BAB 2
LANDASAN TEORI

2.1 *Al-Qur'an*

Al-Qur'an menurut bahasa berarti bacaan atau yang dibaca. Menurut istilah, *Al-Qur'an* adalah wahyu Allah *Subhanahu wa ta'ala* yang diturunkan kepada nabi Muhammad *Shallallahu 'alaihi wasallam* melalui malaikat Jibril *'Alaihissalam* sebagai petunjuk bagi umat manusia. *Al-Qur'an* diturunkan untuk menjadi pegangan bagi mereka yang ingin mencapai kebahagian dunia dan akhirat. *Al-Qur'an* menggunakan bahasa Arab dan merupakan mukjizat bagi rasul. Sebagian besar ayat-ayat *Al-Qur'an* diturunkan di kota Mekah dan kota Madinah. Isi yang terkandung dalam *Al-Qur'an* terdapat 6236 ayat 114 surat dan 30 juz.

2.2 Pembelajaran

Pembelajaran adalah adalah proses interaksi peserta didik dengan pendidik dan sumber belajar pada suatu lingkungan belajar. Dengan kata lain, pembelajaran adalah proses untuk membantu peserta didik agar dapat belajar dengan baik. Dalam konteks pendidikan, guru mengajar agar peserta didik dapat belajar dan menguasai isi pelajaran hingga mencapai sesuatu objektif yang ditentukan (aspek kognitif), juga dapat memengaruhi perubahan sikap (aspek afektif), serta keterampilan (aspek psikomotor) seorang peserta didik, namun proses pengajaran ini memberi kesan hanya sebagai pekerjaan satu pihak, yaitu pekerjaan pengajar saja.

2.3 Pengertian Media

Kata media berasal dari bahasa latin dan merupakan bentuk jamak dari kata medium yang secara harfiah berarti perantara atau pengantar. Media adalah segala sesatu yang dapat digunakan untuk menyalurkan pesan dari pengirim ke penerima[3]. Berikut beberapa unsur dari media yaitu :

1. Orang
2. Bahan atau material

3. Alat
4. Teknik
5. Lingkungan

2.3.1 Media Pembelajaran

Media pembelajaran adalah bahan, alat atau teknik yang digunakan dalam kegiatan mengajar dengan maksud agar proses interaksi komunikasi edukasi dapat berlangsung secara tepat guna dan berdaya guna[4]. Media pembelajaran berfungsi sebagai alat penunjang dalam membantu proses pembelajaran sehingga dapat menstimulasi pikiran, perhatian dan minat dalam mempelajari ilmu pengetahuan serta dapat membantu tercapainya tujuan pembelajaran. Secara umum manfaat dari media pembelajaran adalah sebagai berikut :

1. Memperjelas penyajian agar tidak terlalu bersifat verbalistis.
2. Mengatasi keterbatasan ruang,waktu dan daya indra.
3. Mengatasi sikap pasif pembelajar.
4. Mengatasi kesulitan dalam pembelajaran.

2.4 Pengertian Multimedia

Multimedia adalah media yang menggabungkan dua unsur atau lebih media yang terdiri dari teks, grafik, gambar, audio, video dan animasi secara terintegrasi[5]. Multimedia terbagi menjadi dua kategori yaitu :

1. Multimedia *Linier* adalah suatu multimedia yang tidak dilengkapi dengan alat pengontrol apapun yang dapat dioperasikan oleh pengguna. Multimedia *linier* berjalan secara sekuensial. Contoh dari multimedia *linier* adalah TV dan Film.
2. Multimedia Interaktif adalah suatu multimedia yang dilengkapi dengan alat pengontrol yang dapat dioperasikan oleh pengguna sehingga pengguna dapat memilih apa yang dikehendaki untuk proses selanjuntya. Contoh multimedia interaktif adalah multimedia pembelajaran interaktif, aplikasi *game* dan lain-lain.

Menurut beberapa ahli multimedia memiliki beberapa definisi yaitu :

1. Rosch

   Multimedia adalah kombinasi dari komputer dan video.

2. McComick

   Multimedia adalah kombinasi dari komputer dari tiga elemen yaitu suara, gambar dan teks.

3. Turban dan kawan-kawan

   Multimedia adalah kombinasi dari paling sedikit dua media *input* dan *output*. Media ini dapat berupa audio, animasi, video, teks, grafik dan gambar.

4. Robin dan Linda

   Multimedia adalah alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio dan video.

5. Hofstetter

   Multimedia adalah pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, audio, video dengan menggunakan *tool* yang memungkinkan pemakai berinteraksi, berkreasi dan berkomunikasi. Dalam definisi ini terdapat empat komponen multimedia yaitu :

   a. Harus ada peralatan elektronik yang dapat mengkoordinasikan apa yang dapat dilihat dan didengar dan yang dapat berinteraksi dengan pengguna.
   b. Harus ada *link* yang dapat menghubungkan pengguna dengan informasi.
   c. Harus ada alat navigasi yang dapat memandu pengguna untuk menjelajahi jaringan informasi yang saling terhubung.
   d. Harus dapat menyediakan tempat kepada pengguna untuk mengumpulkan, memproses dan mengkomunikasikan informasi dan ide dari pengguna itu sendiri.

Penggunaan multimedia dapat digunakan di beberapa bidang seperti:

1. Bidang periklanan yang efektif dan interaktif.

2. Bidang pendidikan dalam penyampaian bahan pengajaran secara interaktif dan dapat mempermudah pembelajaran karena didukung oleh berbagai aspek seperti suara, video, animasi, teks dan gambar.
3. Bidang jaringan dan *internet* yang dapat membantu dalam pembuatan *website* yang menarik, informatif dan interaktif.

