BAB II

LANDASAN TEORI

2.1. Pengertian Aplikasi

Aplikasi menurut wikipedia (2011) adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yag 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.

Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer.

Program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini. Program inilah yang mengendalikan semua aktivitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instruction set*.

Program aplikasi merupakan program siap pakai. Program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh-contoh aplikasi ialah program pemroses kata dan *web browser*. Aplikasi akan mengunakan sistem operasi (OS) komputier dan aplikasi yang lainnya yang mendukung.

2.2. Pengertian Diet

Diet adalah menjaga pola makan. Makan merupakan cara alamiah agar tubuh mendapatkan energi yang dibutuhkan untuk beraktifitas. Apa yang kita makan, akan sangat berpengaruh kepada tubuh. Salah kaprah tentang pengertian "Diet" yang sering mengidentikkan dengan mengurangi makan, adalah hal ang salah. Diet adalah pengaturan pola makan.

Diet mempunyai arti kombinasi makanan dan minuman di dalam hidangan makan yang dikonsumsi sehari-hari. Ada begitu banyak metode berdiet menurunkan berat badan yang tersedia. Dari Atkins Diet (diet rendah karbohidrat), Cabbage Soup Diet, Grapefruit Diet, dan lain-lain.

Diet yang benar adalah tetap mengonsumsi makanan dengan komposisi yang dibutuhkan tubuh dalam jumlah seimbang. Tentu saja dengan total kalori yang lebih rendah dari yang biasa dikonsumsi, sehingga tubuh akan menggunakan simpanan energi tubuh, yaitu lemak tubuh, baik yang berlokasi di bawah kulit maupun yang berat di dalam tubuh. Dengan hilangnya massa tubuh, maka akan terjadi penurunan berat badan.

Berikut beberapa pola makan yang kerap diasarankan oleh para ahli gizi dan nutrisi:

1. Aturlah jadwal makan menjadi 3 kali sehari, yaitu sarapan, makan siang, dan makan malam, serta 3 kali melakukan *snacking* (makan camilan) 2-3 jam setelah makan, dengan rincian sarapan memenuhi 20 persen dari kebutuhan kalori per hari, makan siang memenuhi 30 persen dari kebutuhan kalori per hari, dan makan malam memenuhi 30 persen dari

kebutuhan kalori per hari. Sedangkan 3 kali *snacking* masing-masing memenuhi kebutuhan 10 persen kalori per hari.

2. Perhatikan asupan nutrisi dari makanan yang dikonsumsi. Secara garis besar, ada tiga nutrisi pokok yang dibutuhkan tubuh, yaitu karbohidrat, protein, dan lemak. Aturlah pola makan dengan komposisi kurang lebih karbohidrat 60 persen, protein 15 persen, dan lemak 25 persen.

Bila berbicara mengenai berat badan tidak ideal, maka kita harus mengetahui bahwa berat badan tidak ideal bisa disebabkan oleh banyak hal. Adapun beberapa penyebab kegemukan di antaranya adalah sebagai berikut:

1. Faktor makan yang melebihi kebutuhan tubuh

   Faktor makan ternyata menduduki peringkat pertama dari faktor-faktor penyebab berat badan tidak ideal. Faktor ini bisa meliputi pola makan yang salah, makan yang berlebihan dan bisa juga karena salah memilih makan dan sebagainya. Untuk mengantisipasinya, sebaiknya kita perbaiki pola makan kita, jauhi makanan yang terlalu berlemak dan atur waktu makan kita.

2. Kurang menggunakan energi

   Pekerjaan dan aktivitas seseorang sehari-hari bisa saja memengaruhi gaya hidupnya. Gaya hidup yang kurang menggunakan aktivitas fisik akan sangat berpengaruh terhadap kondisi tubuh individual tersebut. Karena sesungguhnya, aktivitas fisik tersebut sangat kita perlukan untuk membakar kalori yang ada dalam tubuh kita. Jika jumlah kalori yang masuk di dalam tubuh kita berlebihan dan tidak diimbangi dengan

aktivitas fisik maka seseorang akan menjadi gemuk. Maka dari itu, sudah sewajarnya bila kita mengatur pola hidup kita.

