BAB 2
LANDASAN TEORI

2.1 Profil Victory Eka Trinina

Berikut ini adalah profil dari lembaga kursus bahasa inggris Victory Eka Tribina yang merupakan tempat penelitian untuk membangun Aplikasi Tutorial Bahasa Inggris Berbabsis *mobile*

2.1.1 Sejarah

Lembaga Pendidikan ini sudah berdiri dan beroperasi di Ujungbatu semenjak 31 Juli l993. Tepatnya pada 9 Agustus l993, pihak Kandepdikbud Kecamatan memberikan izin operasional dan kemudian pihak Kanwil Depdikbud Propinsi Riau mengeluarkan izin penyelenggaraan sepenuhnya pada 30 Oktober 1993. Lembaga ini pada mulanya bernama Lembaga Pendidikan Bahasa Inggris "VICTORY ENGLISH TRAINING" yang disingkat dengan "VET". Kemudian pada awal tahun 1996 berganti nama menjadi Lembaga Pendidikan Bahasa Inggris "VICTORY EKA TRIBINA" yang juga masih disingkat "VET".

Keberadaan dan perkembangan Lembaga Pendidikan Bahasa Inggris VET tentunya bukanlah hanya cerita tentang kesuksesan semata atau serta merta seperti apa yang kita saksikan saat sekarang ini. Selama bertahun-tahun dalam perjalanannya VET telah sering mengalami pasang surut dan malang melintang dalam kiprahnya menyuguhkan dan mewujudkan niat yang tulus, suci dan ikhlas dibidang pendidikan ditengah-tengah masyarakat. Sungguhpun demikian adanya, kami dari pihak VET tetap tegar dan optimis serta berpikiran positif menatap masa depan. Berawal dari jumlah warga belajar yang sedikit dibanding saat ini dan dengan suasana yang serba sederhana dan terbatas, baik dibidang sarana dan prasarana belajar mengajar. Namun keberadaan lembaga ini akhirnya mendapat tempat dihati sebahagian besar masyarakat Ujungbatu dan sekitarnya, maka secara perlahan namun pasti VET terus tumbuh membaik dari tahun ketahun.

Berdasarkan pengalaman dan pantauan kami, tumbuh dan tetap bertahannya Lembaga Pendidikan Bahasa Inggris Victory Eka Tribina ini sangat dipengaruhi oleh variable-variable berikut:

1. Kepercayaan dan dukungan masyarakat.
2. Kekuatan ekonomi masyarakat.
3. Kemitraan dengan beberapa sekolah.
4. Dukungan pemerintah dan instansi yang terkait.

Dengan dilandasi niat tulus, suci dan ikhlas dibidang pendidikan, diharapkan VET senantiasa mendapat tempat utama dihati masyarakat sampai masa-masa yang akan datang.

2.1.2 Eksistensi (Keberadaan) Victory Eka tribina

1. Nama Lembaga : Lembaga Pendidikan Bahasa Inggris VICTORY EKA TRIBINA (VET)
2. Alamat Lembaga

Jalan	: Kutilang
RT/RW	: 01/ 05
Kelurahan	: Ujungbatu
Kecamatan	: Ujungbatu
Kabupaten	: Rokan Hulu
Propinsi	: Riau
Kode Pos	: 28454
No. Telp.	: 0762-62229

2. Pemimpin/Pemilik/Penyelenggara : Drs. ARIFIN S
3. Tahun Berdiri Lembaga : 31 Juli 1993
4. NILEK Lama : 09109.3.1.0014/09
5. NILEK Nasional : 09109.1.0014
6. No. SK Izin Dinas Pendidikan Tahap : “A”
7. No. Rek. Bank Lembaga : BRI Simpedes 3363.0100.4815539.
8. No. NPWP : 02.850.094.0-221.000
9. No. Akta Notaris : 19, Tgl. 25 Nopember 2010

10. Pejabat Notaris : ENI ENDAHWATI, SH.

2.1.3 Nama dan Logo Victory Eka Tribina

a. Nama lengkap adalah "Lembaga Pendidikan Bahasa Inggris VICTORY EKA TRIBINA"
b. Makna yang terkandung pada VICTORY EKA TRIBINA yaitu: *"Kemenangan tunggal dalam tiga pembinaan"*
    1. VICTORY : Kemenangan
    2. EKA : Tunggal
    3. TRIBINA :Tiga pembinaan (moral, wawasan dan pengetahuan bahasa Inggris)
c. Dalam pemakaian nama sehari-hari dipergunakan nama lebih singkat yaitu VET.

