

## **BAB II**

### **LANDASAN TEORI**

#### **II.1 Konsep Dasar Sistem**

##### **II.1.1 Pengertian Sistem**

Pada bab ini akan dibahas mengenai definisi sistem informasi, namun harus diketahui terlebih dahulu konsep sistem informasi. Dari definisi sistem dan informasi memberikan gambaran mengenai perbedaan antara sistem dan informasi. Definisi tersebut akan membentuk suatu pengetahuan tentang konsep dasar sistem informasi.[1]

##### **II.1.2 Elemen-Elemen Sistem**

Sistem mempunyai elemen-elemen yang mendukung sebuah sistem itu sendiri berupa:

1. Tujuan Sistem

Tujuan sistem merupakan tujuan dari sistem tersebut disebut. Tujuan sistem dapat berguna tujuan organisasi, kebutuhan organisasi, permasalahan yang ada dalam suatu organisasi maupun urutan prosedur untuk mencapai tujuan organisasi.

2. Batasan Sistem

Batasan sistem merupakan sesuatu yang membatasi sistem dalam mencapai tujuan sistem. Batasan sistem dapat berupa peraturan-peraturan yang ada dalam suatu organisasi, biaya-biaya yang dikeluarkan, orang-orang yang ada dalam organisasi, fasilitas baik itu sarana dan prasarana maupun batasan yang lain.

3. Kontrol Sistem

Kontrol atau pengawasan sistem merupakan pengawasan terhadap pelaksanaan pencapaian tujuan dari sistem tersebut. Kontrol sistem dapat berupa kontrol terhadap pemasukan data (*input*), kontrol terhadap keluaran data (*output*), kontrol terhadap pengolahan data, kontrol terhadap umpan balik dan sebagainya.

4. Input

Input merupakan elemen dari sistem yang bertugas untuk menerimaseluruh masukan data, dimana masukan tersebut dapat berupa jenisdata, frekuensi pemasukan data dan sebagainya.

#### 5. Proses

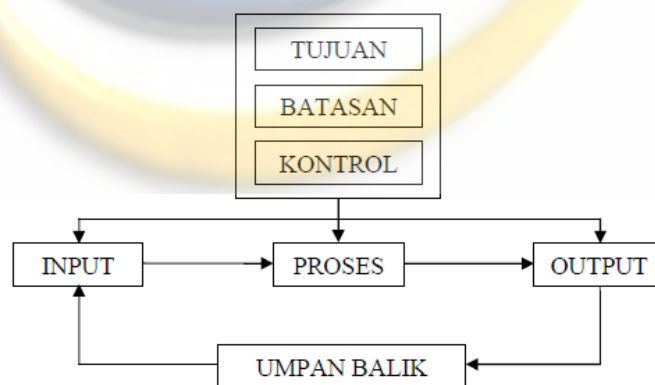
Proses merupakan elemen dari sistem yang bertugas untuk mengolahatau memproses seluruh masukan data menjadi suatu yang lebihberguna. Misalkan sistem produksi akan mengolah bahan baku yangberupa bahan mentah menjadi bahan jadi yang siap untuk digunakan.

#### 6. Output

Output merupakan hasil dari inout yang telah diproses oleh bagianpengolah dan merupakan tuijuan akhir sistem.Output ini bisa berupalaporan grafik, diagram batang dan sebagainya.

#### 7. Umpan Balik

Umpan balik merupakan elemen dalam sistem yang bertugasmengevaluasi bagian dari output yang dikeluarkan, dimana elemenini sangat penting demi kemajuan sebuah sistem. Umpan balik inidapat merupakan perbaikan sistem, pemeliharaan sistem dansebagainya.



Gambar II.1 – Siklus Informasi<sup>[4]</sup>

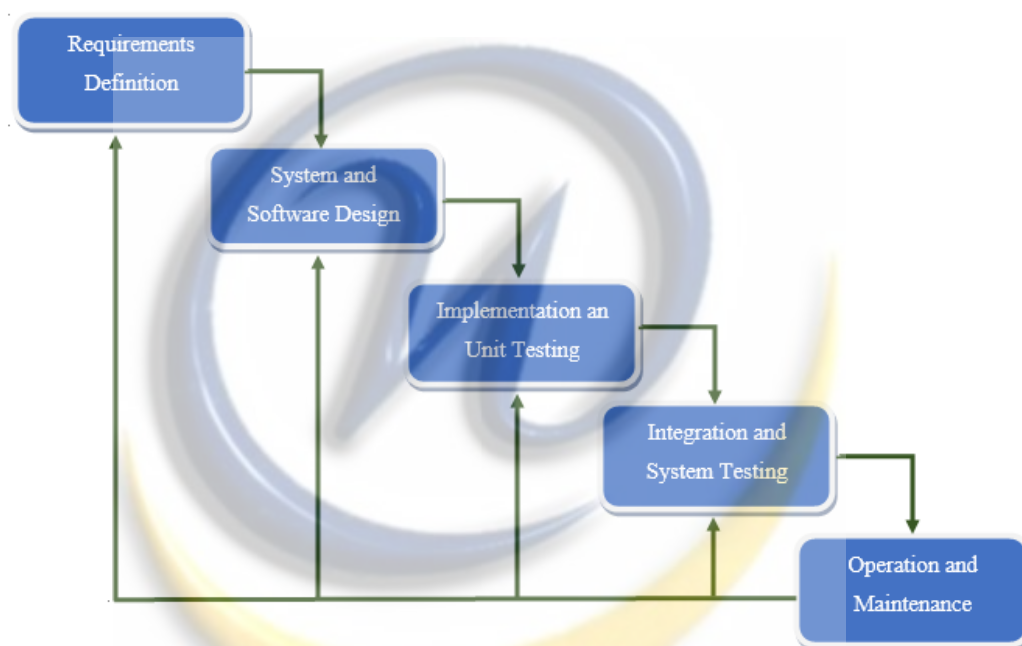
## II.2 Konsep Dasar Perancangan

### II.2.1 *Waterfall Model*

*Waterfall Model* merupakan dasar dari aktivitas proses yang terdiri dari spesifikasi, pengembangan, validasi, evolusi dan semua direpresentasikan

dalam tahapan proses yang terpisah seperti spesifikasi kebutuhan, perancangan perangkat lunak, implementasi, pengujian, dan sebagainya.

Model dari *software development process* ini adalah model yang pertama kali dipublikasikan yang diperoleh dari *system engineering process* yang umum. Karena satu tahap ke tahap lainnya mengalir ke bawah, model ini disebut sebagai *Waterfall Model*. *Waterfall Model* ini adalah contoh dari sebuah proses yang bersifat *plan-driven* - secara prinsip, semua aktivitas proses harus direncanakan dan diproses terlebih dahulu sebelum mulai mengerjakannya[2].



**Gambar II.2 – Waterfall Model<sup>[5]</sup>**

Tahapan dari *Waterfall Model* merefleksikan pokok-pokok dari aktivitas pengembangan<sup>[5]</sup>:

1. *Requirements Analysis and Definition*

Layanan yang diberikan oleh sistem, batasan sistem, dan tujuan ditetapkan setelah melakukan konsultasi dengan pengguna sistem. Semua didefinisikan secara rinci dan dibuat sebagai spesifikasi dari sistem.

2. *System and Software Design*

Proses perancangan sistem menyediakan kebutuhan hardware atau software dengan menyediakan arsitektur dari keseluruhan sistem. Perancangan sistem melibatkan pengidentifikasian dan penjelasan dari abstraksi sistem dan hubungannya.

### 3. *Implementation and Unit Testing*

Pada tahap ini, perancangan sistem direalisasikan menjadi sebuah program atau unit program. Pengujian unit melibatkan verifikasi untuk memastikan apakah setiap unit memenuhi spesifikasi sistem.

### 4. *Integration and System Testing*

Setiap unit program dan program-program yang sudah ada diintegrasikan dan diuji sebagai satu keutuhan sistem untuk memastikan apakah kebutuhan sistem sudah terpenuhi. Setelah melakukan pengujian, sistem baru disebarkan ke pengguna.

### 5. *Operation and Maintenance*

Dilakukan instalasi terhadap sistem dan digunakan dalam prakteknya. *Maintenance* melibatkan koreksi terhadap *error* yang tidak ditemukan pada tahap sebelumnya, memperbaiki implementasi dari unit sistem dan meningkatkan layanan yang diberikan oleh sistem sebagai kebutuhan baru yang ditemukan.

*Waterfall model* digunakan hanya jika semua kebutuhan sudah dimengerti dan tidak berubah secara radikal pada tahap pengembangan<sup>[5]</sup>.

