

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Barter**

##### 2.1.1. Pengertian Barter

Barter adalah kegiatan tukar-menukar barang atau jasa yang terjadi tanpa perantaran uang. Tahap selanjutnya menghadapkan manusia pada kenyataan bahwa apa yang diproduksi sendiri tidak cukup untuk memenuhi kebutuhannya. Untuk memperoleh barang-barang yang tidak dapat dihasilkan sendiri mereka mencari dari orang yang mau menukarkan barang yang dimilikinya dengan barang lain yang dibutuhkannya. Akibatnya barter, yaitu barang ditukar dengan barang. Pada masa ini timbul benda-benda yang selalu dipakai dalam pertukaran. Kesulitan yang dialami oleh manusia dalam barter adalah kesulitan mempertemukan orang-orang yang saling membutuhkan dalam waktu bersamaan. Kesulitan itu telah mendorong manusia untuk menciptakan kemudahan dalam hal pertukaran, dengan menetapkan benda-benda tertentu sebagai alat tukar. Sampai sekarang barter masih dipergunakan pada saat terjadi krisis ekonomi di mana nilai mata uang mengalami devaluasi akibat hiperinflasi dan jatuhnya sistem pemerintahan.(Wikipedia.com)

##### 2.1.2. Tahapan Barter

Sebelum manusia mulai mengenal hidup bermasyarakat mula-mula semua kebutuhan hidupnya dicukupi sendiri tanpa bantuan orang lain. Namun setelah kehidupan semakin maju, manusia menyadari bahwa mereka tak mungkin memenuhi semua kebutuhan hidupnya sendiri. Oleh karena itu, mereka kemudian saling bertukar barang, misalnya singkong ditukar dengan ayam atau buah ditukar dengan kelapa. Tukar menukar antara barang dengan barang disebut dengan barter, dan inilah awal mula terjadinya uang.

### 2.1.2.1. Syarat – Syarat terjadi barter

Transaksi barter memiliki syarat agar terjadi, yaitu sebagai berikut :

- Orang yang diajak bertukar barang membutuhkan barang yang akan ditukarkan oleh orang yang akan mengajak bertukar barang.
- Barang yang akan ditukarkan dianggap mempunyai nilai sama.
- Orang yang diajak bertukar barang memiliki barang yang dibutuhkan oleh orang yang mengajak bertukar barang.

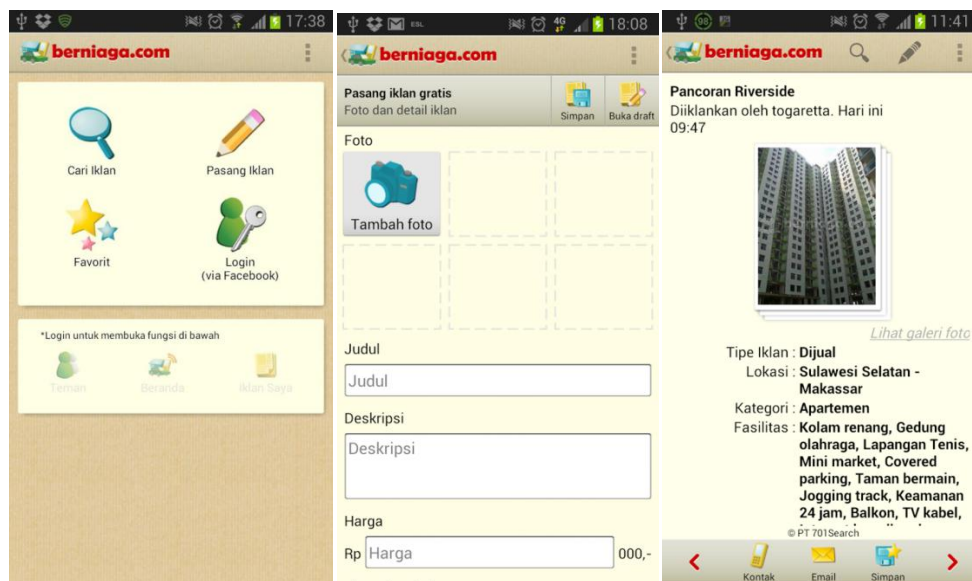
### 2.1.2.2. Kelemahan Transaksi Barter

Transaksi barter memiliki Kelemahan, yaitu sebagai berikut :