2.1.4 Logo



Gambar 2.1 Logo Victory Eka Tribina

a. Warna logo dan maknanya:

1. Biru/dongker : Bijaksana, arif
2. Merah : Berani
3. Putih : Jujur, tulus, Suci

b. Spesifikasi dan makna logo VET:

1. Buku terkembang : Pengetahuan bahasa Inggris
2. Perahu layar : Bahtera, sarana
3. Keris mengacung ke atas : Kekuatan
4. Air : Hidup dankehidupan
5. Segi lima : Bingkai kebersamaan/kekeluargaan
6. Victory Eka Tribina – VET : Nama Lembaga dan singkatannya

2.1.5 Visi dan Misi Victory Eka Tribina

2.1.5.1 Visi

Visi VET adalah menjadi pusat lembaga pendidikan dan keterampilan terbaik, berkwalitas juga terfavorit di Ujungbatu pada khususnya dan di Kabupaten Rokan Hulu pada umumnya, terutama pada bidang pendidikan yang disajikan yaitu pendidikan bahasa Inggris.

2.1.5.2 Misi

a. Menyajikan pendidikan dan keterampilan yang bermakna untuk memenuhi kebutuhan tiga dimensi, yaitu intelektual, emosional dan spiritual.
b. Menciptakan insan-insan yang memiliki kecakapan hidup dan sikap yang mandiri untuk memanfaatkan potensi diri yang pada akhirnya bermuara pada ketajaman pandang untuk memanfaatkan prospek dan potensi yang ada pada lingkungan dimana mereka berada.

2.1.6 Tujuan dan Fungsi

2.1.6.1 Tujuan

Adapun tujan penyelenggaraan lembaga pendidikan ini, yaitu:

a. Untuk ikut serta menyelenggarakan pendidikan Nasional pada umumnya dan pendidikan di daerah operasi VICTORY EKA TRIBINA (VET) pada khususnya.
b. Untuk mempersiapkan masyarakat dalam menghadapi era globalisasiinformasi, saat ini.

c. Untuk meningkatkan sumber daya manusia di daerah pada umumnya dan membantu para pelajar serta pencari kerja pada khususnya dengan keterampilan kecakapan hidup.

2.1.6.2 Fungsi

Kegunaan penyelenggaraan lembaga pendidikan ini, yaitu:

a. Menambah akses pendidikan yang telah ada di daerah dan merupakan alternatif bagi generasi muda dalam membekali diri untuk lebih kompetitif dibidang sumber daya manusia.
b. Membantu para pelajar agar lebih mudah memahami dan menguasai bahasa Inggris dan sekaligus memberikan STTB-PLUS yang dapat mendampingi Ijazah/STTB yang telah dan akan mereka miliki.
c. Meningkatkan kualitas dan daya saing masyarakat pada umumnya dan para pencari kerja pada khususnya, yakni dibidang kemampuan berbahasa Inggris yang sangat diperlukan sekali diberbagai bidang dan aspek pekerjaaan serta kehidupan manusia dizaman sekarang ini.

2.1.7 Kepengurusan

1. Pimpinan/ Penyelenggara Victory Eka Tribina

Nama	: Drs. Arifin S
Tempat lahir	: Langga Payung
Tanggal lahir	: 15 Maret 1967
Jenis kelamin	: Laki-laki
Kebangsaan/Agama	: Indonesia/Islam
Pendidikan terakhir	: Sarjana (S.1)
Alamat	: Jalan Kutilang Ujungbatu-Rokan Hulu

2. Ketua

Nama	: Afrizal, SS
Tempat lahir	: Pekanbaru
Tanggal lahir	: 16 April 1978
Jenis kelamin	: Laki-laki
Kebangsaan/Agama	: Indonesia/Islam
Pendidikan terakhir	: Sarjana (S.1)
Alamat	: Petakur Bawah (Sukadamai) Ujungbatu

3. Bendahara

Nama	: Dra. Nelwiza Darwis
Tempat lahir	: Padang Panjang
Tanggal lahir	: 03 September 1967
Jenis kelamin	: Perempuan
Kebangsaan/Agama	: Indonesia/Islam
Pendidikan terakhir	: Sarjana (S.1)
Alamat	: Jalan Kutilang Ujungbatu-RokanHulu

4. Sekretaris

Nama	: Qurrata Aini
Tempat lahir	: Ujungbatu
Tanggal lahir	: 09 Maret 1995
Jenis kelamin	: Perempuan
Kebangsaan/Agama	: Indonesia/Islam
Pendidikan terakhir	: SLTA

Alamat : Jalan Kutilang Ujungbatu-Rokan Hulu

5. Pengawas

Nama	: Runi Erizal, S.Pd.SD
Tempat lahir	: P. Brandan
Tanggal lahir	: 23 Pebruari 1983
Jenis kelamin	: Laki-laki
Kebangsaan/Agama	: Indonesia/Islam
Pendidikan terakhir	: Sarjana (S.1)
Alamat	: Jalan Ngaso Ujungbatu-Rokan Hulu

2.1.8 Layanan Pendidikan

A. BASIC LEVEL

Basic 1

Untuk : SD/MI Kelas I (Lama Pendidikan 1 thn)

Basic 2

Untuk : SD/MI Kelas II (Lama Pendidikan1thn)

Basic 3

Untuk : SD/MI Kelas III (Lama Pendidikan 1 thn)

B. PRIMARY LEVEL

Primary 1

Untuk : SD kelas IV dan V (Lama Pendidikan 1 thn).

