BAB II

LANDASAN TEORI

2.1 Aplikasi

Sumber Wikipedia (28:05:2013) 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.

2.2 Hotel

Sumber Wikipedia (28:05:2013) Hotel berasal dari kata hostel, konon diambil dari bahasa Perancis kuno. Bangunan publik ini sudah disebut-sebut sejak akhir abad ke-17. Maknanya kira-kira, "tempat penampungan buat pendatang" atau bisa juga "bangunan penyedia pondokan dan makanan untuk umum". Jadi, pada mulanya hotel memang diciptakan untuk meladeni masyarakat.

Tak aneh kalau di Inggris dan Amerika, yang namanya pegawai hotel dulunya mirip pegawai negeri alias abdi masyarakat. Tapi, seiring perkembangan zaman dan bertambahnya pemakai jasa, layanan inap-makan ini mulai meninggalkan misi sosialnya. Tamu pun dipungut bayaran. Sementara bangunan

dan kamar-kamarnya mulai ditata sedemikian rupa agar membuat tamu betah. Meskipun demikian, bertahun-tahun standar layanan hotel tak banyak berubah.

Sampai pada tahun 1793, saat City Hotel dibangun di cikal bakal wilayah kota New York. City Hotel itulah pelopor pembangunan penginapan gaya baru yang lebih *fashionable*. Sebab, dasar pembangunannya tak hanya mementingkan lokasi yang strategis. Tapi juga pemikiran bahwa hotel juga tempat istirahat yang mumpuni. Jadi, tak ada salahnya didirikan di pinggir kota.

Di Indonesia, kata hotel selalu dikonotasikan sebagai bangunan penginapan yang cukup mahal. Umumnya di Indonesia dikenal hotel berbintang, hotel melati yang tarifnya cukup terjangkau namun hanya menyediakan tempat menginap dan sarapan pagi, serta *guest house* baik yang dikelola sebagai usaha swasta (seperti halnya hotel melati) ataupun *mess* yang dikelola oleh perusahaan-perusahaan sebagai tempat menginap bagi para tamu yang ada kaitannya dengan kegiatan atau urusan perusahaan.

Biaya dan kualitas hotel biasanya berdasarkan jangkauan dan tipe pelayanan yang tersedia. Dikarenakan kenaikan besar-besaran dalam turisme di seluruh dunia, selama dekade terakhir abad 20, pendirian hotel terutama yang kecil telah meningkat secara drastis. Untuk pembandingan, sistem penilaian telah diperkenalkan satu sampai lima bintang dan juga tingkat melati di Indonesia yang lebih murah.

2.3 Android

Menurut Nazruddin Safaat H. (2012:1) Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi.

Android dikembangkan bersama antara Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, NVIDIA yang tergabung dalam OHA (*Open Handset Alliance*) dengan tujuan membuat sebuah standar terbuka untuk perangkat bergerak (*mobile device*).

Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android dipuji sebagai "*platform mobile* pertama yang Lengkap, Terbuka dan Bebas"

a. Lengkap (*Complete Platform*): Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* android. Android merupakan sistem operasi yang aman dan banyak meyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.
b. Terbuka (*Open Source Platform*): Platform Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.
c. Bebas (*Free Platform*): Android adalah *platform*/aplikasi yang bebas untuk *develop*. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan

biaya pengujian. Tidak ada kontrak yang diperlukan. Aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

2.3.1 Sejarah Android

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuat lisensi perangkat lunak dan *open platform* perangkat selular.

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Telepon selular 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 Holdingd, Atheros *Communicatons*, 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 *mobile* yang merupakan modifikasi kernel Linux 2.6. sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan *bug* dan penambahan fitur baru.

Pada masa saat ini sebagian besar *vendor-vendor smartphone* sudah memproduksi *smartphone* berbasis android, *vendor-vendor* ini antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell,

Nexus, SciPhone, WayteQ, Sony Ericsson, Acer, Philips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi *vendor smartphone* di dunia yang memproduksi android. Hal ini, karena android itu adalah sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh *vendor* manapun.

2.3.2 Arsitektur Android

Secara garis besar, Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut:

1. *Application* dan *Widgets*

Application dan widget ini adalah *layer* sebagai pengubung dengan aplikasi, di mana biasanya aplikasi tersebut dapat di-*download*, diinstalasi dan kemudian dijalankan. Di *layer* terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

2. *Application Frameworks*

Android adalah *Open Development Platform*, yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan *status notifications*, dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang

supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*).