- Pertukaran sifatnya hanya kebetulan;
- Barter Memerlukan “Kehendak Ganda yang Selaras” Maksud dari “kehendak ganda yang selaras” atau “double coincidence of wants” adalah tiap pihak yang ingin melakukan pertukaran memiliki barang yang diinginkan pihak lain, dan mencari barang yang dimiliki pihak lain
- Kesulitan mencari lawan tukar karena pertukaran terjadi dasarnya suka sama suka;
- Tidak ada satuan ukuran nilai yang pasti.

## 2.2. Penelitian Terdahulu

Saat ini ditemui aplikasi perniagaan berbasis web maupun berbasis Android. Salah satunya Berniaga.com berbasis Android. Keunggulan dari aplikasi ini adalah fitur yang sangat mendekati lengkap. Tetapi punya kekurangan diantaranya tampilan yang kurang menarik atau terlalu sederhana, pasang tidak adanya batasan waktu kadaluarsa untuk iklan yang di posting.



Gambar 2.1. Penelitian terdahulu

## 2.3. Program Pendukung

Aplikasi ini akan dibuat dalam bentuk Android sehingga membutuhkan beberapa program pendukung dalam pengerjaannya. *Software* yang digunakan adalah *Eclipse*, *plugin Phonegap*, serta *database mysql*.

### 2.3.1. Eclipse (IDE)

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

- *Multi-platform* : Target sistem operasi *Eclipse* adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- *Multi-language* : *Eclipse* dikembangkan dengan bahasa pemrograman *Java*, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
- *Multi-role* : Selain sebagai IDE untuk pengembangan aplikasi. *Eclipse* pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya. (Wina Noviani, 2011)

### 2.3.2. Android SDK (Software Development Kit)

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman *Java*. Android merupakan subnet perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* aplikasi-netral, Android memberi Anda kesempatan

untuk membuat Aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Handphone/Smartphone.

### 2.3.3. ADT (Android Development Tools)

*Android Development Tools* (ADT) adalah *plugin* yang didesain untuk IDE *Eclipse* yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE *Eclipse*. Dengan menggunakan ADT untuk *Eclipse* akan memudahkan kita dalam membuat aplikasi *project* android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, begitu juga kita dapat melakukan running aplikasi menggunakan Android SDK melalui *eclipse*. Dengan ADT juga kita dapat melakukan pembuatan *package* android (.apk) yang digunakan untuk distribusi aplikasi android yang kita rancang.

### 2.3.4. JDK (Java Development Kit)

*Java Development Kit* (JDK) adalah lingkungan pemrograman untuk menulis program-program aplikasi dan *applet java*, JDK terdiri dari lingkungan eksekusi program yang berada diatas Operating System, sebagaimana dibutuhkan oleh para *programmer* untuk meng-compile, membenahi *bug(s)* yang ada, dan menjalankan tambahan2 dari program intinya (*applets*) yang ditulis dengan menggunakan Bahasa pemrograman *Java*.

