler.Android yang digunakan pada perangkat bergerak (mobile) merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis dilakukan berbagai pembaharuan berupa perbaikan bug dan penambahan fitur baru.

b. Android Versi 1.1

Dirilis oleh Google pada bulan Maret 2009. Android versi ini di lengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail dan pemberitahuan *email*.

c. *Android Cupcake (*1.5)

Hanya selang 1 bulan Google merilis Android 1.5 **(***Cupcake***)** dengan SDK (*Software Development Kit*). Terdapat beberapa pembaharuan termasuk juga penambahan beberapa fitur dalam seluler yakni kemampuan merekam dan menonton video dengan modus kamera, menggugah 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 disesuai dengan system.

d. *Android Donut (*1.6)

Versi ini dirilis pada September dengan menampilkan proses pencarian yang lebih baik, penggunaan baterai indikator dan kontrol *applet* VPN (*Virtual Private Network*) atau jaringan pribadi. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder, dan galeri

yang diintegrasikan : CDMA/EVDO, 802.1x, VPN, Gestures dan Text to speech engine, kemampuan dial kontak, teknologi text to change speech (tidak tersedia dalam ponsel), pengadaan resolusi VWGA.

e. Android Eclair (2.0 / 2.1)

Versi ini dirilis 3 desember 2009 yang mulai dipakai oleh banyak smartphone, fitur utama Eclair yaitu perubahan total struktur dan tampilan user interface dan merupakan versi Android yang pertama kali mendukung format HTML5. Perubahan yang lain yaitu : pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru , daftar kontak yang baru, dukungan *flash* untuk kamera 3.2 MP, *digital zoom*, dan *Bluetooth* 2.1.

f. Android Froyo / *Frozen Yoghurt* (2.2)

Versi ini dirilis 20 mei 2010, android 2.2 dirilis dengan 20 fitur baru, antara lain peningkatan kecepatan kinerja, fitur Wi-Fi hotspot tethering dan dukungan terhadap Adobe Flash 10.1, aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 Javascript *engine* yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD card, kemampuan WiFi *Hotspot portable*, dan kemampuan *auto update* dalam aplikasi Android *Market*.

g. *Android Gingerbread* (2.3)

Versi ini dirilis 6 desember 2010. Perubahan utama di versi 2.3 ini termasuk update UI, peningkatan fitur *soft keyboard* & *copy/paste, power management*, dan support *Near Field Communication*, peningkatan kemampuan permainan (*gaming*), layanan antarmuka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb, equalization, headphone virtualization* dan *bass boost*), dan dukungan jumlah kamera yang lebih dari satu.

h. *Android Honeycomb* (3.0, 3.1 dan 3.2)

Merupakan versi Android yang ditujukan untuk gadget / device dengan layar besar seperti Tablet PC; Fitur baru Honeycomb yaitu dukungan terhadap prosessor multicore dan grafis dengan hardware acceleration.

i. *Android Ice Cream Sandwich* (4.0)

Android 4.0 Ice Cream Sandwich diumumkan pada 10 Mei 2011 di ajang *Google* 26 *I/O Developer Conference* (San Francisco) dan resmi dirilis pada tanggal 19 Oktober 2011 di Hongkong. Android versi 4.0 ini dapat digunakan di smartphone ataupun tablet. Fitur utama yang ditambahkan di Android 4.0 ialah *Face Unlock*, *Android Beam*, perubahan major *User Interface*, dan ukuran layar standar (*native screen*) beresolusi 720p *(high definition)*, perangkat tambahan fotografi, mencari email secara offline dan berbagai informasi dengan menggunakan NFC.

2.2.2 Fitur-fitur di Android

Berikut ini adalah fitu-fitur yang ada pada Android:

a. **Framework Aplikasi** yang mendukung penggantian komponen dan *reusable*
b. **Mesin virtual Dalvik** dioptimalkan untuk perangkat *mobile*
c. **Integrated browser** berdasarkan *engine open source WebKit*
d. **Grafis** yang dioptimalkan dan didukung oleh *library* grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware)
e. **SQLite** untuk penyimpanan data
f. **Media Support** yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)

g. **GSM Telephony** (tergantung hardware)

h. **Bluetooth, EDGE, 3G, dan WiFi** (tergantung hardware)

i. **Kamera, GPS, kompas, dan accelerometer** (tergantung hardware)

j. **Lingkungan Development** yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk Eclipse IDE.

k. **Dukungna Perangkat Tambahan** : Android dapat memanfaatkan kamera, layar sentuh, *accelerometers*, *magnetometers*, GPS, akselerasi 2D (dengan perangkat orientasi, scalling,konversi format piksel), dan akselerasi grafis 3D.

l. ***Multi-Touch*** : kemampuan layaknya handset modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.

m. **Market** : handphone memiliki tempat penjualan aplikasi, market pada android merupakan katalog aplikasi yang dapat di *download* dan di-*install* melalui internet.