Untuk menghitung apakah berat badan kita ideal diperoleh dari indeks perbandingan antara berat badan dalam satuan kilogram dengan tinggi badan dengan satuan meter, atau sering disebut *Body Mass Index* (BMI). Rumusnya adalah sebagai berikut:

BMI = berat badan / (tinggi badan x tinggi badan) kg/m^2.

Klasifikasi dari indeks tersebut dapat dilihat pada tabel di bawah ini:

Tabel 2.1. Klasifikasi *Body Mass Index*

BMI	**Klasifikasi**
< 14,9	Sangat kurus
15 – 18,4	Kurus
18,5 – 22,9	Normal
23 – 27,5	Gemuk
27,6 – 40	Kegemukan
> 40	Sangat gemuk

Untuk menghitung berat badan ideal kita dapat memakai rumus BMI dengan memasukkan BMI 18,5 untuk batas bawah berat badan ideal dan BMI 22,9 untuk batas atas berat badan ideal. Contohnya sebagai berikut:

Seseorang dengan tinggi 1,7 m akan diukur berat badan idealnya. Maka perhitungannya:

Batas bawah berat badan = BMI x (tinggi badan x tinggi badan)

= 18,5 x (1,7 x 1,7) = 53,46

Batas atas berat badan = BMI x (tinggi badan x tinggi badan)

= 22,9 x (1,7 x 1,7) = 66,18

Jadi orang tersebut memiliki rentang berat badan ideal dari 53,46 sampai 66,18 kg.

Menghitung kalori yang dibutuhkan oleh tubuh memang wajib dilakukan bagi siapa saja. Tidak hanya untuk mereka yang ingin menurunkan berat badan, tetapi semua orang ternyata juga harus mengatur menu diet atau menu makanan sehari-harinya. Segala sesuatu yang berlebihan itu tidaklah baik. Semua harus seimbang di dalam tubuh kita. Terlalu banyak makan bisa mengakibatkan seseorang kelebihan berat badan, dan sebaliknya, terlalu sedikit makan bisa menyebabkan kekurangan gizi.

Untuk orang-orang yang terlanjur berbadan besar, mereka harus berjuang ekstra keras dan juga harus pandai-pandai mengatur siasat agar asupan makanan yang masuk ke dalam tubuh mereka benar-benar pas sesuai standar dan tidak berlebihan. Oleh karena itu, pengaturan menu dieat sebaiknya diterapkan oleh siapa saja. Hal itu bisa dilakukan dengan cara memerhatikan proporsi karbohidrat, protein, lemak serta zat-zat lain dalam tubuh, sehingga seluruhnya berada dalam porsi yang seimbang, pas dan tidak berlebihan.

Bagaimana caranya mengatur sendiri menu diet? Sebelum belajar mengatur menu diet, terlebih dulu harus mengetahui berapa kebutuhan kalori kita perhari. Caranya yaitu sebagai berikut.

Pria = 66 + (13,7 x BB) + (5 x TB) – (6,8 x U)

Wanita = 65,5 + (9,6 x BB) + (1,7 x TB) – (4,7 x U)

Keterangan:

BB = berat badan (kg)

TB = tinggi badan (cm)

U = umur (tahun)

2.3. Perihal *Mobile*

Berbicara tentang teknologi mobile, rasanya di zaman ini bukan istilah baru. Teknologi sudah berkembang sangat pesat. Jauh dari yg dibayangkan di masa lalu. Hampir setiap orang saat ini menggunakan dengan apa yang disebut teknologi mobile. Contoh yang paling sederhana salah satunya adalah *handphone* atau telepon genggam. Teknologi erat kaitannya dengan proses komputasi yang menjadikannya bermanfaat dalam menyelesaikan masalah maupun pekerjaan. Komputasi pada teknologi ini dapat disebut *mobile computing*.