### 2.3.5. Phonegap

Phonegap adalah aplikasi yang menampung dan mengizinkan anda untuk membangun secara *native* aplikasi yang diinstall menggunakan HTML, CSS dan javaScript. Dengan menggunakan phonegap, pengembangan dapat menulis aplikasi mereka sekali dan menyebarkan keenam *platform mobile* utama dan toko aplikasi, termasuk Apple iOS, Android, Blacberry, WebOS, Samsung Bada dan Symbian. (Afandi Nur Azis, 2008)

### 2.3.6. PHPMyAdmin

PHPMyAdmin adalah sebuah aplikasi di mana seseorang dapat mengontrol data mereka dan isi web mereka untuk ditampilkan dalam sebuah website yang mereka buat. Dengan PHPMyAdmin Anda bisa mengontrol semua file yang ada di website Anda

## 2.4. UML (Unified Modeling Language)

### 2.4.1. Use Case Diagram

*Use Case Diagram* menekankan pada ‘apa’ yang akan dikerjakan oleh sistem tanpa mempedulikan bagaimana sistem akan melakukannya secara rinci (penggambaran fungsionalitas sistem yang berinteraksi dengan aktor). Contohnya *login* ke sistem, meng- *create* sebuah daftar belanja, dan sebagainya. Dalam diagram sebuah objek aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Diagram ini membantu penyusunan requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua *feature* yang ada pada sistem. Sebuah use case dapat meng- *include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Setiap kali use case yang meng- *include* dieksekusi secara normal maka use case yang di- *include* akan dipanggil. Sebuah *use case* dapat di- *include* oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah *use case* juga dapat meng- *extend* use case lain dengan *behaviour* -nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

### 2.4.2. Activity Diagram

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

menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity* diagram merupakan state diagram khusus, di mana sebagian besar state adalah *action* dan sebagian besar transisi di- *trigger* oleh selesainya *state* sebelumnya (internal processing). Oleh karena itu *activity* diagram tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari *level* atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel ( *fork* dan *join* ) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity* diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

#### 2.4.3. Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men- *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline vertikal*. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada *fase* desain berikutnya, message akan dipetakan menjadi operasi/metoda dari classx. *Activation bar* menunjukkan lamanya eksekusi

sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity* .

## 2.5. Android

### 2.5.1. Pengertian Android

Android adalah sebuah sistem operasi Linux yang digunakan untuk telepon seluler (*mobile*) seperti telepon pintar (*smartphone*) dan komputer tablet (PDA). Android menyediakan platform terbuka bagi para pengembangnya untuk menciptakan aplikasi mereka sendiri yang digunakan oleh bermacam peranti bergerak. Android kini telah menjelma menjadi sistem operasi mobile terpopuler didunia. Perkembangan Android tidak lepas dari peran sang raksasa Google.

### 2.5.2. Sejarah Android

Pada awal mulanya Android Inc, merupakan sebuah perusahaan software kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Didirikan oleh beberapa senior di beberapa perusahaan yang berbasis IT & *Communication*; Andy Rubin, Rich Miner, Nick Sears dan Chris White. Menurut Rubin, Android Inc didirikan untuk mewujudkan *mobile device* yang lebih peka terhadap lokasi dan preferensi pemilik. Dengan kata lain, Android Inc, ingin mewujudkan *mobile device* yang lebih mengerti pemiliknya.

Konsep yang dimiliki Android Inc, ternyata menggugah minat Google untuk memilikinya. Pada bulan Agustus 2005, akhirnya Android Inc diakuisisi oleh Google Inc. Seluruh sahamnya dibeli oleh Google. Banyak yang memperkirakan nilai pembelian Android Inc oleh Google adalah sebesar USD 50 juta. Saat itu banyak yang berspekulasi, bahwa akuisisi ini adalah langkah awal yang dilakukan Google untuk masuk kepasar mobile phone. Andy Rubin, Rich Miner, Nick Sears dan Chris White tetap di Android Inc yang dibeli Google, sehingga akhirnya mereka semua menjadi bagian dari raksasa Google



dan sejarah android. Saat itulah mereka mulai menggunakan *platform linux* untuk membuat sistem operasi bagi *mobile phone*.

Saat ini terdapat dua jenis distributor sistem operasi ini didunia. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Service (GMS)*, dan yang kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung dari Google atau dikenal dengan *Open Handset Distribution (OHD)*. Sistem operasi ini membuka pintu untuk para *developer* untuk mengembangkan *software* ini dengan Android SDK (*Software Development Kit*), yang menyediakan *tool* dan API yang dibutuhkan untuk memulai mengembangkan aplikasi pada *platform* Android menggunakan pemrograman *Java*.