Sehingga bisa kita simpulkan *Application Frameworks* ini adalah *layer* dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa SMS dan panggilan telepon.

Komponen-komponen yang termasuk ke dalam *Application Framework* adalah sebagai berikut:

a. *Views*
b. *Content Providers*
c. *Resource Manager*
d. *Notification Manager*
e. *Activity Manager*

3. Libraries

Libraries adalah *layer* di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas Kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL, serta:

a. *Libraries* media untuk pemutaran media audio dan video
b. *Libraries* untuk manajemen tampilan
c. *Libraries Graphics* mencangkup SGL dan OpenGL untuk grafis 2D dan 3D

d. *Libraries* SQLite untuk dukungan *database*

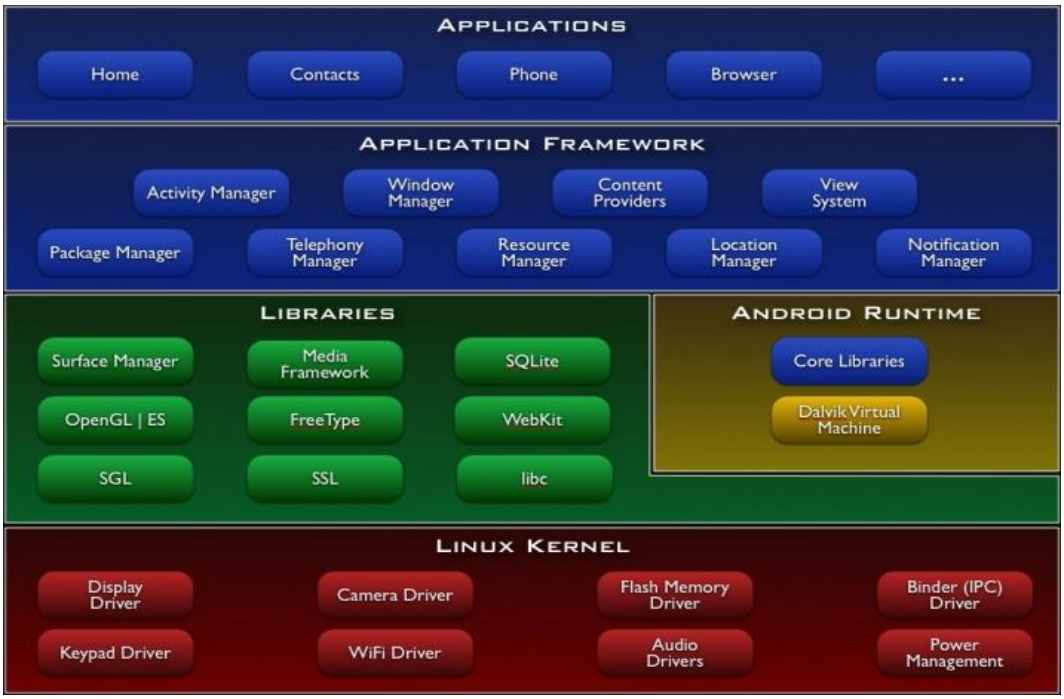
e. *Libraries* SSL dan WebKit terintegrasi dengan *web browser* dan *security*

f. *Libraries* LiveWebcore mencangkup *modern web browser* dengan *engine embeded web view*

g. *Libraries* 3D yang mencangkup implementasi OpenGL ES 1.0 API'S

4. Linux Kernel

Linux Kernel adalah layer dimana inti dari *Operating System* dari Android itu berada. Berisi *file-file system* yang mengatur sistem *processing, memory, resource, drivers* dan sistem-sistem operasi Android lainnya. Linux Kernel yang digunakan Android adalah Linux Kernel release 2.6.