Mobile computing adalah sebuah komputasi menggunakan teknologi yang tidak terhubung secara fisik, atau dalam jarak jauh atau lingkungan mobile (non statik).

Device yang tergolong sebagai *mobile coumputing* cenderung *portable* dan mudah dibawa ke mana saja. Berikut adalah beberapa alat *mobile computing*:

1. Laptop
2. Tablet PC
3. Internet Tablet
4. PDA
5. *Smart phone*

2.4. Pengertian Sistem Operasi

Suatu sistem merupakan rangkaian dari kerja komponen maupun elemen di mana menghasilkan suatu tujuan yang hendak dicapai. Sistem harus diperhatikan agar mencapai sasaran yang dikehendaki.

Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

Menurut Wikipedia (2012) sistem operasi adalah seperangkat program yang mengelola sumber daya perangkat keras komputer, dan menyediakan layanan umum untuk aplikasi perangkat lunak. Sistem operasi adalah jenis yang paling penting dari perangkat lunak sistem dalam sistem komputer. Tanpa sistem operasi, pengguna tidak dapat menjalankan program aplikasi pada komputer mereka, kecuali program aplikasi boot diri.

Biasanya, istilah sistem operasi sering ditujukan kepada semua perangkat lunak yang masuk dalam satu paket dengan sistem komputer sebelum aplikasi-aplikasi perangkat lunak terinstal. Sistem operasi adalah perangkat lunak sistem yang bertugas untuk melakukan kontrol dan manajamen perangkat keras serta operasi-operasi dasar sistem, termasuk menjalankan perangkat lunak aplikasi sperti program-program pengolah kata dan peramban web.

Secara umum, sistem operasi adalah perangkat lunak pada lapisan pertama yang ditempatkan pada memori komputer pada saat komputer dinyalakan. Sedangkan *software-software* lainnya dijalankan setelah sistem operasi berjalan, dan sistem operasi akan melakukan layanan inti umum untuk software-software itu. Layanan inti umum tersebut sperti akses ke disk, manajemen memori, skeduling task, dan antar-muka user. Sehingga masing-masing software tidak perlu lagi melakukan tugas-tugas inti umum tersebut, karena dapat dilayani dan

dilakukan oleh sistem operasi. Bagian kode yang melakukan tugas-tugas inti dan umum tersebut dinamakan dengan "*kernel*" suatu sistem operasi.

2.5. Perihal Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar (*smartphone*) dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak.

2.5.1. Sejarah Android

Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat peluncuran 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, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Service* (GMS)

dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. 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.

Pada masa saat ini kebanyakan vendor-vendor smartphone sudah memproduksi *smartphone* berbasis android, antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Dell, Nexus, SciPhone, Sony Ericsson, LG, Acer, IMO, Asus dan masih banyak lagi vendor yang memakai android pada *smartphone* yang diproduksinya.

2.5.2. Android Sebagai *Platform* Masa Depan

Android dipuji sebagai "*platform mobile* pertama yang lengkap, terbuka, dan bebas".

1. Lengkap (*complete platform*)

   Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* android. Android adalah sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun software dan memungkinkan untuk peluang pembangunan aplikasi.

2. 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.