### 2.5.3. Perkembangan OS Android

Keunikan dari nama sistem operasi (OS) Android adalah dengan menggunakan nama makanan hidangan penutup (*Dessert*). Selain itu juga nama-nama OS Android memiliki huruf awal berurutan sesuai abjad; *Cupcake, Donut, Eclair, Froyo, Gingerbread, HoneyComb, Ice Cream, Jelly Bean*.

Namun juru bicara Google, Randall Sarafa enggan memberi tahu alasannya. Sarafa hanya menyatakan bahwa pemberian nama-nama itu merupakan hasil keputusan internal dan Google memilih tampil sedikit ajaib dalam hal ini.

#### 2.5.3.1. Android Beta

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

#### 2.5.3.2. Android 1.0 Astro

Pertama kali dirilis pada 23 Spetember 2008. Sebenarnya Android versi pertama ini akan dinamai dengan nama “Astro” tapi

karena alasan hak cipta dan trademark nama "Astro" tidak jadi disematkan pada versi pertama dari OS Android ini. HTC Dream adalah ponsel pertama yang menggunakan OS ini.

#### 2.5.3.3. Android 1.1 Bender

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

#### 2.5.3.4. Android 1.5 Cupcake

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

#### 2.5.3.5. Android 1.6 Donut

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

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

#### 2.5.3.6. Android 2.0/2.1 Éclair

Dirilis pertama kali pada 9 Desember 2009. Terjadi penambahan fitur untuk pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth 2.1*. Beberapa versi updatenya antara Android v.2.0 kemudian v2.0.2 dan terakhir v.2.1.

#### 2.5.3.7. Android 2.2 Froyo (Froze Yoghurt)

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

#### 2.5.3.8. Android 2.3 Gingerbread

Pertama kali diperkenalkan pada 6 Desember 2010. Terjadi banyak peningkatan pada versi Android yang satu ini dibandingkan dengan versi sebelumnya. Dirancang untuk memaksimalkan kemampuan aplikasi dan game. Serta mulai digunakannya Near Field Communication (NFC). Perbaikan terhadap dukungan layar resolusi WXGA dan di atasnya. Beberapa versi *update* yang dirilis antara lain

v.2.3.3 hingga v.2.3.7. Sampai saat ini Android Gingerbread merupakan versi Android yang memiliki pengguna terbanyak dibandingkan dengan seri Android lainnya, yaitu mencapai 65% dari seluruh versi Android yang dirilis.

#### 2.5.3.9. Android 3.0/3.1 Honeycomb

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

#### 2.5.3.10. Android 4.0 ICS (Ice Cream Sandwich)

Pertama kali dirilis pada 19 Oktober 2011. Smartphone yang pertama kali menggunakan OS Android ini adalah Samsung Galaxy Nexus. Secara teori semua perangkat seluler yang menggunakan versi Android sebelumnya, Gingerbread, dapat di-*update* ke Android Ice Cream Sandwich.

#### 2.5.3.11. Android 4.1 Jelly Bean

Android Jelly Bean yang diluncurkan pada 9 Juli 2012 di 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 cepat. Tak ketinggalan *Google Now* juga menjadi bagian yang diperbarui. *Google Now* memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan

olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yakni Google Nexus 7.

#### 2.5.3.12. Android 4.2 Jelly Bean

Fitur *photo sphere* untuk panorama, *daydream* sebagai *screensaver*, *power control*, *lock screen widget*, menjalankan banyak *user* (dalam tablet saja), widget terbaru. Android 4.2 Pertama kali dikenalkan melalui LG Google Nexus 4.