Gambar 2.1 Arsitektur Android

Sumber: http://elinux.org

2.3.3 Komponen Aplikasi Android

Aplikasi Android ditulis dalam bahasa pemrograman java. Kode java dikompliasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi di mana prosesnya di package oleh *tools* yang dinamakan "apt *tools*" ke dalam paket Android sehingga menghasilkan *file* dengan ekstensi apk. *File* apk itulah yang kita sebut dengan aplikasi, dan nantinya dapat di-*install* di perangkat *mobile.* Ada empat jenis komponen pada aplikasi Android yaitu:

1. *Activities* merupakan komponen utama untuk menyajikan *user interface* (tampilan program) pada pengguna.
2. *Service* merupakan komponen yang tidak memiliki *user interface* (tampilan program), tetapi service berjalan secara *backgrounds.*
3. *Broadcast Receiver* merupakan komponen yang berfungsi menerima dan bereaksi untuk menyampaikan notifikasi.
4. *Content Provider* merupakan komponen membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain.

2.3.4. Versi Android

a. 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 *e-mail*.

b. 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.

c. 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; kamera, *camcorder* dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-speech engine*, kemampuan dial kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel), pengadaan resolusi VWGA.

d. 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.

e. 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, integrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD Card, kemampuan WiFi *Hotspot portabel*, dan kemampuan *auto update* dalam aplikasi Android Market.

f. 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 antar muka (*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.

g. 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 *multiprocessor* dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.

h. Android versi 4.0 (ICS: *Ice Cream Sandwich*)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk 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. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

i. Android versi 4.1 (*Jelly Bean*)

Android *Jelly Bean* yaang 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 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.

2.4 Sistem Layanan Berbasis Lokasi (*Location Based Service*)

Menurut Nazruddin Safaat (2012:218) *Location-Based Service* (LBS) atau layanan berbasis lokasi, adalah istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan.

Dua unsur utama LBS adalah:

1. *Location Manager* (API *Maps*)

Menyediakan *tools/source* untuk LBS, *Application Programming Interface* (API) Maps menyediakan fasilitas untuk menampilkan, memanipulasi *maps*/peta berserta fitur-fitur lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya. Paket ini berada pada com.gooogle.android.maps.

2. *Location Providers* (API *Location*)

Menyediakan teknologi pencarian lokasi yang digunakan oleh *device*/perangkat. API *Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. API location berada pada paket Android yaitu dalam paket android.location. Dengan *Location Manager*, kita dapat menentukan lokasi kita saat ini, *Track* gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

Ada dua tipe layanan yang bisa digunakan *device*/perangkat dalam LBS untuk memperoleh posisi pengguna, yaitu dengan menggunakan posisi sel jaringan atau *Global Positioning System* (GPS) maupun aGPS. Dari kedua cara ini akan didapatkan posisi pengguna dalam bentuk koordinat *latitude* dan *longitude*. *Latitude* adalah representasi dari arah Utara-Selatan, sedangkan *longitude* adalah representasi dari arah Timur-Barat.

Selain dari sisi posisi pengguna, LBS juga bisa dilihat dari sisi layanan yang diberikan. Dari sisi layanan yang diberikan, LBS bisa dibagi menjadi dua yaitu *reactive* LBS dan *proactive* LBS. *Reactive* LBS adalah layanan yang hanya aktif jika ada aksi yang dilakukan pengguna. Layanan yang hanya akan memberi jika ada permintaan dari pengguna. Sedangkan *proactive* LBS merupakan layanan yang akan selalu memberi informasi kepada pengguna walaupun pengguna tidak melakukan permintaan terhadap layanan.

2.5 Google *Maps*

Menurut Wikipedia (30:05:2013) Google Maps adalah sebuah jasa peta globe virtual gratis dan online disediakan oleh Google dapat ditemukan di http://maps.google.com. Ia menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia

Layanan Google *Maps* ini memungkinkan para pengembang untuk mengintegrasikan Google Mpas ke dalam *website* masing-masing dengan menambahkan data *point* sendiri. Dengan menggunakan Google Maps API, Google Maps dapat ditampilkan pada *website* eksternal. Agar aplikasi Google Maps dapat muncul di *website* tertentu, diperlukan kode unik yang digenerasikan oleh Google untuk suatu *website* tertentu, agar server Google Mpas dapat mengenali.

Pada Google Maps API terdapat 4 jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah:

a. *Roadmap*, untuk menampilkan peta biasa 2 dimensi
b. *Sattelite*, untuk menampilkan foto satelit
c. *Terrain*, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai
d. *Hybrid*, akan menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada Roadmap (jalan dan nama kota).

