BAB II

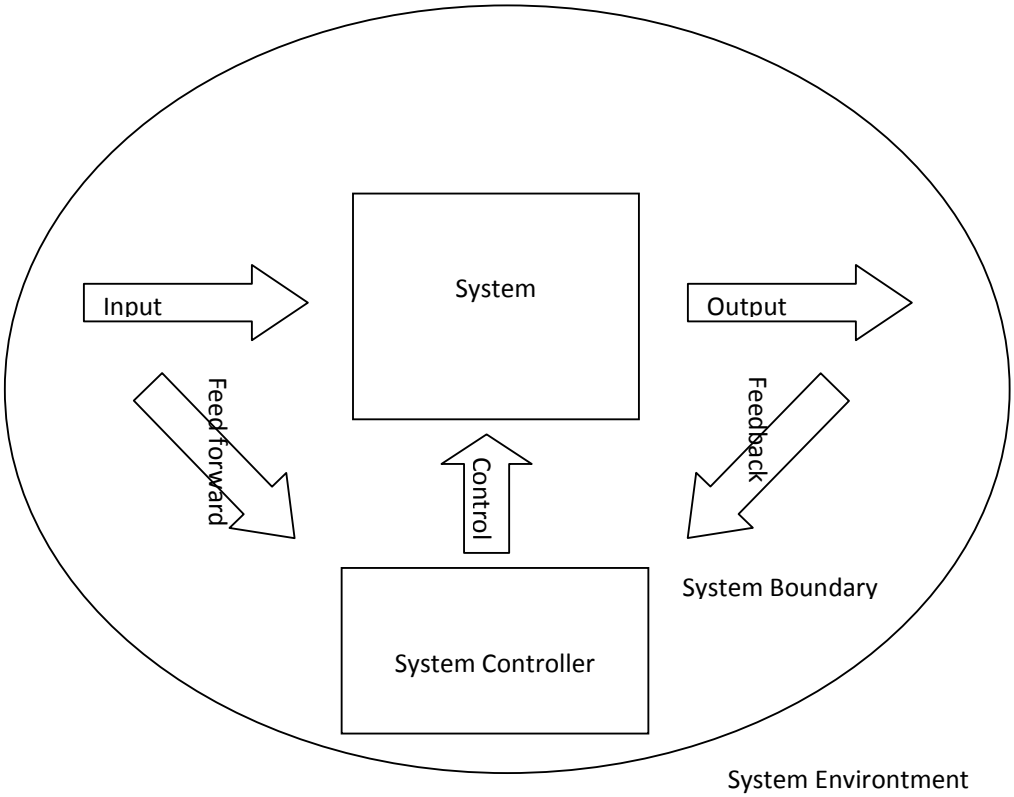
LANDASAN TEORI

2.1 Konsep Dasar Sistem

2.1.1 Definisi sistem

Sistem adalah sekumpulan komponen yang mengimplementasikan model dan fungsionalitas yang dibutuhkan. Komponen-komponen tersebut saling berinteraksi di dalam system guna mentrans-formasi *input* yang diberikan kepada system tersebut menjadi output yang berguna dan bernilai bagi aktor-nya (*Djon Irwanto, S.kom., MM 2005:2)*

Secara umum system dapat digambarkan sebagai berikut :



Gambar 2.1. Gambaran Sistem secara umum

Sumber:Perancangan Object Oriented Software dengan UML : 3

Ada beberapa pendapat yang menjelaskan definisi sistem, yaitu :

1. Menurut (kerz, 2008:175) ”Sistem yaitu gabungan dari sekelompok komponen baik itu gabungan dan / atau bukan manusia (non-human) yang saling mendukung satu sama lain serta diatur menjadi sebuah kesatuan yang utuh untuk mencapai suatu tujuan, sasaran bersama atau akhir ”.
2. Menurut (Hart, 2005:125) ”Sistem mengandung dua pengertian utama yaitu : (a) Pengertian sistem yang menekankan pada komponen atau subsistem- subsistem yang saling berinteraksi satu sama lain, dimana masing-masing bagian tersebut dapat bekerja secara sendiri-sendiri (*indenpendent*) atau bersama-sama serta saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai secara keseluruhan (b) Definisi yang menekankan pada prosedurnya yaitu merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.
3. Menurut (Stoa, 2008:80) ”Pengertian dari sistem merupakan gabungan dari keseluruhan langit dan bumi yang saling bekerja sama yang membentuk suatu keseluruhan dan apabila salah satu unsur tersebut hilang dan tidak berfungsi, maka gabungan keseluruhan tersebut tidak dapat lagi kita sebut sistem.”