2.4.1 Objek Media

Multimedia terdiri dari beberapa objek media yaitu :

1. Teks
2. *Image*
3. Audio
4. Video
5. Animasi

1. Teks

   Teks merupakan dasar dari pengolahan kata dan informasi berbasis multimedia. Teks mudah disimpan dan membentuk kata, surat atau narasi dalam multimedia. Kebutuhan teks tergantung pada kegunaan aplikasi multimedia. *File* teks memiliki struktur *linear* sederhana. Dalam multimedia, teks sangat banyak digunakan karena teks sangat efektif untuk menyampaikan ide serta memberikan panduan kepada pengguna.

   Teks memiliki beberapa jenis yaitu :

   a. *Plaint Text*

      Plaint text adalah jenis teks data dalam bentuk karakter. Teks dalam hal ini adalah kode ASCII (*American Standart Code for Information Interchange*) dan ASCII *extension*.

   b. *Format Text*

      Teks dengan serangkaian karakter yang telah didefinisikan.

   c. *Hypertext*

      Teks yang memiliki fasilitas *linking*.

2. *Image*

Image atau gambar merupakan presentasi multimedia yang lebih menarik perhatian dan dapat mengurangi kebosanan dibandingkan dengan teks. Gambar dapat meringkas dan menyajikan data kompleks dengan cara yang baru dan lebih berguna. Gambar juga dapat berfungsi sebagai ikon yang bila dipadukan dengan teks, menunjukkan berbagai opsi yang bisa dipilih atau gambar bisa muncul *full screen* menggantikan teks, tapi tetap memiliki bagian-bagian tertentu yang berfungsi sebagai pemicu yang bila diklik akan menampilkan objek atau *event* multimedia lain. Titik dimana suatu gambar di sampling disebut *pixel* sedangkan nilai intensitas warna pada suatu *pixel* disebut *gray scale level*. Berikut nilai intensitas warna pada suatu *pixel* yaitu :

a. 1 bit : *binary value image* (0-1).
b. 8 bits : *gray level* (0-255).
c. 16 bits : *high color* (2^{16}).
d. 24 bits : 2^{24} *true color*.
e. 32 bits : *true color* (2^{32}).

Berikut beberapa format *file* dari suatu gambar yaitu :

a. *Bitmap* (.BMP)
    1. Format gambar yang paling umum dan dapat merupakan standar *windows*.
    2. Ukuran *file*nya sangat besar.
    3. *File* ini merupakan format yang belum terkompresi dan menggunakan sistem warna RGB (*Red*, *Green*, *Blue*) dimana masing-masing *pixel*-nya terdiri dari 3 komponen R, G, B yang dicampur menjadi satu.
    4. Tidak bisa digunakan di *web* karena ukurannya besar.
b. *Joint Photographic Expert Group* (.JPEG/JPG)
    1. Berukuran kecil dan bersifat *portable*.
    2. Sering digunakan pada bidang fotografi untuk menyimpan *file* foto.

3. *File* ini bisa digunakan di *web*.

c. *Graphic Interchange Format* (.GIF)
   1. Format GIF ini berikuran kecil dan mendukung gambar yang terdiri dari banyak *frame* sehingga bisa disebut sebagai gambar animasi.
   2. Sering digunakan di *internet* untuk menampilkan gambar-gambar di *web*.

d. *Portable Network Graphic* (.PNG)

   Format standar dan sering digunakan di *internet* untuk menampilkan gambar atau pengiriman gambar. Ukuran *file* ini cukup kecil dan setara dengan ukuran GIF dengan kualitas yang lebih bagus namun tidak mendukung animasi.

3. Audio

Fenomena fisik yang dihasilkan oleh benda yang berupa sinyal analog dengan amplitudo yang berubah secara berkelanjutan terhadap waktu. Pola isolasi yang terjadi dinamakan sebagai gelombang. Gelombang yang memiliki pola yang berulang pada interval tertentu disebut periode sedangkan frekuensi adalah banyaknya periode dalam 1 detik. Berdasarkan frekuensi, audio dibagi menjadi beberapa jenis yaitu :

a. *Infrasound* : 0 Hz – 20 Hz.
b. Pendengaran manusia : 20 Hz – 20KHz.
c. *Ultrasound* : 20 KHz – 1 GHz.
d. *Hypersound* : 1 GHz – 10 GHz.

Berikut beberapa format *file* audio yaitu :

a. *ACC* (.m4a)
   1. Bersifat *lossy compression*.
   2. Audio *codec* yang menyempurnakan MP3 dalam hal medium dan *high* bit *rates*.

b. *Wave Audio* (.WAV)
   1. Menggunakan *coding* PCM (*Pulse Code Modulation*).

2. Data tidak terkompres sehingga seluruh sample audio disimpan semuanya di *harddisk*.

c. *Audio Interchange File Format* (.AIF)
   1. Merupakan format standar Macintosh.
   2. *Software* pendukung yaitu Apple Quick Time.

d. *Mpeg Audio Layer* 3 (.mp3)
   1. File dengan *lossy compression*.
   2. Kompresi dilakukan dengan menghilangkan bagian-bagian bunyi yang kurang berguna bagi pendengaran manusia.

4. Video

   Teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak. Biasanya menggunakan film *seluloid*, sinyal elektronik atau media digital. Digital video adalah jenis video *recording* yang bekerja menggunakan sistem digital dibandingkan dengan analog dalam hal representasi videonya. Video kamera menggunakan 2 teknik yaitu :

   a. *Interlaced* adalah untuk menampilkan gambar dalam *raster scanned display device* seperti CRT dan telivisi *analog*.
   b. *Progressive scan* adalah metode untuk menampilkan,menyimpan dan memancarkan gambar dimana setiap baris untuk setiap *frame* digambar secara berurutan.

   Berikut beberapa format *file* dari video yaitu :

a. Digital Video Compressed.
b. Analog.
c. ASF.
d. MOV.
e. MPEG.
f. DivX.

5. Animasi

   Animasi merupakan *illusion of motion* yang dibuat dari *image* statis yang ditampilkan secara berurutan. Animasi mengacu kepada teknik dimana setiap *frame* dalam film dibuat secara terpisah. *Frame* dapat dihasilkan dari komputer, fotografi atau gambar lukisan.