3. Bebas (*free platform*)

   Android adalah *platform*/aplikasi yang bebas untuk dikembangkan. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* android. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari android dilisensikan di bawah GNU, *General Public* Lisensi versi 2 (GPLv2), yang sering dikenal dengan istilah "*copyleft*" lisensi di mana setiap perbaikan pihak ketiga harus terus jatuh di bawah *terms*. Android didistribusikan di bawah Lisensi *Apache Software* (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen handset khususnya) dapat meilih untuk meningkatkan *platform* tanpa harus memberikan perbaikan mereka ke masyarakat *open source*. Sebaliknya pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka di bawah lisensi apapun yang mereka inginkan. Pengembang aplikasi android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema apapun yang mereka inginkan.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis android. Kebanyakan pengembang menggunakan *Eclipse* yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi android. Eclipse adalah IDE yang paling populer untuk pengembangan android, karena memiliki android *plug-in* yang tersedia untuk memfasilitasi pengembangan android. Selain itu, Eclipse juga mendapat dukungan langsung dari Google untuk menjadi IDE pengembangan aplikasi Android, ini terbukti dengan adanya penambahan *plug-in* untuk Eclipse untuk membuat project android di mana source software langsung dari situs resminya Google. Tetapi hal di atas tidak menutup kemungkinan untuk menggunakan IDE lain seperti Netbeans untuk mengembangkan android.

2.5.3. *The Dalvik Virtual Machine* (DVM)

Salah satu elemen kunci dari Android adalah *Dalvik Virtual Machine* (DVM). Android berjalan di dalam DVM buka di *Java Virtual Machine* (JVM), sebenarnya banyak persamaannya dengan JVM seperti Java ME (*Java Mobile Edition*), tetapi android menggunakan *Virtual Machine* yang disesuaikan dan dirancang untuk memastikan bahwa bebarapa fitur berjalan lebih efisien pada perangkat *mobile*.

DVM adalah "*register based*" sementara JVM adalah "*stack based*", DVM didesain dan ditulis oleh Dan Bornsten dan beberapa *engineer* Google lainnya. DVM menggunakan kernel Linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis Aplikasi C / C + sama halnya seperti pada OS

Linux kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami arsitektur dan proses sistem dari kernel Linux yang digunakan dalam android tersebut.

Semua hardware yang berbasis android dijalankan dengan menggunakan *Virtual Machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executable file,* sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan *tools* yang disediakan dalam SDK Android.

2.5.4. 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 subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Beberapa fitur Android yang penting adalah:

1. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*.
2. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat mobile.

3. *Integrated browser* berdasarkan *engine open source WebKit*.
4. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselarasi harware).
5. SQLite untuk penyimpanan data.
6. Media Support yang mendukung audio, video, dan gambar (tergantung hardware).
7. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware).
8. Kamera, GPS, kompas, dan *accelerometer* (tergantung hardware).
9. Lingkungan *development* yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE eclipse.

2.5.5. Arsitektur Android

Secara garis besar arsitektur android dapat digambarkan sebagai berikut:

1. *Application* and *Widgets*

   Application and *Widgets* adalah *layer* di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita *download* aplikasi kemudian di-*install* dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain.

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 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.

Komponen-komponen yang termasuk di dalam *Application Frameworks* adalah sebagai berikut:

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

3. *Libraries*

Libraries ini 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* mencakup 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* mencakup modern *web browser* dengan *engine embeded web view*

g. *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's

4. *Android Run Time*

Layer yang membuat aplikasi android dapat dijalankan di mana dalam prosesnya mengunakan implementasi Linux. DVM merupakan mesin yang membentuk dasar kerangka aplikasi android. Di dalam *Android Run Time* ada dua bagia yaitu:

a. *Core Libraries*: aplikasi android dibangun dalam bahasa java sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine Java* sehinga diperlukan sebuah libraries yang berfungsi untuk menerjemahkan bahasa java/C yang ditangani oleh *core libraries.*

b. *Dalvik Virtual Machine*: virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, di mana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

5. *Linux Kernel*

Linux kernel adalah layer di mana inti dari sistem operasi dari android itu berada. Berisis 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.

2.5.6. Fundamental Aplikasi

Aplikasi android ditulis dalam bahasa pemrograman java. Kode java dikompilasi 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 enam jenis komponen aplikasi android, yaitu:

1. *Activities*

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa hanya memiliki satu *activity*, tetapi umumnya aplikasi memliki banyak *activity* tergantung kepada tujuan aplikasi dan desain dari aplikasi tersebut.