## II.2.2 *Object Oriented Programming (OOP)*

*Object Oriented Programming* adalah paradigma pemrograman yang memandang perangkat lunak sebagai kumpulan objek yang saling berinteraksi di dalam suatu sistem<sup>[2]</sup>. Beberapa objek berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Masing-masing objek harus berisikan informasi mengenai dirinya sendiri (*encapsulation*) dan objek yang dapat dikaitkan (*inheritance*).

Dalam OOP, *Class* merupakan sekumpulan objek yang memiliki atribut-atribut dan *method*<sup>[10]</sup>. *Class* merupakan deskripsi dari satu atau lebih objek yang memiliki kesamaan atribut, layanan, metode, hubungan, dan semantik, termasuk deskripsi cara membuat objek baru dalam *class*. Ada juga yang disebut dengan *super class*, sebuah *class* induk yang nantinya mempunyai *class-class* yang terdiri dari *class* dan *subclass*<sup>[10]</sup>.

Objek dalam OOP adalah sebuah benda atau unit atau sifat kerja yang memiliki atribut-atribut<sup>[10]</sup>. Objek adalah sebuah abstraksi dari sesuatu pada domain masalah, menggambarkan kemampuan untuk menyimpan informasi mengenai hal tersebut, berinteraksi dengan hal tersebut atau keduanya.

Abstraksi prosedural dalam OOP disebut dengan operasi, yang menspesifikasi tipe dari perilaku dan terdiri dari fungsi-fungsi<sup>[10]</sup>. Istilah lain terdapat *encapsulation*/pengkapsulan, yang merupakan pembatasan ruang lingkup program terhadap data yang diproses supaya data terlindungi oleh prosedur atau objek lain, kecuali prosedur yang berada di objek itu sendiri<sup>[10]</sup>.

*Polymorphism* adalah konsep yang menyatakan bahwa sesuatu yang samadapat mempunyai bentuk dan perilaku yang berbeda, bahwa operasi yang sama mungkin memiliki perbedaan dalam *class* yang berbeda<sup>[10]</sup>.

Pada OOP, terdapat juga yang disebut dengan *inheritance* (pewarisan), yaitu kepemilikan yang bersifat *implicit* dari fitur *subclass* yang didefinisikan dalam *superclass*. Fitur tersebut mencakup *variables* dan *method*<sup>[10]</sup>.

Konsep dasar dari Pemrograman Berorientasi Objek Pemrograman orientasi-objek menekankan konsep berikut:

- a. Kelas - kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah *class* adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. *Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang*

terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

- b. Objek - membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
- c. Abstraksi - Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
- d. Enkapsulasi - Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
- e. Polimorfisme melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tunggal dalam program dapat memegang



berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

- f. Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

### II.3 Unified Modelling Language (UML)

UML adalah bahasa standar pemodelan untuk perangkat lunak dan pengembangan sistem.[2]

Merancang sebuah desain untuk sistem yang besar merupakan hal yang sulit. Dari aplikasi *desktop* yang sederhana sampai sistem *multi-tier* dapat dibangun dari ratusan, bahkan ribuan, komponen perangkat lunak dan perangkat keras. Untuk melacak komponen apa yang dibutuhkan, apa yang dikerjakan, bagaimana mencapai kebutuhan pelanggan, dan bagaimana menjelaskan sistem ke tim kerja lainnya dibutuhkan sebuah pemodelan, ini adalah dimana UML dibutuhkan.

Dalam merancang sebuah sistem, mengatur kompleksitas adalah salah satu alasan utama mengapa harus membuat model. Pemodelan membantu para pengembang untuk dapat fokus, dapat mendokumentasikan, menangkap keseluruhan sistem dan mengkomunikasikan aspek-aspek penting dalam sistem

yang sedang dirancang.

UML tepat digunakan untuk memodelkan sistem dari mulai memodelkan informasi sistem untuk perusahaan hingga aplikasi *web*, bahkan untuk sistem yang rumit sekalipun. UML menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk perancangan aplikasi Berbasis PHP.

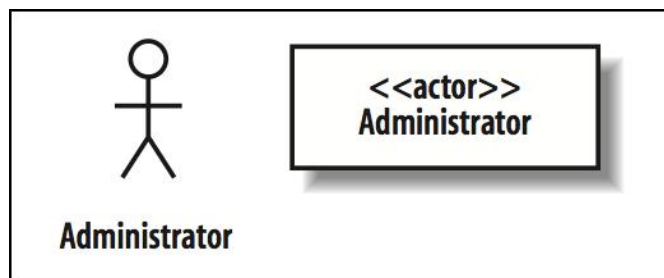
UML mendefinisikan diagram-diaagram, antara lain:

1. *Use case diagram*
2. *Activity diagram*
3. *Class diagram*
4. *Sequence diagram*
5. *Collaboration diagram*
6. *Statechart diagram*
7. *Component diagram*
8. *Deployment diagram*
9. *Package diagram*

### II.3.1 Use Case Diagram

*Use Case diagram* menunjukkan bagaimana sistem yang sedang dibangun digunakan untuk memenuhi kebutuhan pengguna. Sebuah *usecase* menampilkan sebuah fungsionalitas yang diberikan oleh sistem<sup>[5]</sup>. Yang ditekankan pada *use case* adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”.

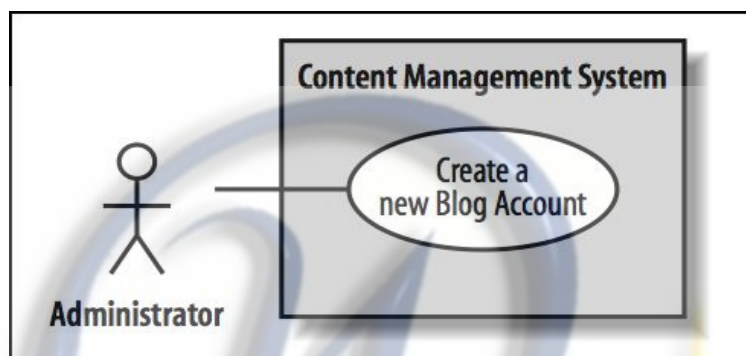
Diluar sistem terdapat aktor yang digambarkan dengan istilah “*stick man*” atau sebuah box. Seperti yang terdapat pada Gambar II.5, dimana terdapat dua pilihan untuk menggambarkan seorang aktor.



Gambar II.3 – Notasi Actor Pada Use Case<sup>[5]</sup>



Selain aktor, terdapat sebuah *use case*, *communication line*, dan *system boundaries* untuk menggambarkan *use case diagram* secara utuh. Sebuah *use case* menampilkan bagaimana sistem digunakan untuk menyelesaikan pekerjaan aktor. *Communication line* menghubungkan aktor dan *use case* untuk menunjukkan bahwa aktor tersebut berpartisipasi di dalam *use case*. Sedangkan *system boundaries* digunakan untuk menandakan pemisahan antara eksternal sistem (aktor) dan internal sistem (*use cases*). Berikut ini adalah gambar dari gabungan antara aktor, *use case*, *communication line* dan *system boundaries*<sup>[5]</sup>.



Gambar II.4 - Contoh *Use Case*<sup>[5]</sup>

*Use case diagram* yang menunjukkan *use cases* dan aktor merupakan awalan yang baik, namun diagram tersebut tidak menampilkan rincian yang cukup untuk mengerti sistem. Cara yang paling baik untuk mengekspresikan informasi tersebut adalah dalam bentuk tulisan yang disebut dengan *Use Case Description*. Gambar-gambar berikut mencontohkan sebuah *Use Case Description* berdasarkan contoh *use case* sebelumnya :

Use case name	Create a new Blog Account
Related Requirements	Requirement A.1.
Goal In Context	A new or existing author requests a new blog account from the Administrator.
Preconditions	The system is limited to recognized authors and so the author needs to have appropriate proof of identity.
Successful End Condition	A new blog account is created for the author.
Failed End Condition	The application for a new blog account is rejected.
Primary Actors	Administrator.
Secondary Actors	Author Credentials Database.
Trigger	The Administrator asks the CMS to create a new blog account.