   Berikut beberapa bentuk jenis animasi yaitu :

   a. Animasi *Cel*
      1. Merupakan lembaran-lemabaran yang membentuk animasi tunggal. Masing-masing sel merupakan bagian terpisah.
      2. Biasanya digambar menggunakan tangan.
   b. Animasi *Frame*
      1. Bentuk animasi yang paling sederhana.
      2. Dalam sebuah film,serangkaian *frame* bergerak dengan kecepatan minimal 24 *frame* per detik agar tidak terjadi *jitter*.
   c. Animasi *Sprite*
      1. Gambar digerakkan dengan latar belakang yang diam.
      2. Bagian dari animasi yang bergerak secara mandiri.
      3. Hanya dapat mengedit dari layar yang mengandung *sprite*.
   d. Animasi *Path*
      1. Animasi dari objek yang bergerak sepanjang garis kurva yang ditentukan sebagai lintasan.
      2. Semua animasi *path* dilakukan *looping* yang membuat gerakan *path* terjadi secara terus menerus.
   e. Animasi *Spline*

      Represenstasi metematis kurva sehingga gerakan objek tidak hanya mengikuti garis lurus melainkan berbentuk kurva.
   f. Animasi Vektor
      1. Garis yang memiliki ujung pangkal, arah dan panjang.
      2. Menggunakan rumus matematika untuk menggambarkan *sprite*.

g. Animasi karakter

Biasanya terdapat di film kartun. Semua bagian dalam film kartun selalu bergerak bersamaan.

2.5 Aplikasi

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

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

2.6 Adobe Photoshop

Adobe Photoshop merupakan suatu aplikasi yang paling populer dikalangan desainer grafis dan pengolahan citra (*image processing*). Sejak pertama diperkenalkan oleh Adobe corporation pada dekade 90-an, Photoshop langsung mendapatkan hati dikalangan profesional dan praktisi *imege-editing* dengan segala kecanggihan fitur dan kemampuan yang maksimal serta kemudahan dalam penggunaannya (*user friendly*).

Adobe Photoshop yang dirilis pada tahun 2003 yang lalu oleh Adobe corp., adalah Adobe Photoshop CS dengan slogan yang cukup menjanjikan *The Professional Standard in Desktop Digital Imaging* merupakan pengembangan dari Adobe Photoshop generasi sebelumnya, dan pada edisi terakhir ini sudah dilengkapi dengan fasilitas *Image Ready CS*. Kelengkapan ini dimaksudkan untuk memberikan fleksibelitas, efisiensi dan kemudahan dalam penggunaan, sehingga dapat dengan mudah membuat dan menyunting *image* dengan kualitas tinggi yang siap untuk dicetak, *publisher*, ataupun untuk ditempatkan di *web*, yang selama ini merupakan kendala untuk menempatkan *image* berkualitas tinggi pada *web*, karena berhubungan erat dengan beban *loading* data di sisi *client* pengguna.

Adobe Photoshop CS, yang kemudian hanya disebut Photoshop CS saja, merupakan program aplikasi yang canggih yang telah membantu pekerjaan para profesional dalam bidang *image processing*, seperti fotografi, pracetak, periklanan dan *web designer* serta berbagai disiplin pekerjaan lainnya yang berhubungan dengan *imaging*.

Adobe Photoshop CS3 disusun untuk membantu pekerjaan yang fokus pada pembuatan dan pengeditan gambar. Menu ruang kerja yang ada berupa *tools* dan *palettes* yang berfungsi untuk melihat, mengedit, dan menambahkan elemen ke gambar yang diinginkan[6].

2.6.1 Area Kerja Photoshop

Beberapa Area kerja Photoshop, diantaranya :

a. Menu Bar

Berisi perintah utama untuk membuka *file*, *save*, mengubah ukuran gambar, *filter* dan lain-lain.

b. *Option*

Berisi pilihan dari *tool* yang dipilih. Misalnya dipilih kuas/*brush*, maka ukuran/diameter *brush* ada di sini.

c. Gambar

Menampilkan gambar yang sedang dibuat atau diperbaiki.

d. *Pallete Well*

Cara cepat untuk mengakses *palet brushes*, *tool resets* dan *Layer Comps*. Juga dapat digunakan untuk meletakkan palet yang sering digunakan.

e. *Toolbox*

Berisi *tool* untuk menyeleksi dan memodifikasi gambar.

f. *Palette*

Berisi jendela-jendela kecil yang di dalamnya terdapat perintah dan pilihan untuk dokumen/gambar yang sedang dikerjakan.

2.7 Android

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

2.7.1 Sejarah Android

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

Android Inc didirikan di Silicon Valley, California, pada bulan Oktober 2003, dengan gagasan menyediakan platform *mobile* yang lebih menyadari akan lokasi dan keinginan pengguna. Google mengakuisisi Android Inc pada bulan Agustus 2005 sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google Inc. Tujuan utama Google adalah untuk menyediakan *open* platform yang didukung oleh teknologi Google, baik untuk pengguna dan aplikasi pengembang. Pada bulan November 2007, Open Handset Alliance didirikan sebagai konsorsium untuk mengembangkan *open standard* perangkat *mobile*. Open Handset Alliance memulai perjalanannya dengan mengumumkan platform Android. Dalam waktu kurang dari satu tahun, anggota baru mulai bergabung dengan konsorsium ini[7].

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*Mobile*) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

1. Android 1.0 beta

Android Beta pertama kali dirilis tanggal 5 November 2007 dan versi SDK (*Software Development Kit*) beta dirilis pada tanggal 12 November 2007.

2. Android 1.0

Android 1.0 merupakan versi *software* Android komersial pertama yang dirilis pada tanggal 23 September 2008. Dan perangkat Android pertama yang menggunakan dan memperkenalkan Android 1.0 adalah ponsel HTC Dream(G1).