Primary 2

Untuk : SD kelas VI dan SLTP kelas I atau tamatan primary (Lama Pendidikan 1 thn).

C.ELEMENTARY LEVEL

Untuk : SLTP kelas II & III, SLTA dan Umum atau tamatan Primary 2 (Lama Pendidikan 8 bln).

D. INTERMEDIATE LEVEL

Untuk : Pelajar & Umum tamatan Elementary level. (Lama Pendidikan 10 bln).

E. ADVANCED LEVEL

Untuk : Pelajar & Umum tamatan Intermediate level. (Lama Pendidikan 12 bln).

F. EXPRESS ENGLISH (Bahasa Inggris Cepat)

Untuk : Umum & pencari kerja, pendidikan terakhir minimal SLTA. (Lama Pendidikan 3 bln).

2.2 Landasan Teori

Teori pendukung atau landasan teori dalam penyusunan skripsi ini sangat diperlukan karena sebagai referensi untuk menunjang atau memperdalam pemahaman terhadap informasi-informasi yang disajikan.

2.2.1 Multimedia dalam Pembelajaran

Multimedia mencakup berbagai media yang terintegrasi menjadi satu. Setiap komponen media dapat merangsang satu atau lebih indra manusia. Teori Koehnert mengatakan bahwa semakin banyak indra yang terlibat dalam proses belajar, maka proses belajar tersebut akan menjadi lebih efektif. Secara tegas teori ini menyarankan penggunaan lebih dari satu indera manusia. Oleh karena itu, pemanfaatan *multimedia* dalam pembelajaran dapat diharapkan meningkatkan hasil belajar.[6]

Pernyataan di atas berkaitan dengan pendapat Dale yang menyatakan bahwa pemerolehan hasil belajar melalui indra pandang berkisar 75%, melalui indra dengar sekitar 13%, dan melalui indra lainnya sekitar 12%. Hal senada ditegaskan oleh Baugh yang menyatakan bahwa kurang lebih 90% hasil belajar seseorang diperoleh melalui indra pandang, 5% diperoleh melalui indra dengar, dan 5% lagi diperoleh melalui indra lainnya.[7]

Selain itu multimedia juga fleksibel dalam menyesuaikan dengan kecepatan belajar seseorang. Seorang pebelajar yang memiliki kecepatan belajar lebih tinggi dapat lebih cepat menyelesaikan kegiatan belajarnya, sedangkan pebelajar dengan kecepatan belajar lambat dapat menyelesaikan aktivitas

belajarnya sesuai dengan kecepatannya masing-masing. Hal ini sangat berbeda dengan pembelajaran di kelas konvensional, dimana setiap pebelajar dipaksa belajar dengan kecepatan yang ditentukan oleh guru. Pebelajar yang mempunyai kecepatan belajar tinggi dapat merasa bosan, sebaliknya pebelajar dengan kecepatan belajar rendah merasa pembelajaran terlalu cepat untuk diikuti.[6]

2.3 Pengertian Multimedia

Multimedia adalah suatu kombinasi dari berbagai medium, dimana kombinasi tersebut dapat digunakan untuk tujuan pembelajaran. *Multimedia* juga dapat diartikan sebagai gabungan dari teks, suara, gambar, animasi dan video dengan alat bantu (*tool*) dan koneksi (*link*) sehingga pengguna dapat bernavigasi, berinteraksi, berkarya dan berkomunikasi.[5]

Multimedia berasal dari kata multi dan media. Multi berarti banyak, dan media berarti tempat, sarana atau alat yang digunakan untuk menyimpan informasi. Jadi berdasarkan kata "multimedia" dapat diasumsikan sebagai wadah atau penyatuan beberapa media yang kemudian didefinisikan sebagai elemen-elemen pembentukan *multimedia*. Elemen-elemen tersebut berupa teks, gambar, suara, animasi, dan video. *Multimedia* merupakan suatu konsep dan teknologi baru bidang bidang teknologi informasi, dimana informasi dalam bentuk teks, gambar, suara, animasi, dan video disatukan dalam komputer untuk disimpan, diproses, dan disajikan baik secara linier maupun interaktif.