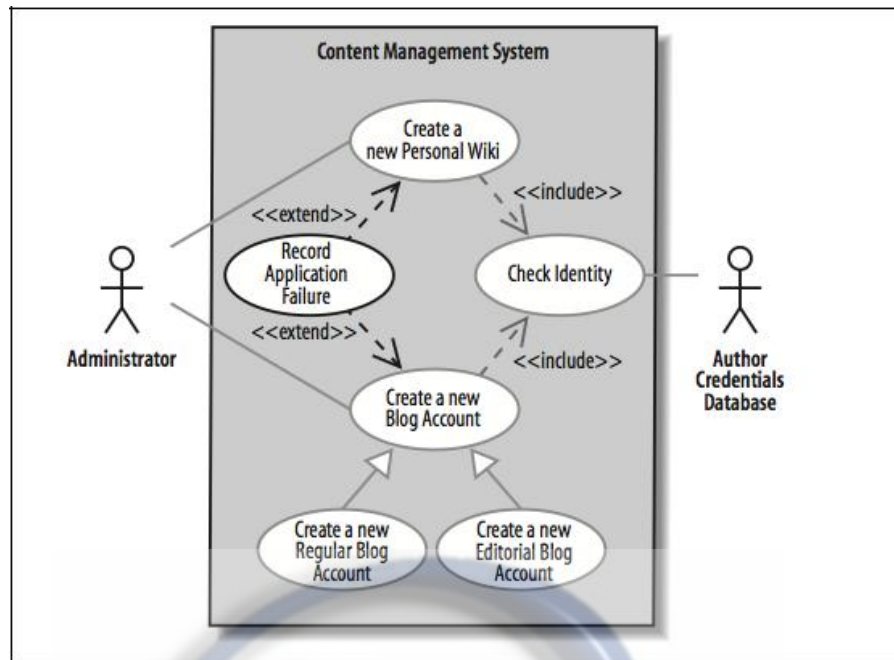
Gambar II.5 - Contoh *Use Case Description*<sup>[5]</sup>

Use case name	Create a new Blog Account	
Main Flow	<b>Step</b>	<b>Action</b>
	1	The Administrator asks the system to create a new blog account.
	2	The Administrator selects an account type.
	3	The Administrator enters the author's details.
	4	The author's details are verified using the Author Credentials Database.
	5	The new blog account is created.
Extensions	6	A summary of the new blog account's details are emailed to the author.
	<b>Step</b>	<b>Branching Action</b>
	4.1	The Author Credentials Database does not verify the author's details.
	4.2	The author's new blog account application is rejected.

Gambar II.6 - Contoh USe Case Description (Lanjutan)<sup>[5]</sup>

Hubungan antar *use case* dapat diekspresikan menggunakan `<<include>>` dan `<<extend>>`. Relasi `<<include>>` digunakan untuk memecahkan sistem menjadi bagian-bagian yang lebih mudah untuk diatur. Sedangkan relasi `<<extend>>` adalah sebuah cara untuk menunjukkan bahwa sebuah *use case* mungkin dapat sepenuhnya menggunakan *behavior* dari *use case* lain. Selain itu *use case diagram* juga dapat menggunakan generalisasi untuk beberapa kasus yang berbeda, misalnya terdapat beberapa tipe dalam membangun sebuah *blog*. *Use case* “*Create a new Blog Account*” bisa digeneralisasikan menjadi dua tipe, yaitu “*Regular Blog Account*” atau “*Editorial Blog Account*”<sup>[5]</sup>.

Berikut adalah contoh *use case diagram* secara utuh:

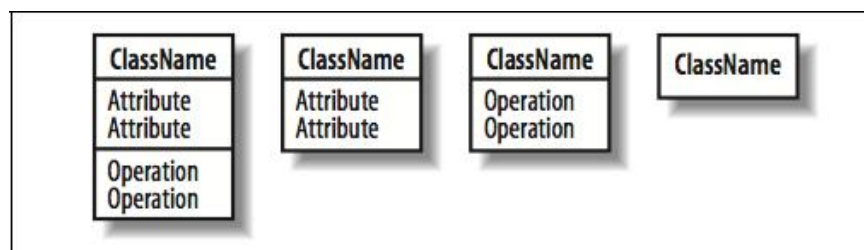
Gambar II.7 - Diagram Use Case<sup>[5]</sup>

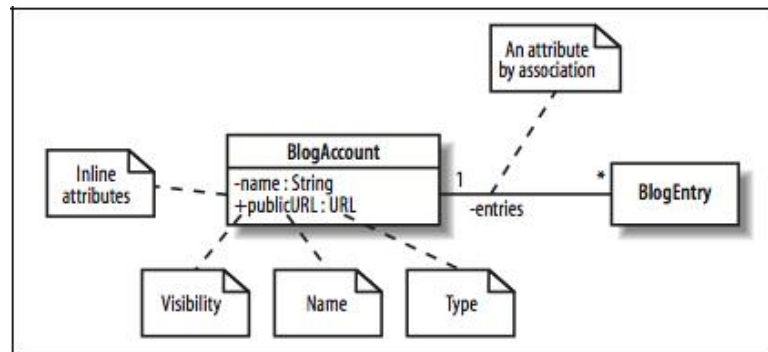
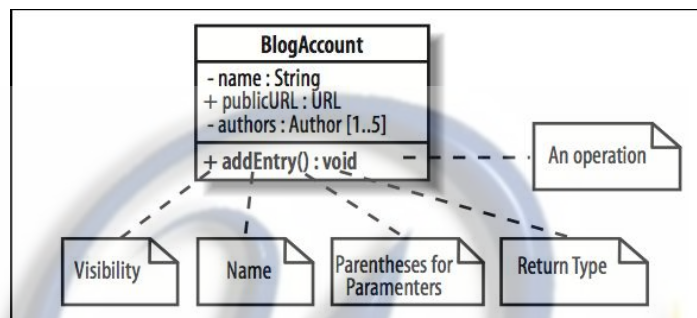
### II.3.2 Class Diagram

*Class* bisa disebut sebagai bagan yang digunakan untuk membangun dan mendefinisikan objek-objek. Sehingga setiap hal yang dibangun dari sebuah *class* disebut sebagai objek atau *instances*.

Deskripsi dari sebuah *class* mencakup dua bagian: informasi yang akan dimiliki oleh objek dan *behaviour* yang akan didukung objek tersebut. Ini yang membedakan *object-oriented* dengan yang lainnya, bahwa *state information* dan *behaviour* digabungkan dalam satu definisi *class*, dimana objek-objek dapat terbentuk dari *class* tersebut<sup>[5]</sup>.

*Class* dalam UML digambarkan sebagai persegi panjang dibagi menjadi tiga bagian. Bagian paling atas berisi nama *class*, bagian tengah berisi atribut atau informasi yang dimiliki oleh *class* tersebut, dan bagian akhir berisi operasi yang menunjukkan *behaviour* dari *class*<sup>[5]</sup>.

Gambar II.8 - Class Diagram<sup>[5]</sup>

Gambar II.9 - Class State : Attribute<sup>[5]</sup>Gambar II.10 - Class Behaviour : Operation<sup>[5]</sup>

*Class* mempunyai sifat *visibility*, sebuah sifat yang dapat digunakan untuk mengatur akses ke atribut dan metode atau *operation*, yaitu :

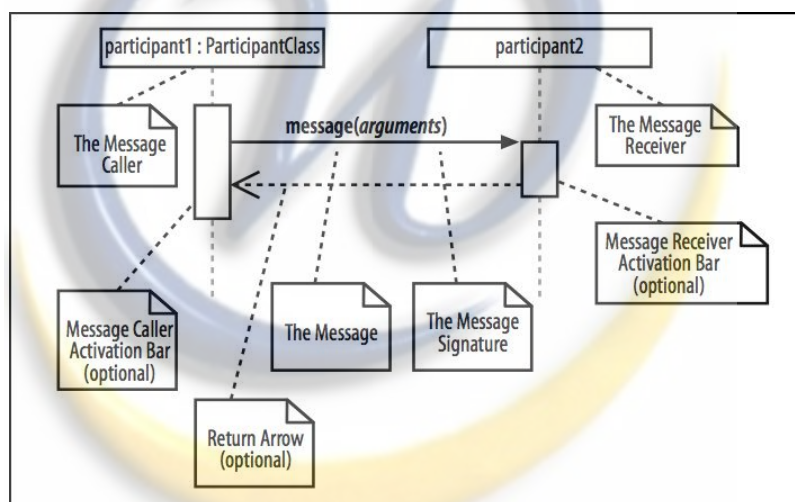
1. *Private*, hanya dapat digunakan oleh metode pada *class* tersebut, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan (metode) dan anak-anak yang mewarisinya (*generalization*).
3. *Public*, dapat dipanggil oleh siapa saja.
4. *Package*, dapat dipanggil oleh *class* yang bersangkutan (metode) dan *class* lain yang berada di dalam *package* yang sama.

### II.3.3 Sequence Diagram

*Sequence diagram* memodelkan bagaimana sistem yang sedang dibangun mendatangi tugasnya. Diagram ini menangkap semua urutan interaksi pada bagian-bagian dari sistem. Dengan menggunakan *sequence diagram*, pengembang bisa menjelaskan interaksi apa yang akan dipanggil ketika sebuah *use case* dieksekusi dan dalam urutan seperti apa interaksi-interaksi tersebut akan terjadi<sup>[5]</sup>.