2.2.3 Open Handset Alliance

Dibentuklah aliansi bisnis yang bernama *Open Handset Alliance* (5 November 2007), yang dikepalai/dipimpin oleh Google, terdiri dari 78 perusahaan, dimana didalamnya terlibat banyak perusahaan-perusahaan besar. Tujuan dibentuknya OHA yaitu untuk bersaing dengan platform perangkat ponsel lainnya, seperti yang dikembangkan oleh Apple, Microsoft, Nokia, Palm, Research In Motion, Symbian, and bada.

Trend

Saat ini trend penggunaan ponsel Android semakin meningkat, terbukti pada tahun 2010 penjualan ponsel dengan OS Android di US telah mencapai peringkat pertama (33%), kemudian disusul oleh ponsel dengan OS BlackBerry (28%), dan yang ketiga oleh ponsel dengan iOS (22%).

Developer (*Pengembang*)

Bahasa yang digunakan untuk membuat aplikasi Android adalah Java. Oleh karena itu jangan heran jika Android memiliki komunitas pengembang aplikasi yang sangat luas dan besar. Tentunya dengan besarnya komunitas ini fungsionalitas dari perangkat yang menggunakan OS Android juga meningkat. Saat ini telah terdapat **70,000** aplikasi Android yang tersedia, dan menjadikan Android sebagai *environment* pengembangan aplikasi *mobile* **terpopuler** kedua saat ini.

Lisensi

Sebagian besar source code yang dirilis oleh Android berada dalam lisensi Apache, serta lisensi free software dan open source.

2.2.4 Arsitektur Android

Sistem operasi Android memiliki 4 lapisan (layer) yang merupakan komponen system Android. Gambar berikut merupakan lapisan arsitektur system operasi android.

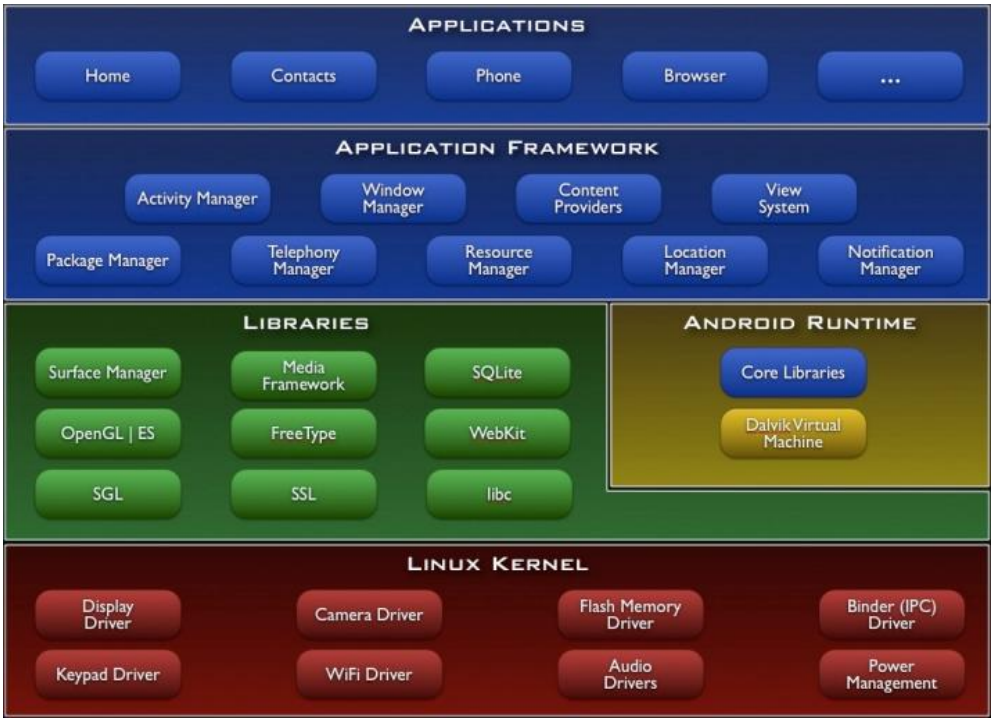


Diagram 2-1 Diagram arsitektur sistem operasi android

Google mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari system operasi. Berikut ini susunan dari lapisan – lapisan tersebut dilihat dari lapisan dasar hingga lapisan teratas.

Linux Kernel

Tumpukan paling bawah pada arsitektur Android ini adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun system Android, yang mencakup memory management, security setting, power management dan beberapa driver hardware. Kernel berperan sebagai abstraction layer antara hardware dan keseluruhan software. Sebagai contoh : HTC GI dilengkapi dengan kamera. Kernel Android terdapat driver kamera yang memungkinkan pengguna mengirimkan perintah kepada hardware kamera.