Menurut Arsyad *multimedia* adalah berbagai macam kombinasi grafik, teks, audio, suara, dan animasi. Penggabungan ini merupakan suatu kesatuan yang secara bersama-sama menampilkan informasi, pesan, atau isi pembelajaran.[3]

Sedangkan Gayeski mengartikan *multimedia* ialah suatu sistem hubungan komunikasi interaktif melalui komputer yang mampu menciptakan, menyimpan, memindahkan, dan mencapai kembali data dan maklumat dalam bentuk teks, grafik, animasi, dan sistem audio.[4]

Penyajian dengan menggabungkan seluruh elemen *multimedia* tersebut menjadikan informasi dalam bentuk *multimedia* yang dapat diterima oleh indera penglihatan dan pendengaran, lebih mendekati bentuk aslinya dalam dunia

sebenarnya. *Multimedia* interaktif adalah bila suatu aplikasi terdapat seluruh elemen *multimedia* yang ada dan pemakai (*user*) diberi kebebasan/kemampuan untuk mengontrol dan menghidupkan elemen-elemen tersebut.

2.3.1 Komponen Multimedia

Multimedia terdiri dari beberapa komponen [5], yaitu:

1. Teks

   Teks merupakan dasar dari pengolahan kata dan informasi berbasis *multimedia*.

2. Grafik

   Secara umum grafik berarti gambar garis (*line drawing*). Manusia sangat berorientasi pada *visual*, dan gambar merupakan sarana yang sangat baik untuk menyajikan informasi.

3. *Images*

   Secara umum *images* berarti gambar raster (*halftone drawing*), seperti foto.

4. Animasi

   Animasi menggambarkan objek yang bergerak agar kelihatan hidup.

5. Suara

   Suara dapat lebih memperjelas pengertian yang ditampilkan dengan cara lain.

2.4 Pengertian Android

Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware* dan aplikasi inti yang diterbitkan oleh Google. Sedangkan Android SDK (*Software Development Kit*) menyediakan *Tools* dan API yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java.

Dikembangkan bersama antara Google, HTC, Intel, Motorola, Qualcomm, TMobile, NVIDIA yang tergabung dalam OHA (Open Handset Alliance) dengantujuan membuat sebuah standar terbuka untuk perangkat bergerak (*mobile device*).[1]

2.4.1 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.

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.[10]

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-speechengine*. Kemampuan *dial* kontak teknologi *texttochangespeech* (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 (*kille rapps* - 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 *Weather Bug*. 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. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan *Adobe Flash* 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 Java Script *engine* yang dipakai *Google Chrome* yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD Card, kemampuan *WiFi Hotspot portabel*, dan kemampuan *autoupdate* 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 *copypaste*, layar antar muka (*UserInterface*) didesain ulang, dukungan *form*at video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *head phone virtualization*, dan *bassboost*), 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. *UserInterface* 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. *Sourcecode* 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*)

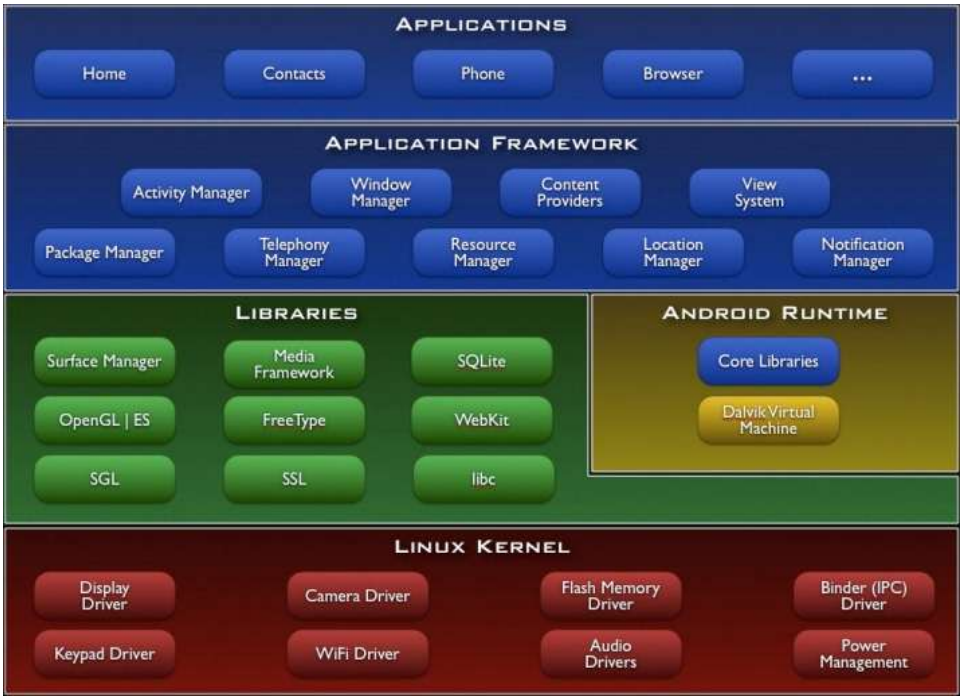
Android ini diluncurkan pada acara Google *I/O* lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan *input keyboard*, desain baru fitur pencarian, UI yang baru dan pencarian melalui *voicesearch* yang lebih tepat.