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa ” sistem adalah kumpulan bagian-bagian atau subsistem-subsistem yang disatukan dan dirancang untuk mencapai suatu tujuan”.

2.2 Konsep Dasar Informasi

2.2.1 Definisi Data

Sumber informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau data item.

1. Data merupakan deskrpsi dari suatu kejadian yang kita hadapi serta menggambarkan kesatuan nyata yang terjadi pada saat tertentu (Prabu, 2006:185).
2. Data merupakan kumpulan objek-objek beserta atributnya yang menunjukkan karakteristik dari objek tersebut (Phil, 2006:170).

   Informasi tanpa adanya data maka informasi tersebut tidak akan terbentuk. Begitu pentingnya peranan data dalam terjadinya suatu informasi yang berkualitas. Keakuratan data sangat mempengaruhi terhadap keluaran informasi yang akan terbentuk.

2.2.2 Definisi Informasi

Informasi ibarat darah yang mangalir di dalam tubuh suatu organisasi, sehingga informasi ini sangat penting di dalam suatu organisasi. Informasi (*information*) dapat didefinisikan sebagai berikut :

1. Informasi (*Information*) adalah data yang telah dibentuk menjadi sesuatu yang memiliki arti dan berguna bagi manusia (kent, 2008:150).

Berdasarkan pendapat yang dikemukan diatas dapat ditarik kesimpulan bahwa "Informasi adalah data yang sudah diolah, dibentuk, atau dimanipulasi sesuai dengan keperluan tertentu".

2.2.3 Kualitas Informasi

Kualitas Informasi (*quality of information*) (Prabu, 2006:200) diantaranya ditentukan oleh beberapa hal, yaitu:

1. Relevan artinya suatu informasi harus bermanfaat bagi pengguna nya dalam waktu pendek ataupun dalam waktu panjang.
2. Akurat artinya suatu informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan bagi si pemakai serta harus dapat mencerminkan dengan jelas maksud dari informasi tersebut. Ketidak akuratan data terjadi karena sumber dari informasi tersebut mengalami gangguan dalam penyampaiannya baik hal itu dilakukan secara sengaja maupun tidak sehingga menyebabkan data asli tersebut berubah atau rusak.

Komponen keakuratan suatu informasi diantaranya:

a. ***Completeness;*** *Are necessary massage items present ?* hal ini dapat berarti bahwa informasi yang dihasilkan atau dibutuhkan harus memliki kelengkapan yang baik, karena bila informasi yang dihasilkan sebagian-sebagian tentunya akan mempengaruhi dalam

pengambilan keputusan atau menentukan tindakan secara keseluruhan, sehingga akan berpengaruh terhadap kemampuannya untuk mengontrol atau memecahkan suatu masalah yang terjadi dalam suatu organisasi tersebut.

b. ***Correctness;*** *Are message item correct ?* maksudnya bahwa informasi yang diterima kebenarannya tidak perlu diragukan lagi. Kebenaran dari informasi tersebut harus dapat dipertanggung jawabkan.

c. ***Security;*** *Did the massage reach all or anly the intended systems users?* Informasi yang diterima harus terjamin keamanan datanya.

d. ***Time Lines*** (Tepat Waktu); Informasi yang dibutuhkan oleh sipemakai tidak dalam hal penyampaiannya tdak boleh terlambat (usang) karena informasi yang usang maka informasi tersebut tidak mempunyai nilai yang bagus dan kualitasnya pun menjadi buruk sehingga tidak berguna lagi. Jika informasi tersebut digunakan sebagai dasar pengambilan keputusan maka akan berakibat fatal sehingga salah dalam pengambilan keputusan tersebut. Kondisi tersebut mengakibatkan mahalnya nilai suatu informasi, sehingga kecepatan untuk mendapatkan, mengolah serta mengirimnya memerlukan teknologi terbaru.

e. ***Economy*** (Ekonomis); *What level of resources is needed to move information through the problem-solving cycle?*. Kualitas dari

informasi yang digunakan dalam pengambilan keputusan juga bergantung pada nilai ekonomi yang terdapat didalamnya.

f. ***Efficiency*** (Efisien); *What level of resources is required for each unit of information output?*

g. ***Realability*** (Dapat dipercaya); Informasi yang didapatkan oleh pemakai harus dapat di percaya, hal ini menentukan terhadap kualitas informasi serta dalam hal pengambilan setiap tingkatan manajemen.