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, WiFi, *Flash Memory*, audio dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

Libraries

Bertempat di level yang sama dengan android *runtime* adalah *libraries*. Android menyertakan satu set *library-library* dalam bahasan C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem android. Kemampuan ini dapat diakses oleh programmer melewati android *application* framework. Sebagai contoh android mendukung pemutaran format audio, video dan gambar.

Berikut beberapa core library tersebut:

1. *System C Library* – variasi dari implemetasi standard C sistem library (libc) milik BSD, dioptimasi untuk piranti embedded berbasis linux.
2. *Media Libraries* – berdasarkan packet video's openCORE, library-library ini mendukung playback dan recording dari berbagai format audio dan video popular, meliputi MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG.
3. *Surface Manager* – Mengelola atau mengatur akses pada display dan lapisan composites 2D dan grafis 3D dari berbagai aplikasi.

4. *LibWebCore* – web browser engine modern yang mensupport android browser maupun embeddable web view.
5. SGL – Mendasari grafis 2D
6. 3D libraries – implementasi berdasarkan OpenGL ES 1.0 APIs, library ini menggunakan perangkat keras akselerasi 3D dan pengoptimalan 3D *software rasterizer*.
7. *FreeType* – bitmap dan vector font rendering
8. SQLite – mesin database yang kuat dan ringan tersedia untuk semua aplikasi.

Library - library 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 NDK (*Native Development Toolkit*).

Android Runtime

Lapisan berikutnya adalah android *runtime* yang berisi *core libraries* dan *dalvik virtual machine*. *Core libraries* mencakup serangkaian inti library java, artinya android menyertakan satu set *library-library* dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada *library-library* dasar bahasa pemprograman java. Dalvik adalah java virtual machine yang memberikan kekuatan pada sistem android. Dalvik VM ini di optimalkan untuk telepon seluler. Setiao aplikasi yang berjalan pada android berjalan pada prosesnya sendiri, dengan *instance* dari dalvik *virtual machine*. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi *virtual machine* dengan

efisien. Dalvik VM dapat mengeksekusi file dengan format .dex (*Dalvik Executable*) yang telah dioptimasi untuk menggunakan minimal memory *footprint*. *Virtual machine* ini *register-based*, dan menjalankan *class-class* yang *decompile* menggunakan "dx" *tool* yang telah disertakan. Dalvik virtual machine menggunakan kernel linux untuk menjalankan fungsi – fungsi seperti *threading* dan *low level memory management*.

Framework Aplikasi

Lapisan selanjutnya adalah *framework* aplikasi, yang mencakup program untuk mengatur fungsi – fungsi dasar smartphone. Application framework merupakan serangkaian tool dasar seperti alokasi resources smartphone, aplikasi telepon, pergantian antar proses atau program dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada *tool – tool* dasar tersebut dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks.

Programmer mendapatkan akses penuh untuk memanfaatkan API (*Android Protocol Interface*) yang juga digunakan core applications. Arsitektur aplikasi didesain untuk menyederhanakan pemakaian kembali komponen – komponen, dimana setiap aplikasi dapat menunjukkan kemampuannya dan aplikasi lain dapat memakai kemampuan tersebut. Mekanisme yang sama memungkinkan pengguna mengganti komponen yang dikehendaki. Di dalam semua aplikasi terdapat servis dan sistem yang meliputi :

a. Satu set *Views* yang dapat digunakan untuk membangun aplikasi meliputi *lists, grids, text boxes, buttons*, dan *embeddable web browser*.

b. *Content Providers* yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain misalnya *contact* atau membagi data yang dimiliki.

c. *Resources Manager,* menyediakan akses ke *non-code resources* misalnya *localized strings, graphics* dan *layout files*.

d. *Notification Manager* yang memungkinkan semua aplikasi untuk menampilkan *custom alerts* pada status bar.

e. *Activity Manager* yang memanage *life cycle* dari aplikasi dan menyediakan *common navigation backstack*.

Applications

Di lapisan teratas terdapat aplikasi itu sendiri. Di lapisan inilah dapat ditemukan fungsi – fungsi dasar smartphone seperti menelepon dan mengirim pesan singkat, menjalankan web browser, mengakses daftar kontak, dan lain – lain. Sebagian penggunan pada lapisan inilah yang paling sering diakses, dimana pengaksesan fungsi – fungsi dasar tersebut melalui *user interface*.

2.2.5 XML pada Android

XML (Extensible Markup Language) adalah bahasa *markup* yang mendifinisikan seperangkat aturan untuk *encoding* dalam dokumen yang bersifat *human-readable* dan *machine-readable*. Pada android sendiri berkas berjenis XML memiliki banyak fungsi, diantaranya :

1. Berkas *Android Manifest*

   Setiap aplikasi android harus memiliki berkas ini (dengan nama AndroidManifest.xml) pada direktori root. Pada berkas ini berisi informasi penting tentang aplikasi untuk sistem android.