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.

2.4.2 Anatomi Aplikasi Android

Dalam paket sistem operasi Android tediri dari beberapa unsur seperti tampak pada gambar 2.7. Secara sederhana arsitektur Android merupakan sebuah

kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *Framework* yang menyediakan dan mengatur alur proses aplikasi. [Google IO, Android Anatomy and Physiology] [11]



Gambar 2.2*Detail* Anatomi *Android*[11].

2.4.2.1 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 servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, *keypad*, *WiFi*, *Flash Memory*, *audio*, dan IPC (*Inter process Communication*) untuk mengatur aplikasi dan lubang keamanan.

2.4.2.2 *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 *form*at 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 Tool kit* (NDK).

2.4.2.2.1 ActionBarSherlock

ActionBarSherlock digunakan untuk mengatur dengan mudah *design panel* pada tampilan aplikasi.[13]

2.4.2.2.1.1 SherlockActivity

Salah satu bagian dari *class* dari *library ActionBarSherlock* yang digunakan untuk menyesuaikan tampilan aplikasi pada sistem operasi dari *gingerbread* keatas.[13].

Berikut method-method yang terdapat pada *class* SherlockActivity

1. getSherlock
2. getSupportActionBar
3. startActionMode
4. onActionModeStarted
5. onActionModeFinished

6. onConfigurationChanged
7. onPostResume
8. onPause
9. onStop
10. onDestroy
11. onPostCreate
12. onTitleChanged
13. onMenuOpened
14. onPanelClosed
15. dispatchKeyEvent
16. getSupportMenuInflater
17. invalidateOptionsMenu
18. supportInvalidateOptionsMenu
19. onCreateOptionsMenu
20. setSupportSecondaryProgress
21. setSupportProgressBarVisibility

2.4.2.2.2 WebViewClient

WebViewClient Adalah Sebuah *library* yang dirancang untuk menampilkan halaman tampilan yang berbentuk web.[14]

Berikut method-method yang terdapat pada class WebViewClient

1. shouldOverrideUrlLoading

Untuk memberikan sejumlah besar sebuah aplikasi untuk melakukan perubahan untuk menangani kunci secara *synchronous.*

2. onReceivedError

Berguna untuk melaporkan sejumlah kesalahan pada aplikasi.

3. onPageStarted

Berguna untuk menandakan sejumlah aplikasi yang terdapat didalam halaman yang akan dimulai.

4. onPageFinished

Berguna untuk menandakan sejumlah aplikasi yang terdapat dalam halaman yang akan selesai.

2.4.2.2.3 Activity

Salah satu *class* yang terdapat pada *library* sangat penting untuk mengatur alur hidup pada siklus android.

Berikut method yang terdapat pada class Activity

1. onCreate()
2. onRestoreInstanceState()
3. onPostCreate()
4. onStart()
5. onRestart()
6. onResume()
7. onPostResume()
8. onNewIntent()
9. onSaveInstanceState()
10. onPause()
11. onUserLeaveHint()
12. onStop()
13. onDestroy()
14. onDestroy()

2.4.2.2.4 SQLiteOpenHelper

SQLiteOpenHelper adalah suatu *class* yang membantu mengatur dalam menciptakan *database* dan mengatur versi dari *database.*

Berikut method-method yang terdapat pada class SQLiteOpenHelper :

1. Close ()

Berfungsi untuk menutup objek *database.*

2. getDatabaseName()

Berfungsi untuk mengembalikan nama *database* dari sqlite yang sedang dibuka, dan memberikan kepada konstruktor.

3. getReadableDatabase()

Berfungsi untuk membuat dan atau membuka *database.*

4. getWritableDatabase()

Berfungsi untuk menciptakan atau menggunakan database yang akan digunakan untuk dibaca atau ditulis.

5. onConfigure(SQLiteDatabase db)

Berfungsi untuk memanggil ketika *database* sudah terkoneksi dan terkonfigurasi dengan benar, untuk menjalankan fitur ini seperti menulis di log pada kunci utama.

6. onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion)

Digunakan ketika *database* membutuhkan untuk diturunkan versinya.

7. onOpen(SQLiteDatabase db)

Digunakan untuk membuka *database* yang telah dibuka.

8. onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)

Digunakan ketika database dibutuhkan untuk dinaikkan kelas versinya.