Kualitas suatu informasi dilihat dari 3 hal yaitu :

a. Akurat artinya suatu informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan bagi si pemakai.
b. Tepat waktu artinya suatu informasi harus sesuai dengan keadaan saat itu, keterlambatan suatu informasi bisa berakibat fatal bagi organisasi pemakainya dalam hal ini dikarenakan informasi merupakan landasan dalam pengambilan sebuah keputusan.
c. Relevan artinya suatu informasi harus bermanfaat bagi pengguna nya dalam waktu pendek ataupun dalam waktu panjang.

2.2.4 Nilai Informasi

Fungsi informasi yang dibutuhkan untuk pengambilan keputusan terkadang diperlukan dengan proses yang cepat dan tidak terduga. Hal itu mengakibatkan penggunaan informasi hanya berdasarkan perkiraan-perkiraan serta informasi yang apa adanya. Dengan perlakukan seperti ini mengakibatkan keputusan yang diambil tidak sesuai dengan yang

diharapkan. Oleh karena itu untuk memperbaiki keputusan yang telah diambil maka pencarian informasi yang lebih tepat perlu dilakukan. Suatu informasi memiliki nilai karena informasi tersebut dapat menjadikan keputusan yang baik dan menguntungkan (memilki nilai informasi yang tepat). Besarnya nilai informasi yang tepat dapat diperoleh dari perbedaan hasil yang didapat dari keputusan yang baru dengan hasil keputusan yang lama dikurangi dengan biaya untuk mendapatkan informasi tersebut. Perhitungan atas informasi yang tepat memberikan banyak manfaat diantaranya untuk menghilangkan pemborosan biaya yang dilakukan untuk mendapatkan informasi yang dibutuhkan dalam pengambilan keputusan tersebut (Pakde_Sofa ,2008).

Sumber:http://massofa.wordpress.com/2008/02/04/nilai-informasi-bagi-pengambil-keputusan.

Menurut Gordon B. Davis nilai informasi dikatakan sempurna apabila perbedaan antara kebijakan optimal, tanpa informasi yang sempurna dan kebijakan optimal menggunakan informasi yang sempurna dapat dinyatakan dengan jelas.

Nilai suatu informasi dapat ditentukan berdasarkan sifatnya. Tentang 10 sifat yang dapat ditentukan nilai informasi, yaitu sebagai berikut :

1. Kemudahan dalam memperoleh

   Informasi memperoleh nilai yang lebih sempurna apabila dapat diperoleh secara mudah. Informasi yang sangat penting dan sangat dibutuhkan menjadi tidak bernilai jika sulit diperoleh.

2. Sifat luas dan kelengkapanya

   Informasi memiliki nilai yang lebih sempurna apabila mempunyai lingkup atau cangkupan yang luas dan lengkap. Informasi yang sepotong dan tidak lengkap menjadi tidak bernilai, karena tidak dapat di gunakan secara baik.

3. Ketelitan (*accurancy*)

   Informasi memiliki nilai yang lebih sempurna apabila mempunyai keteltian yang tinggi / akurat. Informas menjadi tidak bernilai jika tda akurat, karena akan mengakibatkan kesalahan pengambilan keputusan.

4. Kecocokan dengan penggun (*relevance*)

   Informasi memiliki nilai yang lebih sempurna apabila sesuai dengan kebutuhan pengunanya. Informasi berharga dan penting menjadi tidak bernilai jika tidak sesuai dengan kebutuhan penggunanya, karena tidak dapat dimanfaatkan untuk pengambilan keputusan.

5. Tepat waktu

   Informasi memiliki nilai yang lebih sempurna apabila dapat diterima oleh pengguna pada saat yang tepat. Informasi

berharga dan penting menjadi tidak bernilai jika terlambat diterima / *using*, karena tidak dapat dimanfaatkan pada saat pengambilan keputusan.

6. Kejelasan (*clarity*)

   Informasi yang jelas akan meningkatkan kesempurnaan nilai informasi. Kejelasan informasi dipengaruhi oleh bentuk dan format informasi.

7. Fleksibilitas/keluwesan

   Nilai informasi semakin sempurna apabila memiliki fleksibilitas tinggi. Fleksibilitas informasi diperlukan oleh para manajer/pimpinan pada saat pengambilan keputusan

8. Dapat dibuktikan

   Nilai informasi semakin sempurna apabila informasi tersebut dapat dibuktikan kebenarannya. Kebenaran informasi bergantung pada validasi data sumber yang diolah.