3. Android versi 1.1

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

4. Android versi 1.5 (Cupcake)

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

5. Android versi 1.6 (Donut)

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

6. Android versi 2.0/2.1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan *Google Maps* 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth* 2.1.

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

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

7. Android versi 2.2 (Froyo: Frozen Yoghurt)

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

8. Android versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*,

layar antarmuka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

9. Android versi 3.0 (Honeycomb)

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

10. Android versi 4.0 (Ice Cream Sandwich)

SDK untuk Android 4.0 atau Ice Cream Sandwich dirilis untuk publik pada 19 Oktober 2011 dan berbasis Linux kernel 3.0.1. Gabe Cohen dari Google menegaskan bahwa Android 4.0 secara teoritis cocok dengan perangkat Android 2.3 keatas dalam produksi masa itu. Android ICS 4.0 membawa fitur Honeycomb untuk *smartphone* dan menambahkan fitur baru berupa membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *offline*, dan berbagi informasi dengan menggunakan NFC. *Source code* untuk Android 4.0 tersedia secara publik pada tanggal 14 November 2011. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

11. Android versi 4.1 (Jelly Bean)

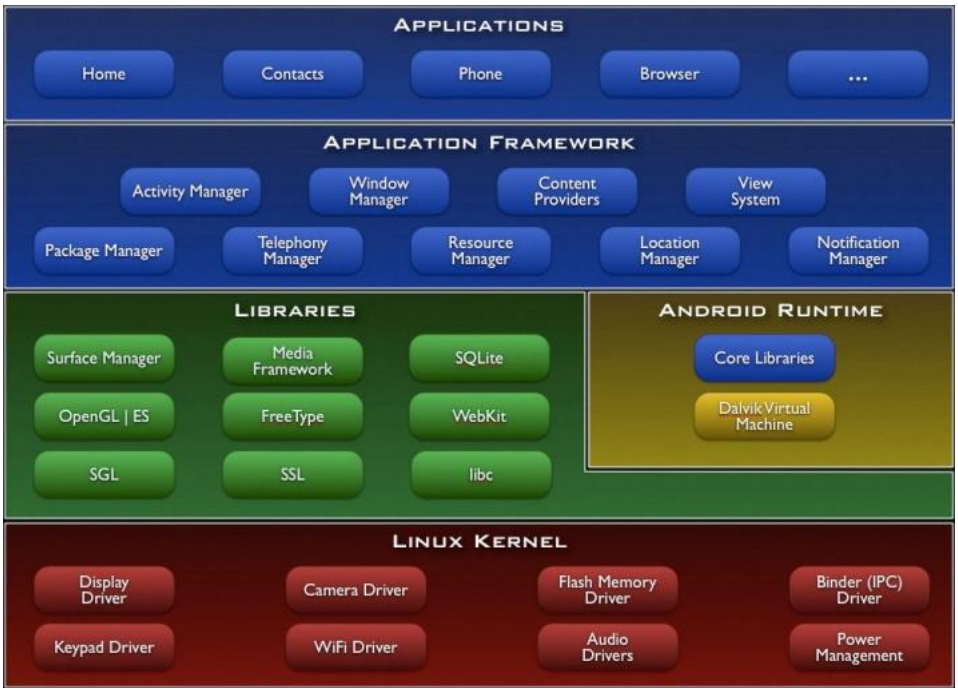
Android ini diluncurkan pada acara Google *I/O* lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan *input keyboard*, desain baru fitur pencarian, UI yang baru dan pencarian melalui *voice search* yang lebih tepat.

Google Now juga menjadi bagian yang diperbarui. *Google Now* memberikan iformasi yang tepat pada waktu yang tepat juga. Salah satu

kemampuannya adalah dapat mengetahui informasi cuaca, lalu lintas, ataupun hasil pertandingan olahraga. Sistem operasi tipe ini muncul pertama kali dalam produk tablet Asus, yakni Google Nexus 7.

2.7.2 Anatomi Aplikasi Android

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



Gambar 2.1 Detail Anatomi Android[8]

2.7.2.1 Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan Android bukanlah linux, karena dalam Android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada Android hanya terdapat beberapa pelayanan yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan *driver*. Kernel linux menyediakan *driver*

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

2.7.2.2 *Libraries*

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

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

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

2.7.2.3 Android *Runtime*

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

1. Pustaka Inti, Android dikembangkan melalui bahasa pemrograman Java, tapi Android *Runtime* bukanlah mesin virtual Java. Pustaka inti Android menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus Android.

2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format *Dalvik Executable* (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

2.7.2.4 *Application Framework*

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

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

2.7.2.5 *Application Layer*

Puncak dari diagram arsitektur Android adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan

tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam Android *runtime* dengan menggunakan kelas dan service yang tersedia pada *Framework* aplikasi.

Lapisan aplikasi Android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada Android semua aplikasi, baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (*Application Programming Interface*) yang sama.

2.7.3 Komponen Aplikasi Android

Fitur penting Android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur *scroller* dan aplikasi lain telah mengembangkan fitur *scroller* yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup menggunakan *scroller* yang telah ada.

Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya Android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi main() pada aplikasi lain. Sebaliknya, aplikasi memiliki komponen penting yang memungkinkan sistem untuk memanggil dan menjalankan ketika dibutuhkan.

2.7.3.1 *Activities*

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *Activity* menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap *Activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari *Views* dan respon terhadap *Event*. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi dijalankan.

Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack*. Keadaan suatu *activity* ditentukan oleh

posisinya dalam tumpukan *acitivity*, LIFO (*Last In First Out*) dari semua aplikasi yang sedang berjalan. Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *Back*, atau menutup *activity* yang sedang digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory Manager* Android menggunakan tumpukkan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut.

Ketika *activity* diambil dan disimpan dalam tumpukkan *activity* terdapat 4 kemungkinan kondisi transisi yang akan terjadi :

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

4) ***Inactive***, kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. *Inactive activity* telah ditiadakan dari tumpukan *activity* sehingga perlu *restart* ulang agar dapat tampil dan digunakan kembali.