2. *Layout*

   Android menyediakan kosakata XML sederhana yang sesuai dengan kelas View dan subclass-nya, seperti untuk widget dan layout.

3. Database

   Database XML adalah persistensi data perangkat lunak sistem yang memungkinkan data yang akan disimpan dalam format XML. Data ini kemudian dapat dilihat, diekspor dan serialisasi ke dalam format yang diinginkan. XML database biasanya berhubungan dengan dokumen berorientasi database.

2.3 *Java [13]*

Di antara sekian banyak perangkat lunak yang sudah ada, kita sering mendengar istilah Java, atau disebut juga Java J2ME (Java 2 Micro Edition), yang merupakan turunan dari bahasa Java, di mana fungsi-fungsinya difokuskan untuk pembuatan aplikasi pada lingkungan telepon seluler.

Java merupakan suatu teknologi perangkat lunak yang di dalamnya mencakup bahasa pemrograman. Selain itu Java juga merupakan suatu platform yang memiliki virtual machine dan library yang diperlukan untuk menulis dan menjalankan suatu program. Pada awalnya java dikembangkan pada lingkungan

komputer oleh Sun Microsystem Inc. dengan tujuan untuk menghasilkan suatu bahasa komputer sederhana tanpa harus terikat pada arsitektur tertentu.

Pada tahun 1995 Sun meluncurkan sebuah browser berbasis Java dengan julukan Hot Java, kemudian diikuti Netscape yang memutuskan untuk membuat browser dengan dilengkapi bahasa Java. Setelahnya, ikut bergabung pula berbagai pengembang ternama diantaranya IBM dan Microsoft. Pada tahun berikutnya, Sun Microsystem Inc. merilis Java Software Development Kit (JDK) pertamanya, yaitu JDK 1.1. Kemudian terus berkembang dari pemrograman applet yang berjalan di browser menjadi pemrograman kelas dunia yang banyak digunakan untuk pengembangan aneka ragam aplikasi. Bahasa pemrograman Java sendiri secara garis besar dikelompokan menjadi 3, yaitu

a. J2SE (*Java 2 Standar Edition*), digunakan untuk mengembangkan aplikasi-aplikasi desktop dan applet.

b. J2EE (*Java 2 Enterprise Edition*) dipergunakan untuk mengembangkan aplikasi-aplikasi berbasis client/server berskala enterprise

c. J2ME (*Java 2 Micro Edition*). J2ME diaplikasikan pada berbagai perangkat kecil dengan jumlah memori, kapasitas penyimpan dan user interface terbatas, seperti ponsel dan PDA.

Hingga saat ini ada dua versi platform Java J2ME yang umum digunakan pada telepon genggam yaitu MIDP 1.0 yang masih memiliki spesifikasi sederhana dan menyediakan fungsi dasar untuk aplikasi mobile, di antaranya basic user interface dan keamanan jaringan. Dan MIDP 2.0 yang sudah ditambahkan

berbagai fasilitas seperti *game*, multimedia, dukungan berbagai jenis konektivitas, maupun OTA.

2.3.1 Sejarah Perkembangan Java

Bahasa pemprograman java pertama lahir dari The Green Project yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas tahun 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot Duke yang dibuat oleh Joe Palrang.

Pertemuan proyek berlangsung di sebuah gedung perkantoran Sand Hill Road di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program java Oak pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*) seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (*Star Seven*).

Setelah era *star seven* selesai, sebuah anak perusahaan tv kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto.

Perusahaan baru ini bertambah maju, jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, internet masih merupakan rintisan yang dipakai hanya di kalangan akademis dan militer.

Mereka menjadikan perambah (*browser*) Mosaic sebagai landasan awal untuk membuat perambah java pertama yang dinamai Web Runner, terinspirasi dari flim 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java.

Pada sekitar bulan maret 1995 untuk pertama kali kode sumber java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995.

Nama Oak diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja James Gosling. Nama Oak ini tidak dipakai untuk versi release java karena sebuah perangkat lunak sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi java. Nama ini diambil dari kopi murni yang digiling langsung dari biji kopi (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari pulau jawa. Jadi nama bahasa pemprograman Java tidak lain berasal dari kata jawa (bahasa inggris untuk jawa adalah java).

2.3.2 Versi Awal Java

Versi awal java di tahun 1996 sudah merupakan versi release sehingga dinamakan java versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya :

a. Java.lang – diperuntukkan kelas elemen – elemen dasar
b. Java.io – diperuntukkan kelas input dan output, termasuk penggunaan berkas
c. Java.util – diperuntukkan kelas pelengkap seperti kelas struktur data dan kelas - kelas penanggalan

d. Java.net – diperuntukkan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
e. Java.awt – kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)
f. Java.applet – kelas dasar aplikasi antar muka untuk diterapkan pada penjelajahan web.