2.6 Konsep Perancangan Berorientasi Objek

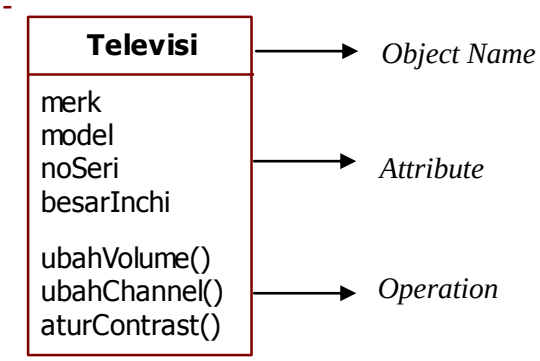
Untuk memahami titik pandang dan maksud dari "berorientasi objek" kita dapat mempelajarinya dari alam secara luas. Objek ada disekeliling kita, baik yang konkrit atau konseptual. Dalam susut pandng Eksekutif perusahaan : Karyawan, Absesnsi, Gaji, Profit dapat disebut sebagai Objek. Seorang Arsitek melihat Gedung, Biaya dan tenaga kerja sebagai objek. Konsep-konsep dasar dalam memahami Objek dapat dilihat pada subjudul berikut:

2.6.1 *Object* / Objek

Menurut Yasmi Afrizal dan Wahyuni (2013:165) Objek adalah orang, tempat, benda, kejadian atau konsep-konsep yang ada di dunia nyata dan penting bagi suatu aplikasi. Sebuah objek adalah Entitas yang memiliki Identitas, *States* (keadaan sesaat) dan *Behavior* (perilaku).

State sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam **Atribut** atau *property*. *Behavior* sebuah objek mendefinisikan bagaimana sebuah objek bertindak/bereaksi yang dinyatakan dalam ***Operation*.** Satu objek dapat diturunkan menjadi objek dalam bentuk lain, kemudian saling mengkait menyusun sesuatu yang lebih rumit.

Langkah pertama yang harus dilakukan dalam pengembangan perangkat lunak berorientasi objek adalah melakukan Abstraksi yaitu kegiatan atau suatu usaha untuk mengenali objek-objek dan mengelompokkannya kedalam suatu kelas. Misalkan objek Hewan : Unggas, Reptil, maka Unggas dan Reptil adalah kelas-kelas dalam objek Hewan. Tata cara atau notasi pembuatan entitas objek digambarkan sebagai berikut:



Gambar 2.2 Notasi Pembuatan Entitas Objek

Sumber : Yasmi Afrizal dan Wahyuni (2012:166)

2.6.2 *Class* / Kelas

Menurut Yasmi Afrizal dan Wahyuni (2013 : 166) *Class* adalah kumpulan atau himpunan objek-objek yang sejenis, memiliki kesamaan atribut/*property*, perilaku, serta relasi dengan objek lain yang mirip.

Notasi kelas digambarkan dengan kotak, dengan nama kelas didalamnya ditulis menggunakan huruf besar di awal kata. Bila sebuah kelas memiliki 2 suku kata atau lebih, maka penulisannya disatukan tanpa spasi dengan huruf awal tiap suku menggunakan huruf besar. Contohnya adalah Barang Elektronik dapat dikatakan sebagai sebuah Kelas apabila memiliki kesamaan dengan objek yang ada padanya misalnya Mesin Cuci, Televisi, Radio, Kulkas adalah objek-objek yang dapat dikelompokkan kedalam satu kelas yaitu Barang Elektronik rumah tangga.



Gambar 2.3 Notasi Class

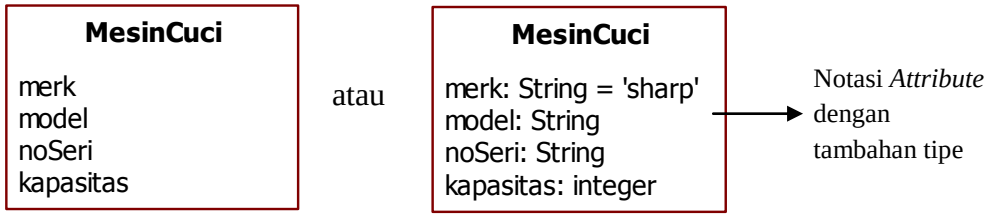
Sumber : Yasmi Afrizal dan Wahyuni (2013:166)

2.6.3 *Attribute* / Atribut

Menurut Yasmi Afrizal dan Wahyuni (2013 : 166) *Attribute* adalah data yang dimiliki suatu objek atau *property* dari sebuah *Class* yang menggambarkan batas nilai yang mungkin ada pada objek dari kelas.