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

2.7.3.2 *Services*

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

2.7.3.3 *Intents*

Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman *web*, dan lain sebagainya. *Intents* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi suatu hal, misal pesan masuk. *Intents* dapat eksplisit atau implisit, contohnya jika suatu aplikasi ingin menampilkan *URL*, sistem akan menentukan komponen apa yang dibutuhkan oleh *Intents* tersebut.

2.7.3.4 *Broadcast Receiver*

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

dapat menggunakan *Notification Manager* untuk memberitahukan sesuatu kepada pengguna.

2.7.3.5 *Content Providers*

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

2.7.4 Tipe Aplikasi Android

Terdapat tiga kategori aplikasi pada Android :

1. ***Foreground Activity***

   Aplikasi yang hanya dapat dijalankan jika tampil pada layar dan tetap efektif walaupun tidak terlihat. Aplikasi dengan tipe ini pasti mempertimbangkan siklus hidup *activity*, sehingga perpindahan antar *activity* dapat berlangsung dengan lancar.

2. ***Background Service***

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

3. ***Intermittent Activity***

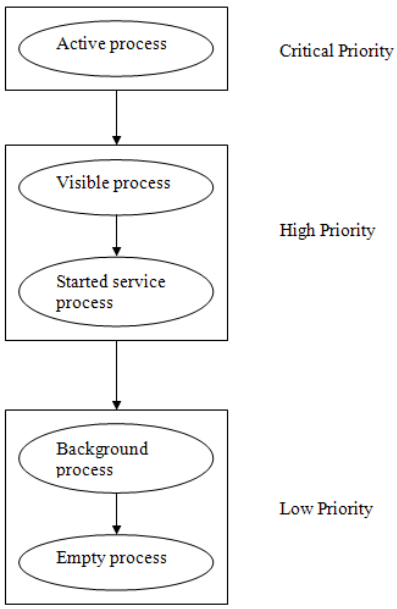
   Aplikasi yang masih membutuhkan beberapa masukkan dari pengguna, namun sebagian sangat efektif jika dijalankan di *background* dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu. Contohnya pemutar musik.

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

2.7.5 Siklus Hidup Android

Siklus hidup aplikasi Android dikelola oleh sistem, berdasarkan kebutuhan pengguna, sumberdaya yang tersedia, dan sebagainya. Misalnya Pengguna ingin menjalankan *browser web*, pada akhirnya sistem yang akan menentukan menjalankan aplikasi. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika pengguna ketika itu sedang menjalankan sebuah *Activity*, maka sistem akan memberikan perioritas utama untuk aplikasi yang tersebut. Sebaliknya, jika suatu *Activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *Activity* yang prioritas rendah akan ditutup.

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin *virtual* pengolah sendiri, dengan ini melindungi penggunaan memori pada aplikasi. Selain itu juga Android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karenanya Android sangat sensitive dengan siklus hidup aplikasi dan komponen-komponennya. Perlu adanya penanganan terhadap setiap kondisi agar aplikasi menjadi stabil. Gambar 2.2 menunjukkan prioritas dari aplikasi.



Gambar 2.2 Prioritas Aplikasi Berdasarkan *Activity*[9]

2.7.6 Kelebihan Android

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

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar royality. Sementara pengembang *software* menyukai karena Android dapat digunakan diperangkat manapun dan tanpa terikat oleh *vendor* manapun.
2. Arsitektur komponen dasar Android terinspirasi dari teknologi *internet Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, *database* SQL, *browser* dan penggunaan peta. Semua itu sudah tertanam pada Android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suara terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh Flash menyatu dalam 3D menggunakan OpenGL memungkinkan membuat aplikasi maupun *game* yang berbeda.
6. *Portabilitas* aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin *virtual*

Dalvik, sehingga kode program *portable* antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard*, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.8 *Rational Unified Process* (RUP)

Rational Unified Process merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak.

2.9 Konsep Perancangan Berorientasi Obyek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah mempermudah *programmer* didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata. Suatu perusahaan *software* yaitu *Rational Software*, telah membentuk konsarium dengan berbagai organisasi untuk meresmikan pemakaian *Unifed Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysist Design* (OOAD).

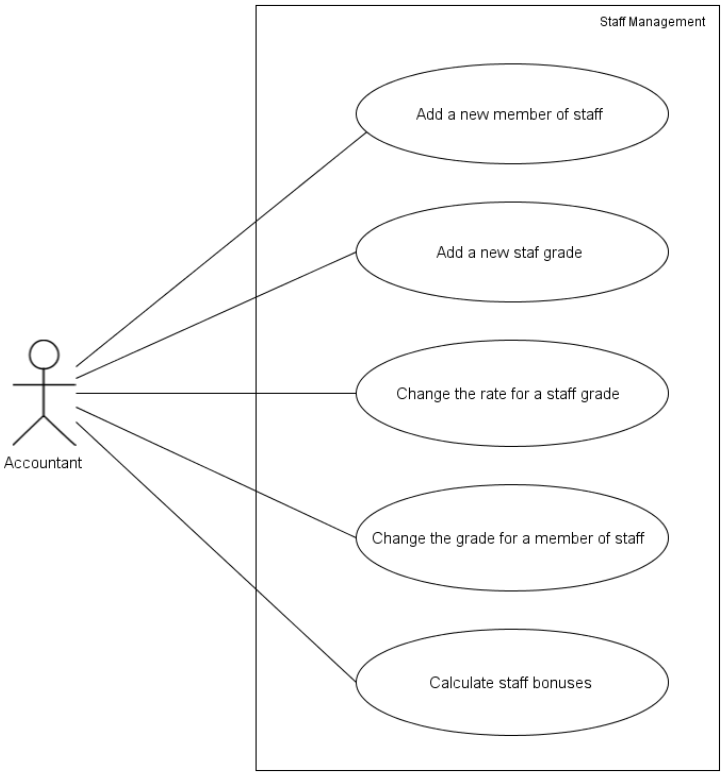
2.9.1 *Unified Modelling Language* (UML)

UML dalam sebuah bahasa untuk menentukan visualisasi, konstruksi, dan mendokumentasikan artifacts dari sistem *software*, untuk memodelkan bisnis, dan sistem *non-software* lainnya. UML merupakan sistem arsitektur yang bekerja dalam OOAD dengan satu bahasa yang konsisten untuk menentukan, visualisasi, konstruksi dan mendokumentasikan *artifact* yang terdapat dalam sistem. *Artifact*

adalah sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*. *Artifact* dapat berupa model, deskripsi atau *software*.