9. setWriteAheadLoggingEnabled(boolean enabled)

Digunakan untuk menjalankan atau tidak menjalankan menggunakan log pada *database.*

2.4.2.2.5 BaseAdapter

Digunakan untuk umumnya dari kelas dasar dari implementasi dasar pada sebuah adapter yang bisa digunakan pada ListView.

Berikut method-method yang terdapat pada class BaseAdapter

1. areAllItemsEnabled()

Berfungsi untuk memastikan apakah seluruh item pada adapter ini telah berjalan.

2. getDropDownView(int position, View convertView, ViewGroup parent)

Berfungsi untuk mendapatkan *view* yang ditampilkan pada drop down pop up pada posisi yang spesifik dalam sekumpulan data.

3. getItemViewType(int position)

Untuk mendapatkan item dari view yang akan diciptkan pada spesifik item.

4. getViewTypeCount()

Untuk mengembalikan nomor dari tipe view yang akan diciptakan

5. hasStableIds()

Untuk mengindikasikan apakah item ids telah stabil pada perubahan ke data utama.

6. isEmpty()

Untuk mengindikasikan data yang kosong.

7. isEnabled(int position)

Untuk mengembalikan nilai benar jika item pada posisi spesifik yang tidak terpisah.

8. notifyDataSetChanged()

Untuk memberitahukan pemasangan pengamatan pada data utama yang telah dirubah dan setiap refleksi dari tampilan akan di diatur ulang secara otomatis.

9. notifyDataSetInvalidated()

Untuk memberitahukan pemasangan pengamatan data pokok sepanjang data benar dan tersedia.

10. registerDataSetObserver(DataSetObserver observer)

Digunakan jika data mengalami perubahan data yang digunakan oleh adapter.

11. unregisterDataSetObserver(DataSetObserver observer)

Unregister an observer digunakan untuk mengulangi *registerd* dengan via adapter ini

2.4.2.2.6 SimpleCursorAdapter

Sebuah adapter yang mudah untuk mengatur kolom dari kursor yang berbentuk tampilan teks atau tampilan gambar yang ditemui pada sebuah *file* XML.

Berikut method-method yang terdapat pada SimpleCursorAdapter :

1. bindView(View view, Context context, Cursor cursor)

   Mengikat seluru nama *field* kedalam parameter dari konstruktor dengan kolom kursor yang sesuai dari parameter tersebut.

2. changeCursorAndColumns(Cursor c, String[] from, int[] to)

   Untuk merubah kursor san merubah tampilan kolom dalam waktu yang sama.

3. convertToString(Cursor cursor)

   Mengembalikan dari representasi CharSequence dari kursor yang spesifik, sebagai pendefinisian dari CursorToStringConverter.

4. getCursorToStringConverter()

   Mengembalikan konverter untuk mengubah penyaringan kursor kepada string.

5. getStringConversionColumn()

   Mengembalikan indeks dari kolom yang digunakan untuk mendapatkan representasi dari string dari kursor.

6. getViewBinder()

   Mengembalikan SimpleCursorAdapter.ViewBinder untuk ditampilkan.

7. setCursorToStringConverter(SimpleCursorAdapter.CursorToStringConverter cursorToStringConverter)

   Meletakkan converter yang digunakan untuk mengubah penyaringan cursor kepada string.

8. setStringConversionColumn(int stringConversionColumn)

   Menjelaskan indeks dari kolom didalam kursor yang digunakan untuk mendapatkan representasi dari string dari kursor.

9. setViewBinder(SimpleCursorAdapter.ViewBinder viewBinder)

Meletakkan pengikat yang digunakan untuk mengikat data untuk ditampilkan

10. setViewImage(ImageView v, String value)

Memanggil bindView() untuk meletakkan gambar dari sebuah *ImageView* tetapi hanya tidak ada *ViewBinder* atau jika keberadaan *ViewBinder* tidak bisa menangani menampilkan sebuah *ImageView.*

11. setViewText(TextView v, String text)

Memanggil sebuah *bindView()* kepada sejumlah teks untuk sebuah *TextView* tetapi jika keberadaan dari ViewBinder tidak atau ViewBinder ada namun tidak bisa menangani TextView.

12. swapCursor(Cursor c)

Menukar kursor baru, dan mengembalikan kursor lama.

2.4.2.3 *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. AndroidRuntime 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 *form*at *DalvikExecutable* (*.dex). Dengan *form*at 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.

2.4.2.4 *Aplication 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 *UserInterface* dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi *Android* adalah sebagai berikut:

1. *ActivityManager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan ”*Backstack*“ untuk navigasi penggunaan.
2. *ContentProviders*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.
3. *ResuourceManager*, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan *filelayout*.
4. *LocationManager*, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat Android berada.
5. *NotificationManager*, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada *statusbar*.