Dalam *sequence diagram* terdapat bagian-bagian yang disebut dengan

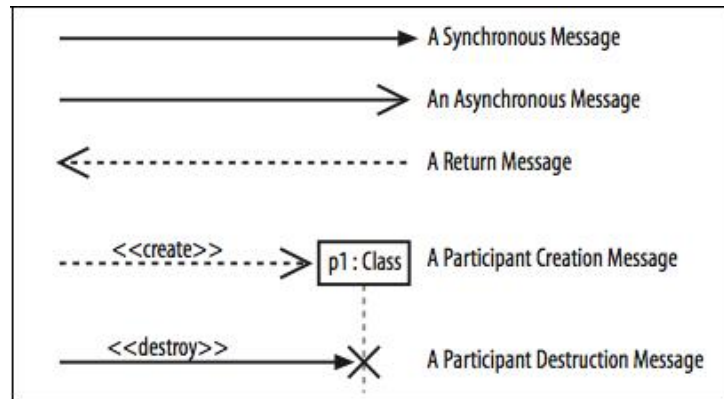
*participant*, *time*, *message*, dan *activation bars*. *Participant* adalah bagian-bagian dari sistem yang berinteraksi dengan satu sama lain selama *sequence* berjalan. *Time* menunjukkan urutan dimana semua interaksi berlangsung sesuai dengan waktu. *Time* pada *sequencediagram* ditunjukkan dengan garis titik-titik vertikal. *Message* dalam *sequence* ditunjukkan menggunakan panah dari *participant* yang ingin mengirimkan pesan ke *participant* lain. *Message Caller* adalah sebutan untuk *participant* yang mengirim *message* dan *Message Receiver* untuk *participant* yang menerima *message*. *Activation bars* menunjukkan bahwa sebuah *participant* sedang melakukan sesuatu untuk jangka waktu tertentu. *Activation bars* dapat ditampilkan pada awal pengiriman *message* dan ujung dari penerimaan *message*. Hal ini menunjukkan bahwa *Message Caller* sibuk pada saat mengirim *message* dan *MessageReceiver* sibuk pada saat *message* sudah diterima<sup>[5]</sup>.



Gambar II.11 - Contoh Interaksi Pada *Sequence Diagram*<sup>[5]</sup>

Dan berikut ini adalah bentuk-bentuk dari panah pada *sequencediagram*, setiap bentuk panah mempunyai arti yang berbeda.

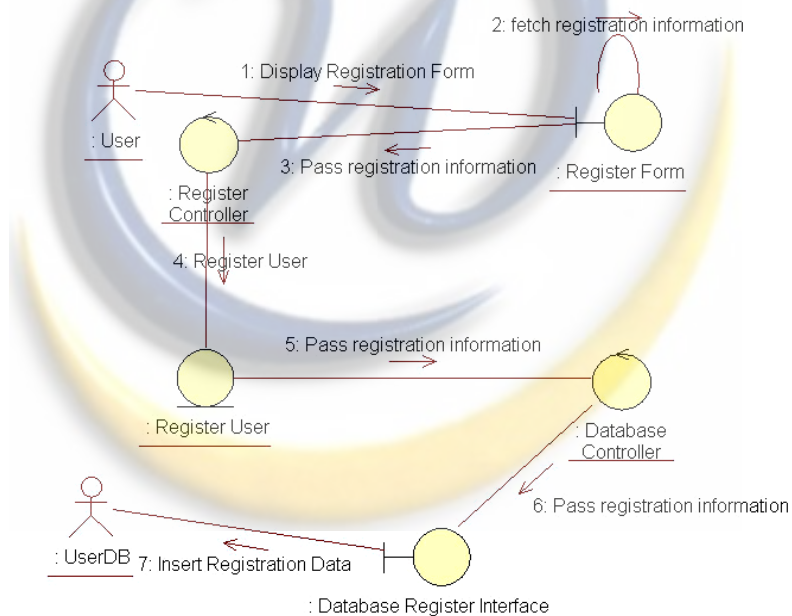




**Gambar II.12 - Bentuk - Bentuk Panah Pada *Sequence Diagram***<sup>[5]</sup>

### II.3.4 *Collaboration Diagram*

Mendefinisikan peran-peran yang dimainkan ketika satu tugas dilakukan. Peran-peran dimainkan oleh instant-instant yang berinteraksi, dan menyatakan hubungan komunikasi antara objek-objek yang menunjuknya.

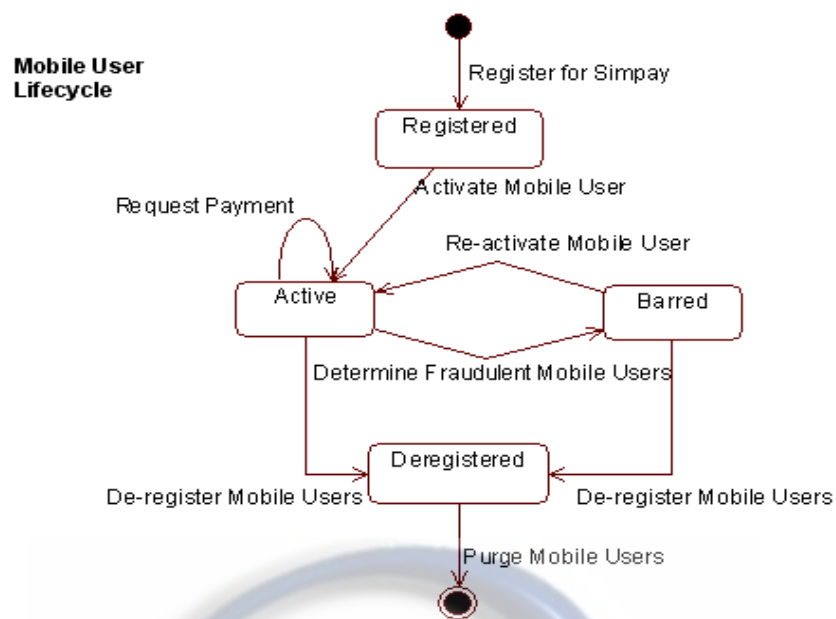


**Gambar II.13 - Contoh *Collaboration Diagram***<sup>[5]</sup>

### II.3.5 *Statechart Diagram*

Merupakan perluasan dari *diagram state* yang mendeskripsikan objek berupa *state-state* yang dimiliki, kejadian-kejadian yang berlangsung beserta transisi yang terjadi.

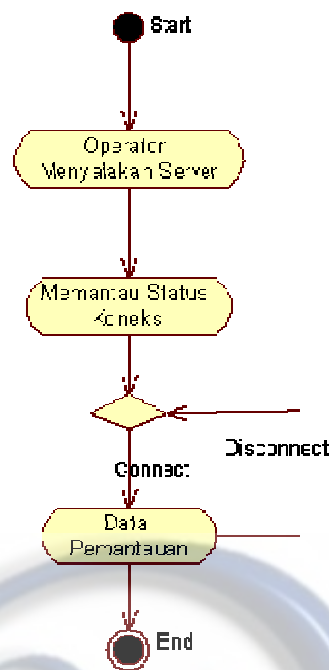




Gambar II.14 - Contoh *Statechart Diagram*<sup>[5]</sup>

### II.3.6 Activity Diagram

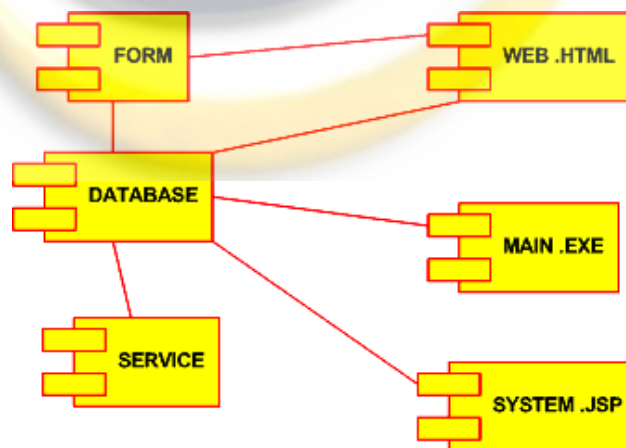
Adalah Representasi secara grafis dari proses dan *control flow* dan berfungsi untuk memperlihatkan alur dari satu aktivitas ke aktivitas yang lain serta menggambarkan perilaku yang kompleks.



Gambar II.15 - Contoh *Activity Diagram*<sup>[5]</sup>

### II.3.7 Component Diagram

Adalah *Diagram* yang menunjukkan organisasi dan kebergantungan di antara sekumpulan komponen. Diagram ini memodelkan pandangan implementasi fisik dari sistem.