2.9.2 *Use Case Diagram*

Use Case Diagram menjelakan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (Aktor). Diagram ini menunjukan fungsionalitas suatu sistem yang berinteraksi dengan dunia luar. *Use Case Diagram* dapat digunakan selama proses analisis untuk menagkap *requitment* sistem dan untuk memahami bagaimana sistem bekerja.

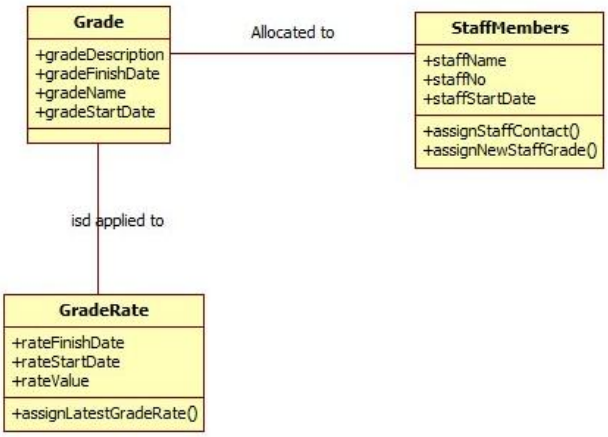


Gambar 2.3 *Use Case Diagram*[10]

2.9.3 *Class Diagram*

Class Diagram menjelaskan dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class Diagram* memperlihatakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain dari suatu sistem. Selama proses analisis, *class diagram*

memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan prilaku sistem. Selama tahap desain, *class diagram* berperan dalam menagkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.



Gambar 2.4 *Class Diagram*[11]

2.9.4 *Behavior Diagram*

Behavior diagram dapat dikelompokan menjadi tiga diagram, yaitu :

a. *Statechart Diagram*

Statechart Diagram berfungsi untuk memodelkan prilaku dinamis satu kelas satu objek.

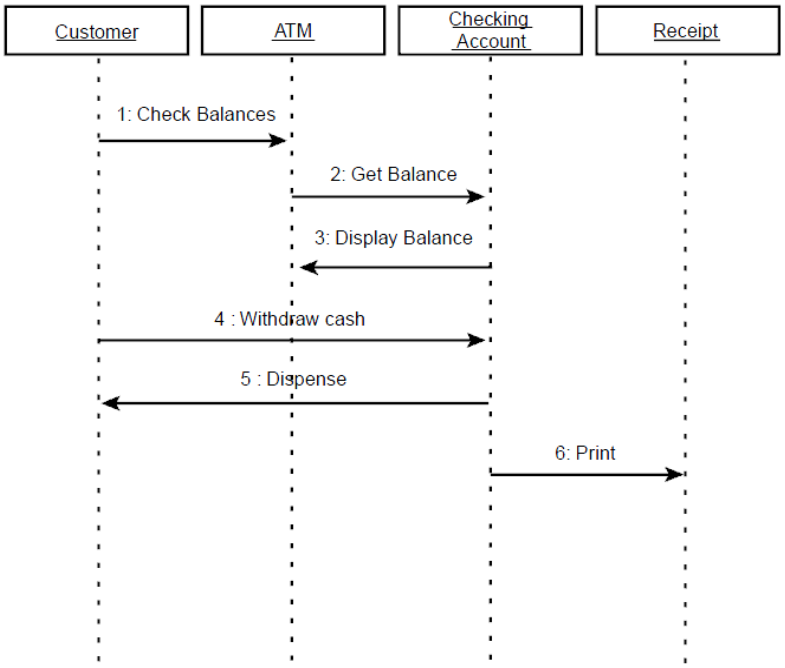
b. *Activity Diagram*

Activity Diagram memodelkan alur kerja (work flow) sebuah proses bisnis dan urutan aktifitas dalam suatu proses.

c. *Interaction Diagram*

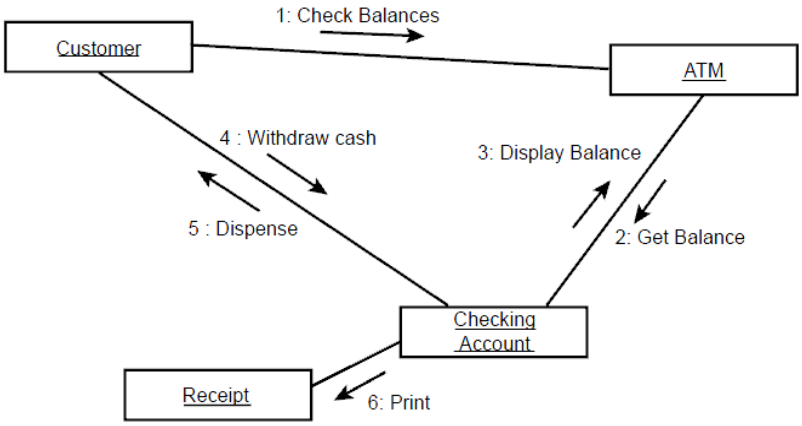
Interaction Diagram dibagi menjadi dua model diagram yaitu :

1. *Sequence Diagram* menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus bersosialisasi dengan *use case*. Sequence diagram, memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu dalam *use case*.