2.4.2.5 Aplication 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 *Frame work* 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.

2.4.3 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.

2.4.3.1 *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 *activitystack*. 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. *Inactiveactivity* 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 *pause dactivity* juga akan ditutup.

2.4.3.2 *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.

2.4.3.3 *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.

2.4.3.4 *Broadcast Receiver*

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

2.4.3.5 *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.

2.4.4 Tipe Aplikasi Android

Terdapat tiga kategori aplikasi pada Android :

1. *ForegroundActivity*

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. *BackgroundService*

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 *autorespon*.

3. *IntermittentActivity*

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.

2.4.5 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 *opensource*. 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 *virtualDalvik*, sehingga kode program *portabel* 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.

2.5 Pengertian Basis Data

Basis data terdiri atas 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul.

Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bumi, atau kombinasinya.[12]

Struktur File *Database* :

1. *Data* adalah satu satuan informasi yang akan diolah, dimana sebelum diolah dikumpulkan di dalam suatu file *database*. Pengumpulan data dilakukan secara sistematis menurut struktur file database tersebut.
2. *RECORD* adalah data yang isinya merupakan satu kesatuan seperti Nama, Alamat, Nomor Telepon. Setiap keterangan yang mencakup nama, alamat dan nomor telepon dinamakan satu record. Dan setiap *record* diberi nomor urut yang disebut nomor *record* (*RecordNumber*). Ukuran suatu *file database* ditentukan oleh jumlah *record* yang tersimpan di dalamnya.
3. *FIELD* adalah sub bagian dari *record*. Dari contoh isi *record* diatas maka terdiri dari 3 *field*, yaitu *field* Nama, *field* Alamat dan *field* Nomor Telepon.

2.6 Pengertian Java

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, *desktop*, web dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform*sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.Sebagai sebuah peralatan pembangun, teknologi Java menyediakan banyak *tools* : *compiler,interpreter*, penyusun dokumentasi, paket kelas dan sebagainya.

Aplikasi dengan teknologi Java secara umum adalah aplikasi serba guna yang dapat dijalankan pada seluruh mesin yang memiliki *Java Runtime Environment* (JRE).Terdapat dua komponen utama dari *Deployment Environment*. Yang pertama adalah JRE, yang terdapat pada paket J2SDK, mengandung kelas–kelas untuk semua paket teknologi Java yang meliputi kelas dasar dari Java, komponen GUI dan sebagainya. Komponen yang lain terdapat pada *Web*

Browser. Hampir seluruh *Web Browser* komersial menyediakan *interpreter* dan *runtime environment* dari teknologi Java

2.7 Pengertian UML

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. *Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an.

Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualiasikan, membangun, dan mendokumentasikan suatu sistem informasi.

UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasiobjek 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 *system*.[8]

UML menyediakan 9 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu :

1. *Use Case* Diagram untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects.*
4. *Collaboration Diagram* untuk memodelkan interaksi antar *objects*.
5. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.

6. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan *objects* dalam *system*
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur *object.*
9. *Component Diagram* untuk memodelkan komponen *object*

2.8 Pengertian Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*)[10]. 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 dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in. Eclipse dibuat dari kerja sama antara perusahaan-perusahaan anggota 'Eclipse *Foundation*' (beserta individu-individu lain). Banyak nama besar yang ikut dalam 'Eclipse.

Foundation', termasuk IBM, BEA, Intel, Nokia, Borland. Eclipse bersaing langsung dengan Netbeans IDE. Plugin tambahan pada Eclipse jauh lebih banyak dan bervariasi dibandingkan IDE lainnya.

1. IntelliJ IDEA (commercial, free 30-day trial)

2. Oracle JDeveloper (free)
3. Xinox JCreator (ada versi berbayar maupun free)

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US$ 40 juta untuk pengembangannya. Semenjak itu konsursium Eclipse *Foundation* mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya. 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 dipasang. 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. Salah satu situs yang menawarkan *plug-in* secara gratis seperti Eclipse downloads by *project*.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 *Deploymentdan maintenance* untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya.

2.8.1 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.

2.8.2 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.

2.8.3 Android Development Tool

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 *debugpanelouput*. Selain itu ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.

2.9 Pengertian SqLite

SQLite merupakan sebuah *library* proses yang menerapkan *serverless* (mandiri tanpa *server*), *zeroconfiguration*, *database* SQL transaksional. Kode untuk SQLite berada dalam public domain dan dengan demikian bebas untukdigunakan untuk tujuan apa pun, komersial atau swasta.