2. *Service*

Service adalah komponen yang bekerja secara tersembunyi di *background*. *Service* dapat dipergunakan untuk menjalankan proses yang terus berjalan meskipun *Activity* sudah tidak aktif lagi atau disembunyikan.

3. *Broadcast Receiver*

Broadcast receiver berfungsi menerima dan beraksi untuk menyampaikan notifikasi. *Broadcast receiver* tidak memiliki *user interface* tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau *vibrating* (getaran) perangkat, dan lain sebagainya.

4. Intent

Intent adalah deskripsi abstrak dari sebuah operasi yang akan dikerjakan. *Intent* dapat dipergunakan untuk menjalankan *Activity* lain, mengirim pesan ke *Broadcast Receiver*, atau berkomunikasi dengan sebuah *Service*. Umumnya, *Intent* lebih sering digunakan sebagai bagian yang menyatukan antar *Activit*y yang satu dengan yang lainnya.

5. *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti database SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta, atau aplikasi yang membutuhkan akses data kontak dan navigasi, di sinilah fungsi *content provider*.

6. *Notification*

Notification dipergunakan untuk memberitahu user mengenai sebuah event tanpa mengganggu Activity yang tengah berjalan. Sebuah notification dapat berbentuk macam-macam; baik lampu yang berkedip, suara, dan yang paling umum adalah *icon* dan *dialog message*.

2.6. *Unified Modeling Language* (UML)

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 adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat analisis dan desain berorientasi objek.

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

1. *Use Case Diagram*

   Use Case Diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem, terdiri atas diagram untuk use case dan *actor*.

   Use case merepresentasikan operasi-operasi yang dilakukan oleh *actor*. Use case digambarkan berbentuk elips dengan nama operasi ditulis di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke use case.

2. *Sequence Diagram*

   Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case, antara lain interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

3. *Activity Diagram*

   Activity diagram lebih memfokuskan diri pada eksekusi dan alur sistem daripada bagaimana sistem itu dirakit. Diagram aktivitas menunjukan aktivitas sistem dalam bentuk kumpulan aksi-aksi.

4. *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class diagram* menunjukan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

2.7. Eclipse IDE *for* Java *Developer*

Eclipse adalah sebuah *Integrated Development Environment* (IDE) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua plafrom. Eclipse saat ini merupakan salah satu IDE favorit karena gratis dan *open source*. Setiap orang boleh melihat kode pemrograman perangkat lunak ini. Kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

Secara standar Eclipse selalu dilengkapi dengan *Java Development Tools* (JDT), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan *Plug-in Development Environment* (PDE) untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka, mudah diperluas untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi bukan hanya untk mengembangkan program Java, tetapi juga dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal *plug-in* yang dibutuhkan.

2.8. Astah Community* 6.1

Astah* dahulu dikenal dengan nama JUDE (*Java and UML Developer's Environment*) adalah alat permodelan UML yang diciptakan oleh perusahaan Jepang *Change Vision*. Astah * bersifat *multi-platform* atau dapat dipakai pada berbagai sistem operasi/*platform*. Astah* dapat menggambarkan permodelan dari bahasa pemrograman Java, C++ , dan C#.

2.9. Pengujian Sistem (Perangkat Lunak)

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan spesifikasi, desain, dan pengkodean. Salah satu macam pendekatan dalam pengujian perangkat lunak adalah *black box testing*. Pendekatan ini bertujuan untuk menunjukkan fungsi perangkat lunak tetang cara beroperasinya, apakah *input* dan *output* telah berjalah sebagaimana diharapkan.

Untuk jenis pengujian dari segi pelaku, salah satunya yaitu pengujian *alpha*, yaitu pengujian yang dilakukan pada sisi pengembang. Perangkat lunak digunakan pada *setting* yang natural dengan pengembang yang memandang melalui bahu pemakai dan merekam semua kesalahan dan masalah pemakaian.