Sebuah atribut bisa memiliki nol atau lebih atribut. Notasi atribut digambarkan dengan kotak dibawah *class*, dengan nama atribut didalamnya ditulis menggunakan huruf kecil. Jika sebuah atribut memiliki 2 atau lebih suku kata, maka semua suku kata ditulis disatukan tanpa spasi, awal suku kata pertama dengan huruf kecil dan awal suku kata berikutnya dengan huruf besar.

Notasi atribut dapat ditambahkan informasi dengan tipe-tipe atribut tersebut. Penulisan tipe pada atribut dipisahkan denngan tanda titik dua (:), tipe ditambahkan berupa String, Floating-Point number, Integer dan Boolean.



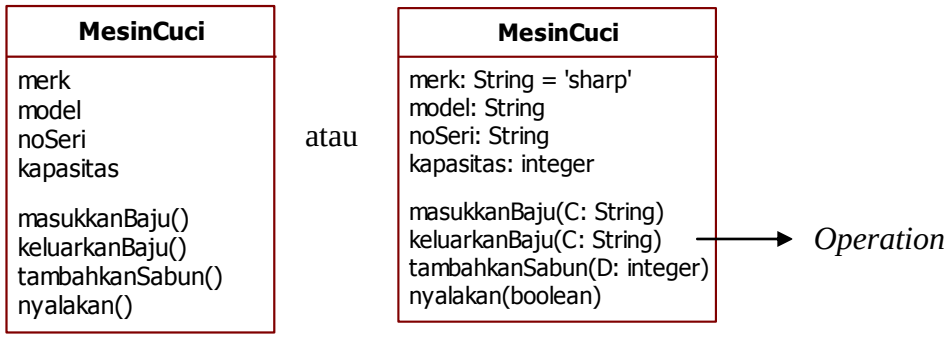
Gambar 2.4 Notasi *Attribute*

Sumber : Yasmi Afrizal dan Wahyuni (2013:167)

2.6.4 *Operation* / Operasi

Menurut Yasmi Afrizal dan Wahyuni (2012 : 167) *Operation* adalah sesuatu yang bisa dilakukan oleh sebuah *class*.

Notasi penulisannya sama dengan atribut. Bagian *operation* ini juga bisa diberikan tambahan informasi, yaitu dengan menambahkan parameter yang akan dilakukan operation dalam tanda kurung. Contoh parameternya adalah *function*.

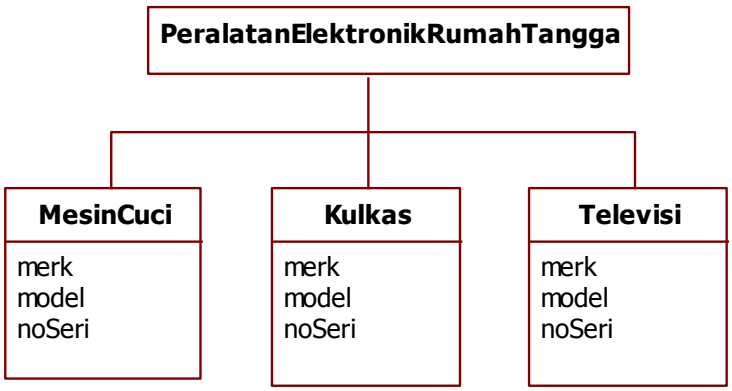


Gambar 2.5 Notasi *Operation*

Sumber : Yasmi Afrizal dan Wahyuni (2013:167)

2.6.5 *Inheritance* / Pewarisan

Menurut Yasmi Afrizal dan Wahyuni (2012 : 168) *Inheritance* atau pewarisan memungkinkan dibuat *class* yang menyerupai *class* lain yang telah ada sebelumnya, tetapi masih memiliki beberapa sifat induknya. Misalkan dari sebuah mobil biasa, anda dapat membuat mobil balap serta mobil angkutan umum. Prosesnya adalah dengan mengubah sifat dari mobil biasa tersebut.