9. Tidak ada prasangka

   Nilai informasi semakin sempurna apabila informasi tersebut tidak menimbulkan prasangka dan keraguan adanya kesalahan informasi.

10. Dapat diukur

    Informasi untuk pengamblan keputusan seharusnya dapat diukur agar dapat mencapai nilai yang sempurna.

2.2.5 Mutu Informasi

Menurut Gordon B. Davis, kesalahan informasi adalah antara lain disebabkan oleh hal-hal sebagai berikut :

1. Metode pengumpulan dan pengukuran data yang tidak tepat.
2. Tidak dapat mengikuti prosedur pengolahan yang benar.
3. Hilang/tidak terolahnya sebagian data.
4. Pemeriksaan/pencatatan data yang salah.
5. Dokumen induk yang salah.
6. Kesalahan dia dalam prosedur pengolahan (contoh : kesalahan program aplikasi Komputer yang digunakan).
7. Kesalahan yang dilakukan secara sengaja.

Penyebab kesalahan tersebut dapat diatasi dengan cara-cara sebagai berikut:

1. Kontrol sistem untuk menemukan kesalahan.
2. Pemeriksaan internal dan eksternal.
3. Penambahan batas ketelitian data.
4. Instruksi dari pemakai yang terprogram secara baik dan dapat menilai adanya kesalahan-kesalahan yang mungkin terjadi.

2.3. Konsep Dasar Sistem Informasi

2.3.1 Pengertian Informasi

Menurut Jogiyanto HM (2001:8) Informasi dapat didefinisikan sebagai berikut : "*informasi adalah data yang diolah menjadi bentuk yang*

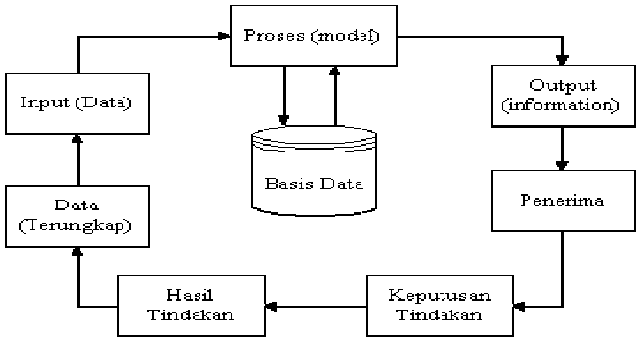
lebih berguna dan lebih berarti bagi yang menerimanya". Sumber dari informasi adalah data.

2.3.2 Pengertian Data

Menurut Azhar Susanto (2000:37) mendefinisikan data sebagai berikut: "data adalah fakta atau apapun yang dapat didigunakan sebagai input untuk menghasilkan informasi". Maka data adalah suatu keterangan tertulis mengenai fakta atau kenyataan yang masih berdiri sendiri, belum mempunyai pengertian sebagai kelompok, belum terkoordinasi satu sama lain dan belum diolah sesuai dengan keperluan tertentu menjadi suatu bentuk yang sesuai dengan keperluan penggguna informasi yang bersangkutan.

2.3.3 Siklus Informasi

Informasi merupakan suatu proses perubahan dunia menjadi informasi. Data yang di olah untuk menghasilkan informasi menggunakan suatu model proses tertentu. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat dan suatu keputusan dan melakukan tindakan yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus, adapun siklus informasi sebagai berikut :



Gambar 2.2 Siklus Informasi

Sumber:http://referensisisteminformasi.blogspot.com/2008/07/siklus-informasi.html

2.4 Pengertian Sistem Informasi

2.4.1 Definisi Sistem Informasi

Terdapat berbagai macam pengertian Sistem Informasi menurut beberapa hal, diantaranya :

1. System Informasi (*Information System*) adalah sekumpulan komponen yang saling berhubungan, mengumpulkan atau mendapatkan, memproses, menyimpan dan mendistribusikan informasi untuk menunjang pengambilan keputusan dan pengawasan dalam suatu organisasi serta membantu manajer dalam mengambil keputusan (kent, 2008:200).
2. Pengertian dari sistem informasi menurut Komunikasi Mahasiswa Sistem Informasi di Yogyakarta memaparkan Sistem Informasi adalah sebuah aplikasi Komputer yang digunakan untuk mendukung operasi dari suatu organisasi serta merupakan aransemen dari orang, data dan proses yang terjadi didalamnya yang berinteraksi satu sama lain dalam

mendukung dan memperbaiki organisasi serta mendukung dalam pemecahan masalah dan kebutuhan pembuat keputusan (KAMI, 2008).