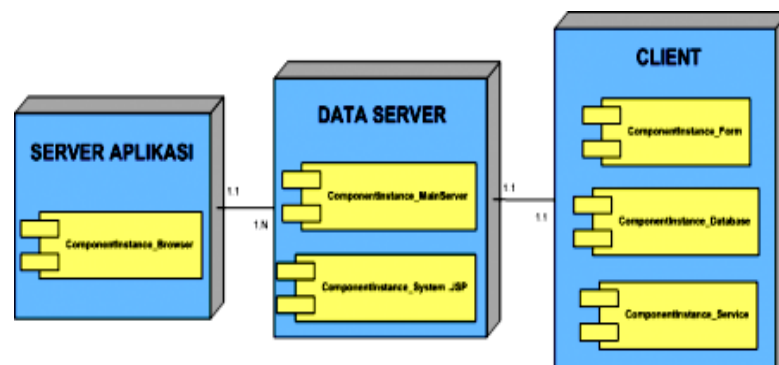


Gambar II.16 - Contoh *Component Diagram*<sup>[5]</sup>

### II.3.8 Deployment Diagram

Adalah penggambaran tugas-tugas kongkrit dari setiap *node/software* yang terlibat dalam jaringan sistem, menampilkan keseluruhan *node* dalam jaringan

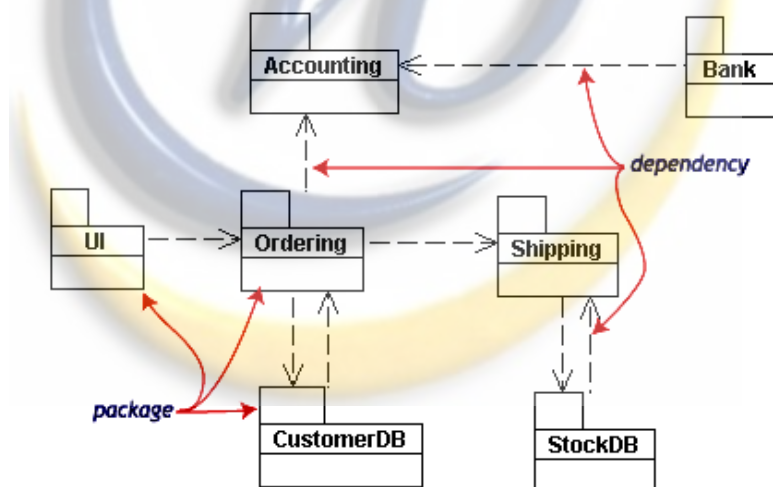
serta hubungan dari *node-node* tersebut termasuk proses-proses yang terlibat di dalamnya.



Gambar II.17 - Contoh *Deployment Diagram*<sup>[5]</sup>

### II.3.9 Package Diagram

Adalah Penggambaran dan pengelompokan kelas-kelas yang terdapat pada perangkat lunak yang dibangun sesuai dengan fungsi atau subsistem aplikasi yang mempunyai ketergantungan satu sama lainnya.



Gambar II.18 - Contoh *Package Diagram*<sup>[5]</sup>

## II.4 Android

### II.4.1 Pengertian Android

Android adalah Sistem Operasi berbasis linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan computer tablet. Android awalnya dikembangkan oleh Android Inc, dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. System operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance*,

konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Antarmuka pengguna Android didasarkan pada manipulasi langsung. Menggunakan masukan sentuh yang serupa dengan tindakan dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikan cubitan untuk memanipulasi obyek di layar. Android adalah system operasi dengan sumber terbuka, dan Google merilis kodenya dibawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada android memungkinkan perangkat lunak untuk memodifikasi secara bebas dan mendistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, android memiliki sejumlah besar komunitas pengembang aplikasi yang memperluas fungsionalitas perangkat, umumnya ditulis dengan versi kostumisasi bahasa pemrograman java. Pada bulan oktober 2012, ada sekitar 700.000 aplikasi yang tersedia untuk android, dan sekitar 25 juta aplikasi telah diunduh dari *google play*, toko aplikasi utama android. Sebuah survey pada bulan April – Mei 2013 menemukan bahwa Android adalah *platform* paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi seluler.

Faktor – faktor di atas telah memberikan kontribusi terhadap perkembangan Android, menjadikanya sebagai system operasi telepon pintar yang paling banyak digunakan di dunia, mengalahkan Symbian pada tahun 2010, Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan system operasi berbiaya rendah, bisa di kostumisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkanya dari awal. Akibatnya, meskipun pada awalnya system operasi ini dirancang khusus untuk telepon pintar dan tablet, android juga dikembangkan menjadi aplikasi tambahan di televisi, konsol permainan, kamera digital, dan perangkat elektronik lainnya. Sifat android yang terbuka telah mendorong munculnya sejumlah besar pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur fitur baru bagi pengguna tingkat lanjut atau mengoperasikan android pada perangkat yang secara resmi dirilis dengan menggunakan system operasi lain.

Pada November 2013, Android menguasai pangsa pasar telepon pintar global, yang dipimpin oleh produk Samsung, dengan presentase 64% pada bulan maret 2013, pada juli 2013, terdapat 11.868 perangkat android berbeda dengan beragam versi. Keberhasilan system operasi ini juga menjadikanya sebagai target litigasi paten “perang telepon pintar” antara perusahaan – perusahaan teknologi. Hingga bulan mei 2013, total 900 juta perangkat android telah diaktifkan di seluruh dunia, dan 28 miliar aplikasi telah dipasang dari Google play. Pada tanggal 3 september 2013, 1 miliar perangkat android telah diaktifkan.[3]

#### II.4.2 Sejarah Perkembangan Android

Android, Inc. didirikan di Palo Alto, California, pada bulan Oktober 2003 oleh Andy Rubin (pendiri Danger),<sup>[30]</sup> Rich Miner (pendiri Wildfire Communications, Inc.),<sup>[31]</sup> Nick Sears<sup>[32]</sup> (mantan VP T-Mobile), dan Chris White (kepala desain dan pengembangan antarmuka WebTV)<sup>[12]</sup> untuk mengembangkan "perangkat seluler pintar yang lebih sadar akan lokasi dan preferensi penggunanya".<sup>[12]</sup> Tujuan awal pengembangan Android adalah untuk mengembangkan sebuah sistem operasi canggih yang diperuntukkan bagi kamera digital, namun kemudian disadari bahwa pasar untuk perangkat tersebut tidak cukup besar, dan pengembangan Android lalu dialihkan bagi pasar telepon pintar untuk menyaingi Symbian dan Windows Mobile (iPhone Apple belum dirilis pada saat itu).<sup>[33]</sup> Meskipun para pengembang Android adalah pakar-pakar teknologi yang berpengalaman, Android Inc. dioperasikan secara diam-diam, hanya diungkapkan bahwa para pengembang sedang menciptakan sebuah perangkat lunak yang diperuntukkan bagi telepon seluler.<sup>[12]</sup> Masih pada tahun yang sama, Rubin kehabisan uang. Steve Perlman, seorang teman dekat Rubin, meminjaminya \$10.000 tunai dan menolak tawaran saham di perusahaan.<sup>[34]</sup>

Google mengakuisisi Android Inc. pada tanggal 17 Agustus 2005, menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google. Pendiri Android Inc. seperti Rubin, Miner dan White tetap bekerja di perusahaan setelah diakuisisi oleh Google.<sup>[12]</sup> Setelah itu, tidak banyak yang diketahui tentang perkembangan Android Inc., namun banyak anggapan yang menyatakan bahwa Google berencana untuk memasuki pasar telepon seluler dengan tindakannya ini.<sup>[12]</sup> Di Google, tim yang dipimpin oleh Rubin mulai mengembangkan platform perangkat seluler dengan menggunakan kernel Linux. Google memasarkan platform tersebut kepada produsen perangkat seluler dan operator nirkabel, dengan janji bahwa mereka menyediakan sistem yang fleksibel dan bisa diperbarui. Google telah memilih beberapa mitra perusahaan perangkat lunak dan perangkat keras, serta mengisyaratkan kepada operator seluler bahwa kerjasama ini terbuka bagi siapapun yang ingin berpartisipasi.

Spekulasi tentang niat Google untuk memasuki pasar komunikasi seluler terus berkembang hingga bulan Desember 2006.<sup>[38]</sup> BBC dan *Wall Street Journal* melaporkan bahwa Google sedang bekerja keras untuk menyertakan aplikasi dan mesin pencari ke perangkat seluler. Berbagai media cetak dan media daring mengabarkan bahwa Google sedang mengembangkan perangkat seluler dengan merek Google. Beberapa di antaranya berspekulasi bahwa Google telah menentukan spesifikasi teknisnya, termasuk produsen telepon seluler dan operator jaringan. Pada bulan Desember 2007, *InformationWeek* melaporkan bahwa Google telah mengajukan beberapa aplikasi paten di bidang telepon seluler.