#### 2.5.3.13. Android 4.4.4 Kitkat

Pada 31 Oktober 2013 lalu diperkenalkan Android versi 4.4 dengan KitKat sebagai penamaannya. Pembaharuan yang dapat dilihat pada Android versi ini, antara lain antarmuka atau interface yang lebih canggih, fitur *screen recording*, *support wireless printing*, peningkatan fitur keamanan dan performa.

#### 2.5.3.14. Android 5.0 Lollipop

Sistem operasi Android mengalami peningkatan versi yakni Android versi 5.0 yang merupakan penerus dari versi sebelumnya yakni Android versi 4.4 KitKat. Android ini mengalami beberapa perubahan yang signifikan, antara lain Material Design yakni desain antarmuka atau *interface* yang lebih berwarna dan responsif tentunya.

Fitur terbaru lainnya pada Android Lollipop ini adalah fitur *Project volta* sebagai fitur penghemat daya baterai hingga 30% lebih tahan lama. Peningkatan fitur lainnya seperti keamanan serta notifikasi merupakan hal yang mungkin sudah wajar di telinga para pengguna Android karena disetiap versinya selalu ada peningkatan sistem keamanan serta notifikasi yang kian canggih.

#### 2.5.4. Fitur Android

Fitur yang tersedia di Android adalah:

- Kerangka aplikasi: itu memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- Dalvik mesin virtual: mesin virtual dioptimalkan untuk perangkat telepon seluler.
- Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
- SQLite: untuk penyimpanan data.
- Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- GSM, Bluetooth, EDGE, 3G, 4G dan WiFi (tergantung piranti keras).
- Kamera, Global Positioning System (GPS), kompas, NFC dan accelerometer (tergantung piranti keras).

#### 2.5.5. Arsitektur Sistem Operasi Android

Sistem operasi Android memiliki 5 lapisan (layer) yang merupakan komponen sistem Android.

Google mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Berikut ini susunan dari lapisan-lapisan tersebut jika di lihat dari lapisan dasar hingga lapisan teratas.

##### 2.5.5.1. Linux Kernel

Tumpukan paling bawah pada arsitektur Android ini adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun sistem Android, yang mencakup memory management, security setting, power management, dan beberapa driver hardware. Kernel berperan sebagai abstraction layer antara hardware dan keseluruhan software.

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 (Interprocess Communication) untuk mengatur aplikasi dan lubang keamanan.

#### 2.5.5.2. Android Runtime

Lapisan setelah Kernel Linux adalah 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:

- 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.
- Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual 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. [Nicolas Gramlich, Andbook, anddev.org]

#### 2.5.5.3. Libraries

Bertempat di level yang sama dengan Android Runtime adalah Libraries. Android menyertakan satu set library-library dalam bahasa C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem Android. Kemampuan ini dapat diakses oleh programmer melewati Android application framework. Sebagai contoh Android mendukung pemutaran format audio, video, dan gambar. Berikut ini beberapa core library tersebut System C library, diturunkan dari implementasi standard C system library (libc) milik BSD, dioptimasi untuk piranti embedded berbasis Linux. Media Libraries, berdasarkan PacketVideo's OpenCORE; library-library ini mendukung playback dan recording dari berbagai format audio and video populer, meliputi MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG. Surface Manager, mengatur akses pada display dan lapisan composites 2D and 3D graphic dari berbagai aplikasi. LibWebCore, web browser engine modern yang mensupport Android browser maupun embeddable web view. SGL, the underlying 2D graphics engine.

3D libraries, implementasi berdasarkan OpenGL ES 1.0 APIs; library ini menggunakan hardware 3D acceleration dan highly optimized 3D software rasterizerFreeType, bitmap dan vector font rendering SQLite, relational database engine yang powerful dan ringan tersedia untuk semua aplikasi. Library-library tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan Native Development Toolkit (NDK).