2.3.3 Kelebihan dan Kekurangan Java

Kelebihan java:

a. *Multiplatform*. Kelebihan utama dari java ialah dapat dijalankan di beberapa platform/sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan dimana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin/*bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas operating sistem linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, linux, Mrac OS dan Sun Solaris. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri untuk menginterpretasikan *bytecode* tersebut.

b. OOP (*Object Oriented Programming*) yang artinya semua aspek yang terdapat di java adalah objek. Java merupakan salah satu bahasan pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut Object. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan, dan mengalokasi kesalahan sebuah program dengan basis java secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan java sebagai salah satu bahasa pemrograman termudah, bahkan untuk fungsi yang advance seperti komunikasi antara komputer sekalipun

c. *Library* yang lengkap, dimana java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program-program yang disertakan dalam pemrogram java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

d. Bergaya C++ , memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke java. Saat ini pengguna java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke java.

e. Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan

memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

Kekurangan Java:

a. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini memungkinkan karena kode jadi java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi seperti nama-nama kelas, metode dan tipe data. Hal yang sama juga terjadi pada Microsoft.Net Platform. Dengan demikian , algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/di-*reverse-engineer*.

b. Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnyaseperti C/C++ dan Pascal (lebih spesifik lagi delphi dan *Object Pascal*). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru, tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.3.4 Tahap Kompilasi Java

Tulis/ubah. Pemrogram menulis program dan menyimpannya di media dalam bentuk berkas ‘java’

a. Kompilasi. Pengkompilasi membentuk *bytecodes* dari program menjadi bentuk berkas ‘class’

b. Muat. Pemuat kelas memuat *bytecodes* ke memori.

c. Verifikasi. Peng-verifikasi memastikan *bytecodes* tidak menganggu sistem keamanan java.

d. Jalankan. Penerjemah menerjemahkan *bytecodes* ke bahasa mesin.

2.4 *Augmented Reality (AR) [2]*

Pengertian sederhana dari AR adalah menambahkan objek virtual ke dalam dunia nyata sehingga nampak berdampingan dalam ruang yang sama di dunia nyata. *Augmented Reality* (AR) adalah sebuah istilah untuk lingkungan yang menggabungkan dunia nyata dan dunia virtual serta dibuat oleh komputer sehingga batas antara keduanya menjadi sangat tipis. Menurut Ronald T. Azuma *augmented reality* adalah penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata [Azuma, 1997]. Ronald T. Azuma juga mendefinisikan *Augmented reality* sebagai sistem yang memiliki karakteristik sebagai berikut:

a. Menggabungkan lingkungan nyata dan virtual

b. Berjalan secara interaktif dalam waktu nyata

c. Integrasi dalam tiga dimensi (3D)

Penggabungan objek nyata dan virtual dimungkinkan dengan teknologi *display* yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat input tertentu. Jika menelisik lebih dalam, sesungguhnya AR berbeda dengan Virtual Reality. Virtual *reality* adalah teknologi yang memungkinkan seseorang

melakukan simulasi terhadap suatu objek nyata dengan menggunakan komputer yang mampu membangkitkan suasana tiga dimensi (3D) sehingga membuat pemakai seolah-olah terlibat secara fisik. AR merupakan variasi dari Virtual *Environments* (VE), atau yang lebih dikenal dengan istilah Virtual *Reality* (VR).

Teknologi VR membuat pengguna tergabung dalam sebuah lingkungan virtual secara keseluruhan. Ketika tergabung dalam lingkungan tersebut, pengguna tidak bisa melihat lingkungan nyata di sekitarnya. Sebaliknya, AR memungkinkan pengguna untuk melihat lingkungan nyata, dengan objek virtual yang ditambahkan atau tergabung dengan lingkungan nyata. Tidak seperti VR yang sepenuhnya menggantikan lingkungan nyata, AR sekedar menambahkan atau melengkapi lingkungan nyata [Acko, 2003]. AR merupakan kebalikan dari VR yang berarti integrasi elemen-elemen digital yang ditambahkan ke dalam dunia nyata secara *real time* dan mengikuti keadaan lingkungan yang ada di dunia nyata. Seperti pada gambar di bawah ini, terdapat sebuah meja yang *real* dengan tambahan objek virtual berupa lampu dan 2 kursi. Objek-objek virtual tersebut telah dikombinasikan dalam 3D, sehingga baik dari objek virtual maupun dari objek nyata dapat menutupi satu sama lainnya. Lampu dapat menutupi meja nyata, dan meja nyata menutupi sebagian dari objek virtual kursi.

Kelebihan utama dari a*ugmented reality* dibandingkan *virtual reality* adalah pengembangannya yang lebih mudah dan murah [Kauffman, 2002]. Tak seperti virtual *reality* yang sampai saat ini masih digunakan secara terbatas oleh kalangan tertentu, *augmented reality* merebak secara cepat di berbagai bidang yang bahkan belum dapat dijangkau oleh pendahulunya tersebut.