Pada tanggal 5 November 2007, Open Handset Alliance (OHA) didirikan. OHA adalah konsorsium dari perusahaan-perusahaan teknologi seperti Google, produsen perangkat seluler seperti HTC, Sony dan Samsung, operator nirkabel seperti Sprint, Nextel dan T-Mobile, serta produsen chipset seperti Qualcomm dan Texas Instruments. OHA sendiri bertujuan untuk mengembangkan standar terbuka bagi perangkat seluler.<sup>[13]</sup> Saat itu, Android diresmikan sebagai produk pertamanya; sebuah platform perangkat seluler yang menggunakan kernel Linux versi 2.6.<sup>[13]</sup> Telepon seluler komersial pertama yang menggunakan sistem operasi Android adalah HTC Dream, yang diluncurkan pada 22 Oktober 2008.

Pada tahun 2010, Google merilis seri Nexus; perangkat telepon pintar dan tablet dengan sistem operasi Android yang diproduksi oleh mitra produsen telepon seluler seperti HTC, LG, dan Samsung. HTC bekerjasama dengan Google dalam merilis produk telepon pintar Nexus pertama, yakni Nexus One.<sup>[42]</sup> Seri ini telah diperbarui dengan perangkat yang lebih baru, misalnya telepon pintar Nexus 4 dan tablet Nexus 10 yang diproduksi oleh LG dan Samsung.<sup>[43]</sup> Pada 13 Maret 2013, Larry Page mengumumkan dalam postingan blognya bahwa Andy Rubin telah pindah dari divisi Android untuk mengerjakan proyek-proyek baru di Google.<sup>[44]</sup> Ia digantikan oleh Sundar Pichai, yang sebelumnya menjabat sebagai kepala divisi Google Chrome, yang mengembangkan Chrome OS.



Sejak tahun 2008, Android secara bertahap telah melakukan sejumlah pembaruan untuk meningkatkan kinerja sistem operasi, menambahkan fitur baru, dan memperbaiki bug yang terdapat pada versi sebelumnya. Setiap versi utama yang dirilis dinamakan secara alfabetis berdasarkan nama-nama makanan pencuci mulut atau cemilan bergula; misalnya, versi 1.5 bernama *Cupcake*, yang kemudian diikuti oleh versi 1.6 *Donut*. Versi terbaru adalah 4.4 *KitKat*, yang dirilis pada 31 Oktober 2013.

## **II.5 Mobile Learning**

### **II.5.1 Pengertian Mobile Learning**

Mobile-Learning didefinisikan oleh Clark Quinn (2000) sebagai :The intersection of mobile computing and e-learning : accesible recources wherever you are, strong search capabilities, rich interaction, powerfull support for effective learnig, and peformance-based assement E-learnig independent of location in time space.

Menunjuk dari definisi tersebut maka, mobile learning adalah model pembelajaran yang memanfaatkan teknologi informasi dan komunikasi. Pada konsep pembelajaran tersebut mobile learning membawa mamfaat ketersediaan materi ajar yang dapat diakses setiap saat dan visualisasi materi yang menarik. Hal penting yang perlu diperhatikan bahwa setiap materi pengajaran cocok memanfaatkan mobile learning.

Stevanus Wisnu Wijaya (2006) menjelaskan bahwa materi ajar yang tidak cocok mengadopsi konsep mobile learning antara lain : materi yang bersifat hands on, keterampilan sebagaimana dokter gigi, seni musik khususnya mencipta lagu, interview skills, team work seperti marketing maupun materi yang membutuhkan pengungkapan ekspresi seperti tarian. Mempertimbangkan hal-hal tersebut di atas maka penerapan mobile learning lebih baik pada jenjang pendidikan tinggi.

Ally (2004) mendefinisikan mobile learning merupakan penyampaian bahan pembelajaran elektronik pada alat komputasi mobile agar dapat diakses dari mana saja dan kapan saja. Pada umumnya, perangkat mobile berupa telepon genggam digital dan PDA. Namun, secara lebih umum dapat didefinisikan sebagai

perangkat apapun yang berukuran cukup kecil, dapat bekerja sendiri, dapat dibawa setiap waktu dalam kehidupan sehari-hari dan dapat digunakan untuk beberapa bentuk pembelajaran. Perangkat kecil ini dapat dilihat sebagai alat untuk mengakses konten, baik disimpan secara lokal pada device maupun dapat dijangkau melalui interkoneksi. Perangkat ini juga dapat menjadi alat untuk berinteraksi dengan orang lain, baik melalui suara, maupun saling bertukar pesan tertulis, gambar diam dan gambar bergerak.[4]

### **II.5.2 Sejarah Singkat *Mobile Learning***

#### **Pada tahun 1970**

Diperdebatkan contoh pertama dari mobile learning kembali sejauh 1901 ketika Linguaphone merilis serangkaian pelajaran bahasa pada silinder lilin . Hal ini diikuti dalam tahun kemudian sebagai teknologi ditingkatkan, untuk menutupi kompak kaset , 8 track tape , dan

#### **Pada Tahun 1970, 1980**

Pada tahun 1968 Alan Kay dan rekan-rekannya di Kelompok Belajar Penelitian di Xerox Palo Alto Research Center [PARC] mengusulkan Dynabook sebagai komputer buku berukuran untuk menjalankan simulasi dinamis untuk belajar. "The Dynabook akan membawa sebuah ensiklopedia informasi di dalam sirkuit dan plug ke jaringan tersedia mengandung jumlah pengetahuan manusia. Siapa saja dari sekolah dasar sampai akan dapat program untuk melakukan apa pun yang diinginkan. Dan tidak ada alasan bahkan akan memiliki untuk menjadi sebuah buku., itu bisa juga menjadi jam tangan, atau ditenun menjadi Kaos.

#### **2000-an**

Komisi Eropa mulai mendanai MOBIlearn multi-nasional besar dan M-learning proyek. Perusahaan, yang mengkhususkan diri dalam tiga bidang inti dari mobile learning dibentuk.

1. Authoring dan Penerbitan
2. Pengiriman dan Pelacakan
3. Pengembangan Konten

Konferensi dan pameran dagang diciptakan untuk secara khusus menangani mobile learning dan pendidikan genggam, termasuk: mLearn, WMUTE, dan seri konferensi internasional IADIS Ponsel Learning, ICML di Yordania, Mobile Learning di Malaysia, Handheld Learning di London, dan SALT Mobile di Amerika Serikat. Pada tahun 2003, "Hello China" memelopori penggunaan ponsel untuk pelatihan skala besar dengan mengirimkan kata-kata untuk belajar ponsel peserta di seluruh China. Dengan mobile learning pertengahan tahun 2000 mulai bergerak di luar sekolah dan ke tempat kerja. Sekolah kejuruan di Inggris (FE) mengirim trainee ke tempat kerja bersenjata dengan PDA (pendahulu untuk smartphone), dan perusahaan itu sendiri mulai menawarkan mobile learning dan dukungan kinerja karyawan. Pada tahun 2007 MoleNET inisiatif di Inggris dialokasikan £ 6m dana yang pergi ke perDosenan tinggi 30 FE, untuk mendukung kejuruan mobile, dan di-pelatihan kerja. Pendanaan lebih lanjut diikuti selama 3 tahun ke depan, mengkonsolidasikan peran ponsel di "luar sekolah" belajar.

### **Tahun 2010 dan seterusnya**

Sebagai sejumlah besar perangkat mobile pintar datang ke pasar, pengembang m-learning menghadapi keputusan untuk yang mereka akan menargetkan perangkat (smartphone atau tablet) dan sistem operasi. Apple, Android dan Blackberry adalah, selama beberapa tahun selama dekade ini, ketiga sistem operasi yang paling populer. Blackberry OS adalah abadi menonjol berkurang. Jelas, sistem operasi Apple iOS dan Android yang utama dan menarik aktivitas yang paling di antara kedua pengembang dan pengguna. Desain pembelajaran multi-perangkat menjadi semakin penting sebagai pembelajaran desainer berangkat untuk merancang sekali dan menyampaikan kepada ketiga sistem operasi serta berbagai ukuran perangkat. Tahun 2011 dan 2012 melihat meningkatnya ketersediaan multi-perangkat authoring tools seperti Captivate, Mengartikulasikan Storyline, Lectora dan GoMoLearning. Alat-alat ini memungkinkan program e-learning yang akan dikirimkan ke berbagai sistem operasi mobile dan perangkat serta PC, meskipun tidak semua alat output ke semua sistem operasi dan perangkat.

### II.5.3 *Smartphone*

Sebuah *Smartphone* (atau ponsel pintar) merupakan ponsel dengan kemampuan komputasi yang lebih maju dan konektivitas dari dasar fitur ponsel.