Gambar 2.5 *Sequence Diagram*[12]

2. *Collaboration Diagram* melihat pada interaksi dan hubungan terstruktur antar objek. Tipe diagram ini menekankan pada hubungan (*relationship*) antar objek, sedangkan *sequence diagram* menekankan pada urutan kejadian. Dalam *collaboration diagram* terdapat beberapa objek, *link*, dan *message*.



Gambar 2.6 *Collaboration Diagram*[13]

2.9.5 *Implementatiton Diagram*

Implementation diagram dibagi menjadi dua diagram, yaitu :

1. *Component Diagram* menggambarkan alokasi semua kelas dan objek kedalam komponen-komponen dalam desain fisik sistem *software*. Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software*, seperti *source code*, *binary code*, dan komponen tereksekusi (*execute components*)
2. *Deployment Diagram* memperlihatkan pemetaan *software* kepada *hardware*. Dimana akan berjalan (di *server/multitier*, *standalone* atau lainnya), dan menggambarkan model koneksi dan kemampuan jaringan dan hal lainya yang bersifat fisik.

2.10 Dropbox

Dropbox adalah perangkat lunak yang menghubungkan semua komputer bersama-sama melalui satu folder. Ini adalah cara termudah untuk membuat cadangan dan sinkronisasi file antara komputer. Dropbox terbagi dalam 3 kategori, yaitu :

- Dropbox *Desktop Application* adalah perangkat lunak yang digunakan untuk melihat folder pada komputer desktop dan sinkronisasi perubahan ke web untuk komputer yang lainnya.
- Dropbox *Website* menyediakan akses file di komputer manapun dari *web browser* dan juga dapat digunakan untuk berbagi file atau folder dengan orang lain.
- Dropbox *Mobile Website* dan Dropbox *for Mobile Device* menyediakan koneksi ke Dropbox dari *pocket*, sehingga dapat mengambil *file* yang dibutuhkan ke manapun pergi.

Dropbox didirikan pada tahun 2007 oleh lulusan Massachusetts Institute of Technology (MIT) Drew Houston dan Arash Ferdowsi. Dropbox merupakan *Public Cloud Service* dengan kategori SaaS (*Storage as a Service*), yang mempunyai karakteristik sebagai aplikasi atau layanan informasi tanpa berbayar

untuk menginstal perangkat lunak serta infrastruktur sendiri, fokus pada konsumsi. Fungsinya sebagai arsip, *backup*, *email*, *office*, *payroll* atau *expense*, *file* atau *data storage*, *photo* atau layanan berbagi informasi dan sebagainya[14].

Bila dibandingkan dengan layanan serupa lainnya, Dropbox menawarkan jumlah pengguna yang relatif besar, dengan penggunaan sistem operasi yang bervariasi, baik untuk perangkat *mobile* ataupun *desktop*. Terdapat berbagai versi untuk berbagai sistem operasi, termasuk untuk Microsoft Windows, Mac OS X, dan Linux (resmi atau tidak resmi). Dan tersedia juga berbagai versi untuk perangkat *mobile*, diantaranya Android, Windows Phone 7, iPhone, iPad, Web OS, dan Blackberry, dan klien yang berbasis web. Dropbox menggunakan model finansial *Freemium*, dan layanan gratisnya menyediakan 2 GB penyimpanan online gratis. Para pengguna yang menyarankan Dropbox ke orang lain bisa meningkatkan kapasitas penyimpanan hingga 8 GB. Kompetitor utama dari Dropbox antara lain, Box.net, FilesAnywhere, CloudMe, CrashPlan, Egnyte, iCloud, Mozy, SpiderOak, SugarSync, TitanFile, Ubuntu One, Windows Live SkyDrive, Wuala dan ZumoDrive.

2.10.1 Sejarah Dropbox

Pendiri Dropbox, Drew Houston memperoleh ide mengenai Dropbox setelah *USB Drive* miliknya beberapa kali tertinggal ketika ia menjadi murid di MIT. Ia mengatakan bahwa layanan yang tersedia saat itu bermasalah dengan penggunaan *internet* yang tidak dimaksimalkan, ukuran data yang besar, banyaknya *error* dan kesalahan, serta tidak praktis dan membingungkan banyak orang. Ia lalu mulai menciptakan sesuatu untuk dirinya sendiri, namun lalu menyadari bahwa hal ini dapat menguntungkan orang lain yang memiliki masalah yang sama. Houston lalu mendirikan Dropbox, Inc. pada tahun 2007, dan tidak lama setelah itu mendapatkan dana awal dari Y Combinator. Dropbox secara resmi diluncurkan pada acara TechCrunch50, sebuah konferensi teknologi tahunan pada tahun 2008.

2.10.2 Teknologi

Baik *server* atau *desktop client* Dropbox, keduanya ditulis dengan Python. Dropbox menggunakan sistem penyimpanan Amazon's S3 untuk menyimpan data; walaupun Houston telah menyatakan bahwa Dropbox mungkin akan mengubah ke penyedia jasa penyimpanan lainnya suatu saat nanti. Dropbox juga menggunakan transfer SSL untuk sinkronisasi dan menyimpan data lewat enkripsi AES-256.

Klien Dropbox memungkinkan para pengguna untuk meletakkan data apapun menjadi sebuah berkas, yang kemudian dapat dihubungkan dengan layanan *internet* Dropbox dan ke komputer dan perangkat yang dimiliki pengguna-pengguna lainnya, yang juga memiliki klien Dropbox. Pengguna juga bisa mengunggah data secara manual lewat *web browser*. Dropbox dapat menjadi alternatif dari *sneakernet* (transportasi fisik lewat media yang dapat dipindahkan), dan bentuk tradisional lainnya dari transfer data, seperti FTP atau lampiran *email*.