Disamping itu, tujuan utama dari AR adalah untuk menciptakan lingkungan baru dengan menggabungkan interaktivitas lingkungan nyata dan virtual sehingga pengguna merasa bahwa lingkungan yang diciptakan adalah nyata. Dengan kata lain, pengguna merasa tidak ada perbedaan yang dirasakan antara AR dengan apa yang mereka lihat/rasakan di lingkungan nyata. Dengan bantuan teknologi AR (seperti visi komputasi dan pengenalan pola) lingkungan nyata disekitar akan dapat berinteraksi dalam bentuk digital (virtual). Informasi tentang objek dan lingkungan disekitar pengguna dapat ditambahkan kedalam sistem AR yang kemudian informasi tersebut ditampilkan diatas layer dunia nyata secara *real time* seolah-olah informasi tersebut adalah nyata. Informasi yang ditampilkan oleh objek virtual membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata.

AR banyak digunakan dalam bidang-bidang seperti kesehatan, militer, industri manufaktur dan juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada telepon genggam.

2.4.1 Cara Kerja AR

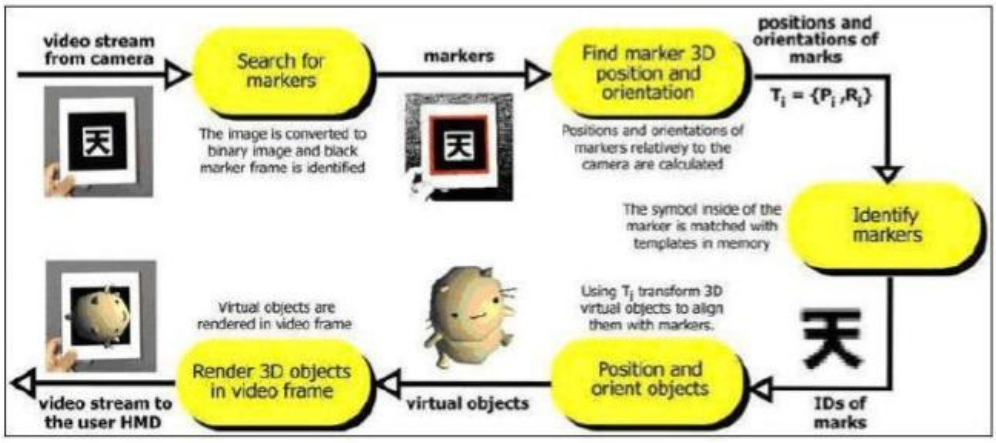
2.4.1.1 Simple AR

Objek 2D atau 3D diciptakan oleh komputer dan diintegrasikan dengan objek atau lingkungan nyata/fisik secara langsung maupun tidak langsung (dilakukan pengeditan terlebih dahulu) kemudian ditampilkan dalam layar. Simple AR ini adalah penerapan teknologi a*ugmented reality* yang paling sederhana dan

paling awal, meskipun begitu masih digunakan secara luas hingga saat ini contohnya dalam acara olahraga di televisi.

2.4.1.2 Marker Based AR

Objek 2D, 3D, teks, video maupun suara diproses menggunakan komputer dan webcam dan ditampilkan dalam layar maupun peralatan *display* khusus melalui pengenalan sebuah *marker* (penanda), setelah *marker* dikenali oleh komputer kemudian objek virtual yang sudah terdapat dalam *library* komputer ditampilkan diatas *marker* tersebut. Marker based AR mempunyai 3 sumbu yaitu X, Y dan Z yang bias menghasilkan virtual 3D dengan titik (0,0,0) Untuk lebih jelasnya dapat dilihat dalam diagram dibawah ini:



Gambar 2-1 AR Flow

Marker based AR telah menjadi tren sejak awal tahun 2010 hingga sekarang dan diimplementasikan pada berbagai media khususnya media cetak. Tidak seperti realitas maya yang sepenuhnya menggantikan knyataan, namun AR harus menambahkan atau melengkapi kenyataan. Pada gambar di bawah keadaan nyata sebuah kubus tidak ada , namun dengan menggunakan AR terlihat sebuah kubus.



Gambar 2-2 Contoh Augmented Reality – gambar kanan sudah ditambahkan augmented reality

2.4.1.3 Markeless AR [11]

Umumnya diterapkan dalam *mobile device* seperti *smartphone*. Sesuai dengan namanya *markerless* AR tidak membutuhkan *marker* yang terlihat secara fisik untuk mengetahui posisi suatu objek. Metode AR ini tidak memerlukan marker lagi dalam menampilkan elemen – elemen digital. Macam – macamnya antara lain :

a. *Face Tracking* – Mengenali wajah manusia secara umum saat dihadapkan pada web cam (kamera) yaitu mulai dari posisi mata, hidung, dan mulut, tetapi mengabaikan objek lain disekitarnya.