Smartphone awal biasanya menggabungkan fitur dari ponsel dengan orang-orang dari perangkat konsumen populer lain, seperti asisten pribadi digital (PDA), sebuah media player, sebuah kamera digital, atau unit navigasi GPS. Smartphone modern mencakup semua fitur tersebut ditambah fitur dari touchscreen komputer, termasuk web browsing, Wi-Fi, dan pihak ke-3 aplikasi.

Saat ini, sekitar 90% dari penjualan handset di seluruh dunia untuk perangkat didorong oleh Google Android dan Apple iOS sistem operasi mobile.[5]

## II.6 Pemrograman

### II.6.1 PHP

PHP adalah singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan script yang terintegrasi dengan HTML dan berada pada *server*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.[3]

Kelebihan PHP dari bahasa pemrograman lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
3. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang cukup banyak.

PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows).

### II.6.2 PHP MyAdmin

PHP MyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui jaringan WWW.phpMyAdmin mendukung berbagai operasi MySQL, diantaranya mengelola basis data, tabel-tabel, bidang (fields), relasi (relation), indeks, pengguna (users), perijinan (permissions) dan lain-lain.

Pada dasarnya, mengelola basis data dengan MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (*database*), ketikkan baris perintah yang sesuai untuk membuat basis data. Jika seseorang menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu.

Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin, seseorang dapat membuat database, membuat tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya.

Fasilitas pada PHP Myadmin antara lain :

1. Membuat dan menghapus database
2. Dapat membuat FDF grafik, dan mampu mencari data didalam database.
3. Membuat, menyalin, menghapus, dapat menambah field.
4. Manajemen pengguna dan privilege (hak akses) pada MySQL.

### II.6.3 MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

MySQL juga merupakan perangkat lunak database server atau sebut saja DatabaseSmart.Database ini semakin lama semakin populer. Dengan



menggunakan database ini, data semakin aman dan berdaya guna. Database ini juga banyak dipakai pada web database sehingga data semakin terintegrasi antara database dekstop dengan database web.

MySQL mempunyai keistimewaan diantaranya adalah :

1. Dapat berjalan stabil pada berbagai sistem operasi seperti: Linux, Windows, FreeBSD, Mac Os X Server, Solaris, dan masih banyak lagi.
2. MySQL didistribusikan secara open source (gratis), dibawah lisensi GNU General Public Licence (GPL).
3. MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
4. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
5. MySQL memiliki kecepatan yang baik dalam menangani query (perintah sql). Dengan kata lain, dapat memproses lebih banyak SQL per satuan waktu.

#### II.6.4 Apache

*Server HTTP Apache* atau *Server Web/WWW Apache* adalah *server web* yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows* dan *Novell Netware* serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan *HTTP*.

*Apache* memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (*GUI*) yang memungkinkan penanganan *server* menjadi mudah.

*Apache* merupakan perangkat lunak *open source* dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.



## II.7 Java

### II.7.1 Pengertian Java

Java merupakan bahasa pemrograman komputer yang bersamaan , berbasis kelas , object-oriented , dan secara khusus dirancang untuk memiliki beberapa dependensi implementasi mungkin. Hal ini dimaksudkan untuk membiarkan pengembang aplikasi " tulis sekali, jalankan di mana saja "(WORA), yang berarti bahwa kode yang berjalan pada satu platform tidak perlu dikompilasi ulang untuk berjalan di lain. Aplikasi Java biasanya dikompilasi ke bytecode ( file kelas ) yang dapat berjalan pada mesin virtual Java (JVM) terlepas dari arsitektur komputer . Java adalah, pada tahun 2014, salah satu bahasa pemrograman yang paling populer digunakan, terutama untuk aplikasi web client-server, dengan melaporkan 9 juta pengembang. Java pada awalnya dikembangkan oleh James Gosling di Sun Microsystems (yang sejak bergabung menjadi Oracle Corporation ) dan dirilis pada tahun 1995 sebagai komponen inti dari Sun Microsystems ' platform Java . Bahasa berasal banyak yang sintaks dari C dan C + + , tetapi memiliki lebih sedikit tingkat rendah fasilitas dari salah satu dari mereka.

Asli dan referensi implementasi Java compiler , mesin virtual, dan perpustakaan kelas yang dikembangkan oleh Sun dari tahun 1991 dan pertama kali dirilis pada tahun 1995. Pada Mei 2007, sesuai dengan spesifikasi Java Community Process , Sun mengganti lisensi sebagian besar dari teknologi Java bawah the GNU General Public License . Selain itu juga telah mengembangkan implementasi alternatif teknologi Sun ini, seperti GNU Compiler Java (bytecode compiler), GNU Classpath (perpustakaan standar), dan IcedTea -Web (plugin browser untuk applet).

## II.8 Eclipse

Dalam pemrograman komputer , Eclipse adalah sebuah lingkungan pengembangan terpadu (IDE). Ini berisi dasar kerja dan extensible plug-in sistem untuk mengubah lingkungan. Ditulis sebagian besar di Jawa , Eclipse dapat digunakan untuk mengembangkan aplikasi. Dengan menggunakan berbagai plug-in, Eclipse juga dapat digunakan untuk mengembangkan aplikasi di lain pemrograman on Rails framework), Scala , Clojure , Groovy , Scheme , dan Erlang . Hal ini juga dapat digunakan untuk mengembangkan paket untuk

software Mathematica . Pembangunan lingkungan termasuk alat Eclipse Java pengembangan (JDT) untuk Jawa dan Scala, Eclipse CDT untuk C / C + + dan Eclipse PDT untuk PHP, antara lain.

Awal basis kode berasal dari IBM VisualAge . The Eclipse software development kit (SDK), yang meliputi alat pengembangan Java, dimaksudkan untuk pengembang Java. Pengguna dapat memperpanjang kemampuannya dengan menginstal plug-in ditulis untuk Eclipse Platform, seperti toolkit pengembangan untuk bahasa pemrograman lain, dan dapat menulis dan berkontribusi modul plug-in mereka sendiri.

Dirilis di bawah ketentuan Eclipse Public License , Eclipse SDK adalah gratis dan open source software (meskipun tidak sesuai dengan GNU General Public License . Itu adalah salah satu IDE pertama yang berjalan di bawah GNU Classpath dan berjalan tanpa masalah di bawah IcedTea .

### **II.8.1 APK**

Android file paket aplikasi ( APK ) adalah paket format file yang digunakan untuk mendistribusikan dan menginstal perangkat lunak aplikasi dan middleware ke Google sistem operasi Android 's, dan sistem operasi tertentu lainnya, seperti Blackberry 10 Perangkat dengan OS versi 10.2.1.

File APK analog dengan lain paket perangkat lunak seperti paket MSI di Windows atau Deb paket dalam sistem operasi berbasis Debian seperti Ubuntu. Untuk membuat file APK, sebuah program untuk Android pertama kali dikompilasi, dan kemudian semua bagian-bagiannya yang dikemas ke dalam satu file. Sebuah file APK berisi semua kode program (seperti . dex file), sumber daya, aset, sertifikat, dan file manifest . Seperti halnya dengan banyak format file, file APK dapat memiliki nama yang diperlukan, asalkan nama file berakhiran ". Apk".

### **II.8.2 SDK**

The Android software development kit (SDK) termasuk seperangkat alat pengembangan. Ini termasuk debugger , perpustakaan , sebuah handset emulator berdasarkan QEMU , dokumentasi, kode contoh, dan tutorial. Platform pengembangan Saat ini didukung termasuk

komputer yang menjalankan Linux (setiap desktop modern distribusi Linux ), Mac OS X 10.5.8 atau yang lebih baru, Windows XP atau yang lebih baru; untuk saat ini orang dapat mengembangkan software Android di Android sendiri dengan menggunakan [AIDE - IDE Android - Java, C + +] app dan [Android Editor java] app. The resmi didukung lingkungan pengembangan terpadu (IDE) adalah Eclipse menggunakan Android Development Tools (ADT) Plugin, meskipun IntelliJ IDEA IDE (semua edisi) mendukung penuh pengembangan Android di luar kotak, dan NetBeans IDE juga mendukung pengembangan Android melalui Plugin. [ 10 ] Selain itu, pengembang dapat menggunakan editor teks apapun untuk mengedit file XML Java dan, kemudian gunakan baris perintah alat ( Java Development Kit dan Apache Ant diperlukan) untuk menciptakan, membangun dan debug aplikasi Android serta pengendalian terpasang perangkat Android (misalnya, memicu reboot, menginstal paket perangkat lunak (s) jarak jauh).

Perangkat tambahan untuk SDK Android berjalan seiring dengan perkembangan platform Android secara keseluruhan. SDK juga mendukung versi platform Android dalam kasus pengembang ingin menargetkan aplikasi mereka di perangkat yang lebih tua. Pengembangan alat adalah komponen download, jadi setelah seseorang telah mendownload versi terbaru dan platform, platform yang lebih tua dan alat-alat juga dapat didownload untuk pengujian kompatibilitas.