#### 2.5.5.4. Applications Framework

Lapisan selanjutnya adalah application framework, yang mencakup program untuk mengatur fungsi-fungsi dasar smartphone. Application Framework merupakan serangkaian tool dasar seperti alokasi resource smartphone, aplikasi telepon, pergantian antar – proses atau program, dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada tool-tool dasar tersebut, dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks. Programmer mendapatkan akses penuh untuk memanfaatkan API-API (Android Protocol Interface) yang juga digunakan core applications. Arsitektur aplikasi didesain untuk menyederhanakan pemakaian kembali komponen-komponen, setiap aplikasi dapat menunjukkan kemampuannya dan aplikasi lain dapat memakai kemampuan tersebut. Mekanisme yang sama memungkinkan pengguna mengganti komponen-komponen yang dikehendaki. Di dalam semua aplikasi terdapat servis dan sistem yang meliputi : Satu set Views yang dapat digunakan untuk membangun aplikasi meliputi lists, grids, text boxes, buttons, dan embeddable web browser. Content Providers yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain (misalnya Contacts), atau untuk membagi data yang dimilikinya. Resource Manager, menyediakan akses ke non-code resources misalnya localized strings, graphics, dan layout files. Notification Manager yang memungkinkan semua aplikasi untuk menampilkan custom alerts pada the status bar. Activity Manager yang manage life cycle of dari aplikasi dan menyediakan common navigation backstack.

#### 2.5.5.5. 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. Selain komponen sistem diatas android juga memiliki komponen yang cukup penting yaitu Komponen Aplikasi. 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 [developer.android.com]. 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.

## 2.6. JQuery Mobile

jQuery Mobile merupakan kerangka kerja Java Script seperti halnya JQuery pada *desktop*. Namun penggunaannya khusus ditargetkan untuk perangkat bergerak seperti iPad, iPhone, Blackberry, Symbian, Android, dll. jQuery Mobile memungkinkan pembuatan aplikasi web yang *multi platform*, atau tidak tergantung pada piranti keras tertentu. jQuery Mobile ini juga telah mendukung penggunaan layar sentuh, sehingga aplikasi kita dapat mengoptimalkan perangkat yang ada. (Afandi Nur Azis,2008)

Satu set jQuery plug-in dan widget yang bertujuan untuk menyediakan platform API lintas untuk membuat aplikasi web mobile. Dalam hal pelaksanaan

kode, jQuery Mobile adalah sangat mirip dengan jQuery UI, tapi sementara jQuery UI difokuskan pada aplikasi desktop, jQuery Mobile dibangun dengan perangkat mobile yang di inginkan.

Mengenai hal tersebut, jQuery Mobile masih dalam rilis Alpha 4, dengan beta hanya sekitar sudut. Masih banyak masalah yang sedang diperbaiki, tetapi jQuery Mobile tim pengembangan telah mengatakan bahwa mereka menganggap perpustakaan sebagai fitur-lengkap untuk rilis 1.0 mereka. Meskipun dokumentasi jarang, terutama dibandingkan dengan dokumentasi yang ekstensif untuk proyek jQuery itu sendiri, forum sangat aktif.

Meski begitu, banyak orang yang sudah menggunakan jQuery Mobile dalam produksi, yang merupakan bukti tidak hanya untuk stabilitas dan kualitas perpustakaan, tapi bagaimana mudahnya untuk digunakan. (Jhon Reid,2011)

#### 2.6.1. Sekilas Library jQuery Mobile

Pada tulisan ini, jQuery Mobile terdiri dari empat file: file JavaScript, file CSS, dan dua sprite grafis PNG.

JavaScript file dimaksudkan untuk dimuat setelah dasar jQuery perpustakaan. File script ini melakukan berbagai tugas, seperti membuat widget, menerapkan pendengar acara, dan memungkinkan API.

jQuery Mobile juga mencakup Cascading Style Sheet yang menentukan tata letak dan penampilan halaman elemen jQuery Mobile. The Style Sheet juga menetapkan transisi dan animasi dengan transformasi CSS3.

Akhirnya, jQuery Mobile mencakup seperangkat kecil grafis untuk elemen antarmuka pengguna. Hal ini sangat sederhana, ikon standar untuk navigasi.