Gambar 2-3 Face Tracking

b. *3D Object Tracking* – Penyempurnaan dari face tracking dimana objek – objek sekitar juga bias dikenali secara keseluruhan.



Gambar 2-4 3D Object Tracking

c. *Motion Tracking* – menangkap gerakan dari manusia atau objek. Dipakai dalam pembuatan film yang mencoba mensimulasikan gerakan.



Gambar 2-5 Motion Tracking

Sebagai gantinya, digunakan informasi dari GPS atau kompas dan cara ini dikenal dengan nama *geotagging* dan *geolocation*. Melalui *geotagging* dan *geolocation* yang bisa disebut sebagai *marker* yang tidak terlihat inilah konten seperti tulisan, video, maupun audio kemudian ditampilkan dilayar *mobile device* tersebut. Contoh aplikasi AR dalam *smartphone* yang menggunakan *geotagging* dan *geolocation* yaitu Wikitude Browser dan layar.

Pada tracking markerless, metaio menyediakan dua macam metode yakni *fast* dan *robust*. Metode *fast* sendiri merupakan metode yang standar atau yang umum digunakan pada tracking markerless dimana metode ini hanya dapat digunakan tanpa kondisi pencahayaan yang beragam, sedangkan metode *robust* digunakan pada target tracking yang bertekstur tinggi dan memiliki ambang yang tetap untuk jumlah iterasi pencarian sehingga dapat digunakan dalam situasi dimana kondisi pencahayaan dapat berubah , atau dengan kata lain metode *fast* tidak memberikan hasil yang memuaskan.

Perbandingan markeless dengan metode *fast* dan *robust* dapat dilihat pada gambar 2-6 (dihasilkan dari penilain gadget yang memiliki kecepatan processor 1 GHz, sedangkan untuk jumlah target dapat bervariasi, bergantung pada ketersediaan memori pada gadget).

	Markerless (FAST)	Markerless (ROBUST)
Robustness	★★★	★★★
Accuracy	★★★★	★★★★
Speed	★★★★	★★★
Number of Targets Configured	up to 10	up to 3
iPhone	✓	✓
Android	✓	✓
Windows Mobile	✓	✓
Symbian	✓	✓

Gambar 2-6 Perbandingan metode markerless *FAST* dan *ROBUST*

2.4.1.3.1 Proses Tracking Markerless [17]

Salah satu pendekatan tracking markerless yang dapat digunakan adalah dengan template matching. Template matching sendiri merupakan teknik dalam *digital image processing* untuk menemukan bagian dari gambar (target) yang cocok dengan template image. Untuk proses pencarian kecocokan digunakan metode normalisasi *cross correlation* (NCC) [18]. Dengan normalisasi *cross correlation* dapat diketahui tingkat kecocokan (*similarity*) antara *reference image* dengan target, ini dapat dilakukan dengan mencari nilai akhir yang memiliki nilai mendekati $0.999 \approx 1$. Berikut rumus perhitungan dari normalisasi *cross correlation* :

$$\text{NCC} = \frac{1}{n-1}\sum_{x,y}\frac{(f(x,y)-\overline{f})(t(x,y)-\overline{t})}{\sigma_f \sigma_t}$$

Salah satu komponen dari normalisasi *cross correlation* adalah mean dan standar deviasi (σ) ,dimana mean merupakan nilai rata – rata dan standar deviasi atau simpangan baku sendiri adalah ukuran sebaran statistik yang paling lazim, yang mengukur bagaimana nilai-nilai data tersebar, bisa juga didefinisikan sebagai, rata-rata jarak penyimpangan titik-titik data yang diukur dari nilai rata-rata data tersebut. Berikut rumus perhitungan dari standar deviasi :

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\overline{x})^2}$$

2.4.1.4 GPS *Based Tracking*

Tracking ini memanfaatkan teknik GPS dan kompas yang ada dalam *smartphone*, maka jika *smartphone* kita arahkan ke suatu objek tempat atau gedung, maka dapat ditampilkan berbagai macam informasi mengenai objek tersebut bahkan ada beberapa aplikasi yang sudah bisa menampilkan secara 3D.

Informasi dari GPS atau kompas dan cara ini dikenal dengan nama *geotagging* dan *geolocation*. Melalui *geotagging* dan *geolocation* yang bisa disebut sebagai *marker* yang tidak terlihat inilah konten seperti tulisan, video, maupun audio kemudian ditampilkan dilayar *mobile device* tersebut. Contoh aplikasi AR dalam *smartphone* yang menggunakan *geotagging* dan *geolocation* yaitu Wikitude Browser dan layar.



Gambar 2-7 GPS based Tracking

2.5 *Unified Modeling Language (UML) [14]*

Unified Modeling Language (UML) merupakan bahasa standar yang bekerja dalam *object-oriented* untuk menentukan, memvisualisasikan, merancang, dan mendokumentasikan elemen-elemen informasi yang terdapat dalam sistem *software*. UML mulai diperkenalkan oleh *Object Management Group*, sebuah

organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh grady Booch, Jim Rumbaugh dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.