Gambar 2.6 Penggambaran *Inheritance*

Sumber : Yasmi Afrizal dan Wahyuni (2013:168)

2.6.6 *Polymorphism* / Kebanyakrupaan

Menurut Yasmi Afrizal dan Wahyuni (2012 : 165) *Polymorphisme* adalah objek yang memiliki fungsi sama dengan objek dasar tetapi memiliki satu atau lebih sifat berbeda.

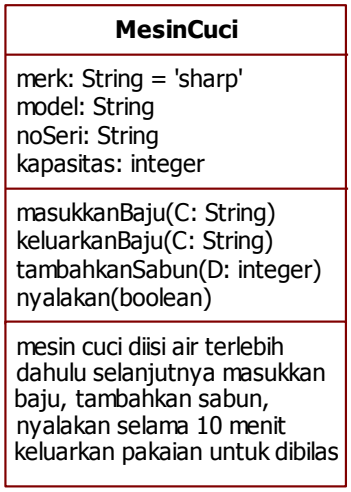
Polymorphism dengan kata lain adalah pemisahan secara jelas diantara subsistem yang berbeda. Sebagai contoh misalkan sebuah kelas memiliki operasi 'OPEN', operasi open ini bisa dipakai untuk membuka pintu, membuka buku, membuka baju dan lainnya. Meskipun 'OPEN' memiliki tujuan yang sama, tapi apa yang dilakukannya berbeda.

2.6.7 *Encapsulation* / Pembungkusan

Menurut Yasmi Afrizal dan Wahyuni (2012 : 168) *Encapsulation* sering disebut dengan penyembunyian informasi (*Hidding*), suatu konsep berdasarkan fakta di dunia nyata yang menyatakan bahwa segala sesuatu tidak perlu diperlihatkan. Misalnya kita tidak perlu tahu apa yang dilakukan sistem ketika kita menekan *remote* untuk menghidupkan televisi.

2.6.8 *Responsibilities* / Tanggung Jawab

Menurut Yasmi Afrizal dan Wahyuni (2012 : 165) *Responsibilities* adalah model tambahan yang digambarkan pada bagian bawah suatu kelas setelah bagian operasi digunakan untuk menjelaskan pernyataan-pernyataan mengenai apa-apa yang bisa dilakukan oleh kelas tersebut.



Gambar 2.7 Pembuatan *Responsibilities*

Sumber : Yasmi Afrizal dan Wahyuni (2013:168)

2.6.9 *Unified Modeling Language* (UML)

Menurut Han (1998) yang dikutip Yasmi Afrizal dan Wahyuni (2013:183) UML (*Unified Modeling Language*) adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasiyang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML merupakan gabungan dari metode Grady Booch (Booch *Method*), James Rumubaugh (OMT) dan Ivar Jacobson (OOSE). Tetapi UML ini akan mencakup lebih luas daripada OOA&D. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan OMG (*Object Managemet Group*) dengan harapan UML akan menjadi bahsa standar pemodelan masa yang akan datang.

UML disebut sebagai pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagai besar grafik) merupakan notasidari metode yang digunakan untuk mendesain secara cepat.

Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka anda hanya membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desain.

UML merupakan bahasa standar untuk penulisa Blueprint Software yang digunakan untuk Visualisasi (*Visualize*), Spesifikasi (*Specify*), Pembentukan (*Contruct*) dan Pendokumentasian (*Documentation*) alat-alat dari sistem perangkat lunak.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan system.
5. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Keunggulan menggunakan UML dibandingkan menggunakan metodologi terstruktur:

1. Uniformity

Pengembang cukup menggunakan 1 metodologi dari tahap analisis hingga perancangan. Memungkinkan merancang komponen antarmuka secara integrasi bersama perancangan PL dan perancangan struktur data

2. Understandability

Kode yang dihasilakn dapat diorganisasi kedalam kelas-kelas yang berhubungan dengan masalah yang sesungguhnya sehingga lebih mudah dipahami.

3. Stability

Kode program yang dihasilkan relative stabil sepanjang waktu, karena mendekati permaslahan yang sesungguhnya.

4. Reusability

Dengan metodologi berorientasi objek, dimungkinkan pengguna ulang kode, sehingga pada akhirnya akan sangat mepercepat waktu pengembangan perangkat lunak (atau sistem informasi).