SQLite merupakan mesin *database* SQL *embedded*. Tidak seperti kebanyakan *database* SQL lainnya, SQLite tidak memiliki proses *server* yang terpisah. SQLite membaca dan menulis secara langsung ke disk. *Database* SQL lengkap dengan *multiple* tabel, *indices*, *triggers*, dan *views*, semua terdapat dalam sebuah disk *file* tunggal. *Format file database* adalah *cross-platform* yaitu kita bebas menggandakan *database* antara 32-bit dan sistem 64-bit. Tidak seperti pada paradigma *client-server* umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latencytimes*, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah *file*. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan *file* basis data pada saat sebuah transaksi dimula.[1]

2.10 Skala Pengukuran

Skala pengukuran merupakan kesepakatan yang digunakan sebagai acuan untuk menentukan panjang pendeknya interval yang ada dalam alat ukur, sehingga alat ukur tersebut bila digunakan dalam pengukuran akan menghasilkan data kuantitatif [9].Dengan skala pengukuran ini, maka nilai variabel yang diukur 10 dengan instrumen tertentu dapat dinyatakan dalam bentuk angka, sehingga akan lebih akurat, efisien, dan komunikatif.

2.10.1 SkalaLikert [9]

Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dengan skala Likert, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item–item instrumen yang dapat berupa pernyataan atau pertanyaan. Jawaban setiapitem.

Instrumen yang menggunakan skala likert mempunyai gradasi dari sangat positif sampai sangat negatif, yang dapat berupa kata–kat antara lain: sangat setuju dengan skor 5, setuju dengan skor 4, ragu–ragu dengan skor 3, tidak setuju dengan skor 2, sangat tidak setuju dengan skor 1.Instrument peneletian yang menggunakan skala likert dapat dibuat dalam bentuk checklist ataupun pilihan ganda.

Contoh :

Berilah jawaban pertanyaan berikut sesuai dengan pernyataan anda, dengan cara memberi tanda (√) pada kolom yang tersedia.

Tabel 2.1 Tabel pertanyaan skala likert

No.	Pertanyaan	Jawaban				
		SS	ST	RG	TS	STS
1. 2.	Prosedur kerjayangbaru itu akan segera diterapkan diperusahaan anda.		√			

SS= SangatSetuju	diberiskor	5
ST= Setuju	diberiskor	4
RG= Ragu-ragu	diberiskor	3
TS= Tidak Setuju	diberiskor	2
STS= Sangat Tidak Setuju	diberiskor	1

Kemudian dengan teknik pengumpulan data angket, maka instrument tersebut misalnya diberikan kepada 100 orang karyawan yang diambil secara *random*. Dari 100 orang karyawan setelah dilakukan analis misalnya:

25	Orangmenjawab	SS
40	Orangmenjawab	ST
5	Orangmenjawab	RG
20	Orangmenjawab	TS
10	Orangmenjawab	STS

Data interval tersebut juga dapat dianalisis dengan menghitung rata-rata

jawaban berdasarkan skoring setiap jawaban dari responden. Berdasarkan skor yang telah ditetapkan dapat dihitung dengan rumus berikut:

$$Y = \frac{X}{Skor\ ideal} X100\%$$

Keterangan:

X = Jumlah frekuensi dikalikan dengan nilai kategori jawaban (∑ = N R)

N = Nilai dari setiap jawaban

R = Frekuensi

Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel (5 x 100= 500)

Y = Nilai persentase yang dicari

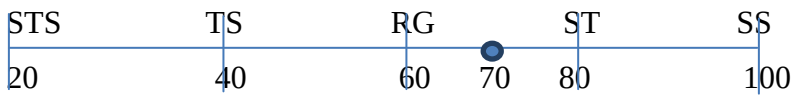
Setelah diperoleh nilai Y maka interpresentasi nilai dapat dilihat sebagai berikut:

Tabel 2.2 interpresentasi nilai

Skala (N)	Keterangan	Responden(R)	N x R
5	Sangat Setuju	25	125
4	Setuju	40	160
3	Ragu-ragu	5	15
2	Tidak setuju	20	40
1	Sangat tidak setuju	10	10
	Jumlah	100	350

$$Y = \frac{350}{500} X100\% = 70\ \%$$

Secarakontinumdapatdigambarkan pada gambar 2.2 :



Gambar 2.2Gambar Interpretasi Skor Persetujuan Metode KerjaBaru

Jadi berdasarkan data yang diperoleh dari 100 responden maka rata-rata 350 terletak pada daerah setuju.

2.11 Metode Pengujian

Pengujian alfa merupakan pengujian yang dilakukan oleh pembuat aplikasi atau orang-orang yang terlibat didalamnya. Sedangkan beta dilakukan oleh orang-orang yang tidak terlibat dalam pengembangan proyek atau pengujian ini dilakukan dari sisi user, baik seorang maupun beberapa orang.[6]