## **II.9 Kuliah**

Kuliah dalam Kamus Besar Bahasa Indonesia mempunyai arti pelajaran yang diberikan atau ceramah. Namun pada umumnya mata kuliah dikaitkan dengan perDosenan tinggi atau pendidikan tinggi yang sering diartikan sebagai proses belajar atau proses pembelajaran.[7]

### **II.9.1 Semester**

Semester merupakan satuan waktu terkecil yang digunakan untuk menyatakan lamanya proses kegiatan belajar-mengajar suatu program dalam suatu jenjang pendidikan. Penyelenggaraan program pendidikan suatu jenjang lengkap dari awal sampai akhir akan dibagi kedalam kegiatan semesteran, sehingga tiap awal semester mahasiswa harus merencanakan dan memutuskan tentang kegiatan belajar apa yang akan ditempuhnya pada semester tersebut.

Satu semester setara dengan kegiatan belajar sekitar 16 minggu kerja, dan diakhiri dengan ujian akhir semester. Satu tahun akademik terdiri dari dua semester reguler yaitu : semester Gasal dan semester Genap.

### **II.9.2 Satuan Kredit Semester**

Satuan Kredit Semester (SKS) adalah satuan yang digunakan untuk menyatakan:

1. Besarnya beban studi mahasiswa
2. Besarnya pengakuan atas keberhasilan usaha belajar mahasiswa
3. Besarnya usaha yang diperlukan mahasiswa untuk menyelesaikan suatu program, baik program semesteran maupun program lengkap
4. Besarnya usaha penyelenggaraan pendidikan bagi tenaga pengajar

### **II.9.3 Beban Studi Kumulatif dan Waktu Studi**

Beban studi semesteran adalah jumlah SKS yang ditempuh mahasiswa pada semester tertentu. Sedangkan Beban Studi Kumulatif

adalah jumlah SKS minimal yang harus ditempuh mahasiswa agar dapat dinyatakan telah menyelesaikan suatu program studi tertentu.

Waktu studi kumulatif adalah batas waktu maksimal yang harus ditempuh mahasiswa dalam menyelesaikan studinya di suatu program pendidikan. Besarnya beban studi kumulatif dan waktu studi kumulatif maksimal bagi tiap program berbeda yaitu :

1. Program Diploma III, minimum 110 SKS dan maksimum 120 SKS yang dijadwalkan untuk masa studi 6 semester dan dapat ditempuh dalam waktu kurang 6 semester dan batas studi selama-lamanya 10 semester;
2. Program Sarjana, minimum 144 SKS dan maksimum 160 SKS yang dijadwalkan untuk masa studi 8 semester dan dapat ditempuh dalam waktu kurang dari 8 semester dan batas studi selama-lamanya 14 semester;
3. Program Magister, beban studi kumulatif ditetapkan oleh program studi masing masing dengan jumlah 36-45 SKS, sudah termasuk 1 SKS Seminar Usulan Penelitian dan 6 SKS Tesis. Waktu studi program magister dijadwalkan untuk 4 semester, dan dapat ditempuh dalam waktu kurang dari 4 semester, dan selama-lamanya 8 semester termasuk penyusunan tesis.

## **II.10 Beban Akademik dan Nilai Kredit Semester**

### **II.10.1 Beban Akademik**

Perencanaan program pendidikan atas dasar beban akademik menyangkut beban belajar untuk mahasiswa dan beban mengajar untuk tenaga pengajar.

### **II.10.2 Nilai Kredit Semester**

Satu SKS adalah program pendidikan atas dasar beban akademik menyangkut beban belajar untuk mahasiswa dan Dosen.

### II.10.3 Nilai Kredit Semester untuk Penyelenggaraan Perkuliahan

Beban studi SKS adalah beban kegiatan yang meliputi yang meliputi keseluruhan 3 macam kegiatan tiap minggu yaitu :

1. Untuk Mahasiswa :

50 menit acara tatap muka terjadwal dengan pengajar 60 menit acara akademik terstruktur yaitu kegiatan studi yang tidak terjadwal tetapi direncanakan dan dipantau oleh tenaga pengajar / Pembimbing Akademik (PA) ; dan 60 menit acara kegiatan akademik mandiri, yaitu kegiatan yang harus dilakukan mahasiswa atas dasar kemampuannya untuk mendalami, mempersiapkan atau tujuan lain dari suatu tugas akademik, dan dipantau oleh tenaga pengajar.

2. Untuk tenaga Pengajar :

50 menit acara tatap muka terjadwal dengan mahasiswa; 60 menit acara perencanaan dan kegiatan akademik terstruktur, dan 60 menit pengembangan materi kuliah.

### II.10.4 Nilai Kredit Semester untuk Praktikum di Laboratorium / Workshop

Satu kredit semester untuk praktikum di laboratorium / Workshop sebanyak 2 sampai 3 jam tiap minggu selama satu semester.

### II.10.5 Nilai Kredit Semester untuk penelitian, Kerja Lapangan, Tugas Akhir dan Sejenisnya

Satu kredit semester untuk penelitian, kerja lapangan, dan sejenisnya, sama dengan penyelesaian kegiatan selama 4 sampai 5 jam tiap minggu untuk satu semester, atau keseluruhannya meliputi 64 sampai 85 jam dalam satu semester, kegiatan tugas akhir dimulai dari perencanaan, penyusunan laporan sampai dengan ujian dihargai 4 SKS.

### II.11 Moodle

Moodle (singkatan dari *Modular Object-Oriented Dynamic Learning Environment*) adalah paket perangkat lunak yang diproduksi untuk kegiatan belajar berbasis internet dan situs web yang menggunakan prinsip *social constructionist pedagogy* Moodle merupakan salah satu konsep dan mekanisme belajar yang memanfaatkan teknologi informasi,



yang dikenal dengan konsep pembelajaran elektronik atau e-learning. Moodle dapat digunakan secara bebas sebagai produk sumber terbuka (*open source*) dibawah lisensi GNU. Moodle dapat diinstal di komputer dan sistem operasi apapun yang bisa dijalankan PHP dan mendukung database SQL.[4]

## II.12 Pembelajaran Jarak Jauh

*E-Learning* memungkinkan mahasiswa untuk menimba ilmu tanpa harus secara fisik menghadiri kelas, mereka bisa belajar di Bandung, sementara instruktur dan pelajaran yang diikuti berada di tempat lain, di kota lain bahkan di Negara lain. Interaksi bisa dijalankan secara *on-line* dan *real-time*.

Pembelajaran belajar dari computer di kantor ataupun dirumah dengan memanfaatkan koneksi jaringan local maupun jaringan internet ataupun menggunakan media CD/DVD yang telah disiapkan. Materi belajar dikelola oleh sebuah pusat penyedia materi dikampus/universitas, atau perusahaan penyedia content tertentu. Pembelajar bisa mengatur sendiri waktu belajar, dan tempat dari mana ia mengakses pelajaran.[5]

## II.13 Layanan Informasi

Layanan informasi adalah penyampaian informasi kepada sasaran layanan agar individu dapat mengelola dan memanfaatkan informasi tersebut.

## II.14 Pengertian E-Learning

Isitilah *e-Learning* mengandung pengertian yang sangat luas, sehingga banyak pakar yang menguraikan tentang definisi e-Learning dari berbagai sudut pandang. Salah satu definisi yang cukup dapat diterima banyak pihak misalnya dari *Darin E Harley* menyatakan :

*e-Learning* merupakan suatu jenis belajar mengajar yang memungkinkan tersampainya bahan ajar ke siswa dengan menggunakan media internet, intranet atau media jaringan computer.

*LearFrame.Com* dalam *Glossary of Learning terms* menyatakan suatu definisi yang luas bahwa:

*E-Learning* adalah sistem edukasi yang menggunakan aplikasi elektronik untuk mendukung sistem mengajar dengan media internet, jaringan komputer, maupun komputer *standalone*. [6]

### II.15 Definisi Informasi

keadaan dari sistem dalam hubungannya dengan keberakhirannya dengan istilah entropy. Informasi yang berguna bagi system akan menghindari proses entropy yang disebut dengan negative entropy atau negentropy.

Apakah sebenarnya informasi itu, sehingga sangat penting artinya bagi suatu system? Informasi dapat didefinisikan sebagai berikut:

Informasi adalah data yang diolah menjadi bentuk yang lebih baik berguna dan lebih berarti bagi yang menerimanya. [7]

### II.16 Konsep Dasar Informasi

Informasi sangat penting di dalam suatu organisasi. Suatu system yang kurang mendapatkan informasi akan menjadi luruh, kacau dan akhirnya berakhir. Informasi sangat penting artinya bagi suatu system sehingga dapat didefinisikan sebagai berikut :

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang. [8]