2.7 PHP

Menurut Muhammad Miftakul Amin (2010:1) PHP merupakan bahasa pemrograman server side scripting yang lahir sejalan dengan perkembangan internet. PHP (PHP: *Hypertext Processor*) merupakan sebuah script yang terintegrasi dengan HTML dan mampu menyajikan informasi yang dinamis.

PHP adalah produk *open source* yang dapat digunakan secara gratis tanpa harus membayar untuk menggunakannya. Intepreter PHP dalam mengeksekusi kode PHP pada sisi server (disebut *server-side*), sedangkan tanpa adanya

interpreter PHP, maka semua skrip dan aplikasi PHP yang dibuat tidak dapat dijalankan.

PHP merupakan bahasa standar yang digunakan dalam dunia *website*, PHP adalah bahasa program yang berbentuk skrip yang diletakkan didalam server web. Jika dilihat dari sejarah mulanya PHP diciptakan dari ide Rasmus Lerdof untuk kebutuhan pribadinya, skrip tersebut sebenarnya dimaksudakan untuk digunakan sebagai keperluan untuk membuat *website* pribadi, akan tetapi kemudian dikembangkan lagi sehingga menjadi sebuah bahasa yang disebut "*Personal Home Page*", inilah awal munculnya PHP sampai saat ini.

2.8 MySQL

Menurut Bunafit Nugroho (2005:1) MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multiuser* serta menggunakan perintah standar SQL (*Structured Query Language*)

MySQL memiliki dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. MySQL yang biasa digunakan adalah MySQL *Free Software* yang berada di bawah lisensi GNU/GPL (*General Public License*). MySQL menggunakan logo standar berbentuk lumba-lumba.

MySQL pertama kali dirintis oleh seorang programmer *database* bernama Michael Widenius. Selain sebagai *database* server, MySQL juga merupakan program yang dapat mengakses suatu *database* MySQL yang berpotensi sebagai *server*. Pada saat itu berarti program kita berposisi sebagai *client*. Jadi MySQL adalah sebuah *database* yang dapat digunakan baik sebagai *client* maupun *server*.

Menurut Bunafit Nugroho (2005:3) Database MySQL memiliki beberapa kelebihan dibandingkan dengan database lain, diantaranya adalah:

a. MySQL sebagai *Database Management System* (DBMS)
b. MySQL sebagai *Relation Database Management System* (RDBMS)
c. MySQL adalah sebuah *software database* yang *open source*, yang artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.
d. MySQL merupakan sebuah *database server*, jadi dengan menggunakan database ini, kita dapat menghubungkannya ke media internet sehingga dapat diakses dari jauh.
e. MySQL merupakan sebuah *database client*. Selain menjadi *server* yang melayani permintaan, MySQL juga dapat melakukan *query* dengan mengakses *database* pada *server*. Jadi MySQL juga dapat berperan sebagai *client.*
f. MySQL mampu menerima *query* yang bertumpuk dalam satu permintaan atau *Multi Threading.*
g. MySQL merupakan sebuah *database* yang mampu menyimpan data berkapasitas sangat besar, hingga berukuran Gigabyte sekalipun.
h. MySQL didukung oleh *driver* ODBC, artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi maupun Visual Basic.
i. MySQL adalah *database* menggunakan enkripsi *password*. Jadi *database* ini cukup aman karena memiliki *password* untuk mengaksesnya.

j. MySQL merupakan *server database* yang *multiuser*, artinya *database* ini tidak hanya digunakan oleh sepihak orang, akan tetapi merupakan *database* yang dapat digunakan oleh banyak pengguna.
k. MySQL dapat menciptakan lebih dari 16 kunci per tabel, dan dalam satu kunci memungkinkan berisi belasan *field* (kolom).
l. MySQL mendukung *field* yang dijadikan sebagai kunci primer dan kunci unik.
m. MySQL didukung oleh sebuah komponen dan Perl API, sehingga *database* MySQL dapat diakses melalui sebuah program aplikasi yang berada di bawah protokol internet berupa *Web*. Biasanya aplikasi yang sering digunakan adalah PHP dan Perl.
n. MySQL memiliki kecepatan dalam pembuatan tabel maupun peng-*update*-an tabel.