2.11 Eclipse

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

1. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman

perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya popular adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

2.11.1 Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM *Visual Age for Java* 4.0. Produk ini diluncurkan oleh IBM pada tahun 2001. Pada tahun 2003 platform Eclipse tumbuh dengan cepat sebagai aplikasi *open source* dan *commercial extensions,* sehingga menjadi pengembang yang terkenal. Pada tahun 2004 Eclipse *Foundation* yang merupakan sebuah organisasi nonprofit dengan keprofesionalannya serta *independent* dapat mengambil alih penuh platform Eclipse. Eclipse adalah pengembang java terkemuka, karena unik dan arsitektur yang dapat diperluas serta dapat digunakan untuk berbagai macam bahasa pemrograman[15].

2.11.2 Arsitektur Eclipse

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

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

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk

mengembangkan *plug-in* baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

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

2.11.3 Perkembangan Eclipse

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

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

2.12 *Software Development Kit* (*SDK*)

Software Development Kit (SDK) adalah suatu kit atau library dari bahasa pemrograman untuk pengembangan atau pembangunan suatu perangkat lunak dan biasanya SDK terdiri dari kumpulan tools yang dibutuhkan. Misalnya bahasa pemrograman java, mempunyai SDK yang berisi suatu *library* yang dapat digunakan untuk membuat suatu aplikasi berbasis java.

2.13 *Java Development Kit* (*JDK*)

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

2.14 *Android Development Tools (ADT)*

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

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

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

2.15 Skala Likert (*Likert Scales*)

Skala likert adalah suatu skala psikometrik yang digunakan dalam kuesioner dan merupakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Rensis Likert telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932 yang sekarang terkenal dengan nama skala Likert[16]. Skala Likert ini merupakan skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena. Dan pada evaluasi, skala likert digunakan untuk (a) Menilai keberhasilan suatu kebijakan atau program (b) Menilai manfaat pelaksanaan suatu kebijakan atau program (c) Mengetahui kepuasan stakeholder terhadap pelaksanaan suatu kebijakan atau program.

Dalam skala Likert terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif, dan pernyataan negatif yang berfungsi untuk mengukur sikap negatif objek. Skor pernyataan positif dimulai dari 1 untuk sangat tidak setuju (ST), 2 untuk tidak setuju (T), 3 untuk tak memutuskan (N), 4 untuk setuju (S), dan 5 untuk sangat setuju (SS). Skor pernyataan negatif dimulai dari 1 untuk sangat setuju (SS), 2 untuk setuju (S), 3 untuk tak memutuskan (N), 4 untuk tidak setuju (T), dan 5 untuk sangat tidak setuju (ST). Skala Likert digunakan untuk mengukur **kesetujuan dan ketidaksetujuan** seseorang terhadap sesuatu rencana program, pelaksanaan program ataupun tingkat keberhasilan suatu program.

Tabel 2.1 Bobot Penilaian Skala Likert

Arah Pertanyaan	Bobot Penilaian				
	Sangat Setuju (SS)	Setuju (S)	Tak Memutuskan (N)	Tidak Setuju (T)	Sangat Tidak Setuju (ST)
Positif	5	4	3	2	1
Negatif	1	2	3	4	5

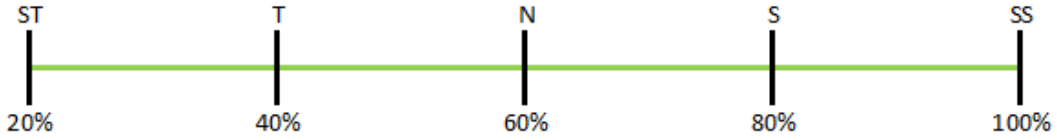
Berdasarkan data hasil kuesioner tersebut, dapat dicari persentase masing-masing jawaban dengan menggunakan rumus :

$$Y = \frac{X}{Skor\,ideal}\ x\ 100\%$$

Keterangan :

Y	=	Nilai persentase yang dicari
X	=	Jumlah nilai kategori jawaban dikalikan dengan frekuensi ($\sum$ = N . R)
N	=	Nilai dari setiap jawaban
R	=	Frekuensi
Skor ideal	=	Nilai tertinggi dikalikan dengan jumlah sampel

Setelah diperoleh nilai Y maka interpretasi nilai dapat dilihat sebagai berikut :



Gambar 2.7 Skala Penilaian Kuesioner

Setelah didapatkan jumlah yang pasti dari setiap kuesioner, selanjutnya penulis menarik kesimpulan dari masing-masing kuesioner yang disediakan.

2.15.1 Kelebihan Skala Likert

Beberapa Kelebihan Skala Likert dalam riset berupa survei, antara lain :

1. Mudah dibuat dan diterapkan.
2. Skala Likert lebih mudah membuatnya dibandingkan dengan skala yang lain seperti skala Thurstone.
3. Terdapat kebebasan dalam memasukan pertanyaan- pertanyaan, asalkan sesuai dengan konteks permasalahan yang diteliti.
4. Jawaban suatu item dapat berupa alternatif, sehingga informasi mengenai item tersebut diperjelas.
5. Reliabilitas pengukuran bisa diperoleh dengan jumlah item tersebut diperjelas.

6. Karena jangka responsi yang lebih besar membuat skala Likert dapat memberikan keterangan yang lebih jelas dan nyata tentang pendapatan atau sikap responden tentang isu yang dipertanyakan.

2.15.2 Kekurangan Skala Likert

Beberapa Kekurangan Skala Likert dalam riset berupa survei, antara lain :

1. Karena ukuran yang digunakan adalah ukuran ordinal, skala Likert hanya dapat mengurutkan individu dalam skala, tetapi tidak dapat membandingkan berapa kali satu individu lebih baik dari individu yang lain.
2. Kadangkala total skor dari individu tidak memberikan arti yang jelas, karena banyak pola respons terhadap beberapa item akan memberikan skor yang sama. Adanya kelemahan di atas sebenarnya dapat dipikirkan sebagai *error* dari respons yang terjadi.