Gambar 2-8 Logo UML

2.5.1 Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu :

a. *Use Case Diagram* untuk memodelkan proses bisnis

b. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi

c. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects*

d. *Collaboration Diagram* untuk memodelkan interaksi antar *objects*

e. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem

f. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan *Objects* di dalam sistem

g. *Class Diagram* untuk memodelkan struktur kelas

h. *Object Diagram* untuk memodelkan struktur object

i. *Component Diagram untuk memodelkan komponen object*

j. *Deployment Diagram* untuk memodelkan distribusi aplikasi

Berikut akan dijelaskan empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu :

a. *Use Case* diagram, untuk memodelkan proses bisnis dan merepresentasikan sebuah interaksi antara aktor dengan sistem

b. *Sequence* diagram, untuk menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus 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 yakni nama (dan stereotype) , atribut , dan metoda .

c. *Statechart diagram* , untuk memodelkan perilaku objects di dalam sistem atau menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem. Pada umumnya statechart diagram menggambarkan class tertentu .

d. *Activity diagram* , untuk memodelkan perilaku *Use Cases* dan *objects* di dalam system.

e. *Sequence diagram* , untuk memodelkan pengiriman pesan (*message*) antar objects dan juga digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu.

f. *Collaboration diagram* , untuk memodelkan interaksi antar objects seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*

g. *Component diagram* , untuk memodelkan komponen object atau menggambarkan struktur dan hubungan antar modul yang berisi code, termasuk ketergantungan (*dependency*) di antaranya.

h. *Deployment diagram*, untuk memodelkan distribusi aplikasi.

2.6 Tools *Pendukung Pembuatan Aplikasi Android*

2.6.1 Java Development Kit (JDK)

JDK adalah software development kit merupakan seperangkat kode, library, form dan paket yang berguna untuk membuat aplikasi atau applet java. Di dalam JDK terdapat *Java Runtime Environment* (JRE) yang akan memproses *byte code* yang dihasilkan dari interpenter java. Tugas dari interpenter java ini adalah

untuk mengcompile source.java menjadi source.class yang selanjutnya source.class inilah yang akan dijalankan oleh JRE.

2.6.2 Software Development Kit (SDK)

SDK adalah suatu kumpulan dari library dan toolkit seperti emulator untuk mengembangkan atau menciptakan aplikasi untuk suatu perangkat lunak, perangkat computer, system operasi atau platform. Di dalam SDK terdapat *tools* yang dibutuhkan dalam pengembangan android, diantaranya yaitu adb shell yang merupakan singkatan dari android development bridge yang dapat menjalankan terminal android seperti terminal pada system operasi linux, dan *command* yang terdapat dalam adb shell sendiri seperti command linux pada umumnya.

2.6.2.1 Metaio Mobile SDK

Pengembangan solusi untuk augmented reality dapat menggunakan platform perangkat lunak metaio. Desain pada metaio saat ini banyak digunakan di pameran pemasaran, industry, aplikasi komersial untuk AR. Desain pada metaio dari sederhana ke kompleks dengan maksud untuk memudahkan pembuatan presentasi AR, pelacakan gambar, dll. Metaio mendukung dua mode render yang berbeda yaitu DirectX dan OpenGLTo.

Pada metaio mobile SDK mendukung dua platform utama IOS dan Android. SDK terdiri dari tiga lapisan yang berbeda. Interface utama adalah metaio-*unifeyemobile* dan antarmuka khusus metaio-*unifeyemobileiphone* dan metaio-*unifeyemobileandroid*. Metaio dianggap sebagai innovator terkemuka

untuk salah satu tren perangkat lunak utama seperti halnya augmented Reality (AR). Dengan AR, informasi digital tentang tempat-tempat dan benda-benda dimasukkan dengan cara yang terlihat alami ke dalam lingkungan nyata, misalnya sebagai gambar bergerak atau animasi 3D menggunakan antarmuka melalui terminal mobile seperti smartphone atau tablet.

Metaio mobile SDK menyediakan teknologi *tracking* optik dan non-optik. Proses *tracking* tersebut dapat terjadi dengan menggunakan konfigurasi file XML (yang juga dikenal sebagai *tracking data file*, atau database marker). Pada berkas ini berisi informasi mengenai *reference image* yang akan digunakan pada proses *tracking* (nama berkas, tinggi dan lebar gambar) dan jenis *tracking* yang akan digunakan.

2.6.3 Eclipse

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

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

b. **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.

c. **Multi-role**: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu Kelebihan dari eclipse yang membuatnya popular adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plugin*.

Secara standar Eclipse selalu dilengkapi dengan *Java Development Tools* (JDE), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan *Plug-in Development Environment* (PDE) untuk 27 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. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*).