Sistem informasi adalah kumpulan dari sub-sub sistem baik fisik maupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan yaitu mengolah data menjadi informasi yang berguna. Komponen dasar sistem informasi meliputi:

1. Blok masukan (input blok)

   Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan di masukkan ,dapat berupa dokumen –dokumen dasar.

2. Blok model (model block)

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang di inginkan.

3. Blok teknologi (technology block)

   Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian sistem secara keseluruhan.

4. Blok keluaran (blok output)

   Keluaran merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

5. Blok basis data (database block)

   Merupakan kumpulan data yang saling berhubungan satu sama lain, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluaan penyediaan informasi lebih lanjut. Data di dalam basis data perlu di organisasikan sedemikian rupa, supaya informasi yang di hasilkan berkualitas.

6. Blok kendali (controls block)

   Beberapa pengendalian perlu di rancang dan di terapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat di cegah ataupun bila terlanjur terjadi kesalahan langsung cepat diatasi.

   Tujuan dilakukan nya tahap ini sebagai berikut:

   a) Membuat keputusan apabila sistem saat ini mempunyai masalah atau sudah tidak berfungsi secara baik dan hasil analisisnya digunakan sebagai dasar untuk memperbaiki sistem
   b) Mengetahui ruang lingkup pekerjaan yang akan ditangani.
   c) Memahami sistem yang sedang berjalan saat ini, mengidentifikasikan dan mencari solusi

2.5 Sistem Basis Data

Menurut Fathansyah,Ir (2002:9) Basis data adalah :

Sistem basis data merupakan sistem yang terdiri dari atas sekumpulan tabel yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan sekumpulan program (DBMS) yang menungkinkan beberapa pemakai lain untuk mengakses dan memanipulasi table-tabel tersebut.

Sistem Basis data pada dasarnya di buat untuk tujuan Sebagai berikut: (fatansyah 2002:5)

1. Kecepatan dan kemudahan

   Dengan sistem basis data diharapkan pengolahan data dapat lebih cepat dan lebih mudah di bandingkan dengan sistem manual.

2. Efisiensi ruang penyimpanan

   Dengan basis data, efisiensi /optimalisasi penggunaan ruang penyimpan dapat dilakukan, karena kita dapat melakukan penekanan jumlah redudansi data.

3. Keakuratan

   Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan atau batasan tipe, data,domain data, keunikan data dan sebagainya, yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidak akuratan masukan ataupun penyimpanan data.

4. Ketersediaan

   Dengan pertumbuhan data yang semakin banyak, maka di perlukan tempat penyimpanan yang sangat besar, karena itu kita perlu memilah data dalam bentuk kategori-kategori tertentu sehingga data yang tidak terlalu penting dapat di hapus.

5. Kelengkapan

   Untuk mengakomodasi kebutuhan kelengkapan data yang semaki berkembang maka kita tidak hanya dapat menambah record-record data tetapi juga melakukan perubahan struktur dalam basis data, baik dalam bentuk penambahan objek baru (tabel) atau dengan penambahan field-field baru pada suatu tabel.

6. Keamanan

   Dengan sistem basis data kita dapat memproteksi semua objek yang ada dengan berbagai tingkatan keamanan.

7. Kebersamaan Pemakaian

   Pemakai basis data biasanya terdiri dari banyak pemakai sehingga hendaknya basis data harus bisa di akses oleh banyak pemakai dalam waktu yang bersamaan.

Dalam sebuah Basis Data secara lengkap akan terdapat komponen-komponen utama yaitu:

1. Hardware
2. Operating system

3. Database
4. Sistem (aplikasi/perangkat lunak)pengelola basis data(DBMS)
5. User

Keuntungan Database adalah :

a. Mereduksi redudansi
b. Data dapat di share antar aplikasi
c. Standarisasi data dapat di lakukan
d. Batasan security dapat diterapkan
e. Mengelola integritas (keterjaminan akurasi) data
f. Menyeimbangkan kebutuhan yang saling konflik
g. Independesi data (obyektif DBS)

Struktur Sistem Database :

a. File manager : mengelola space dan struktur data.
b. Database manager : menyediakan antar muka dengan data fisik
c. Query processor : menterjemahkan query ke instruksi yang dimengerti database manager
d. DML precompiler :mengkonversi perintah DML menjadi calls normal.
e. DDL compiler : menkonversi perintah DDL menjadi meta data.

2.6 *Unified Modeling Language* (UML)

2.6.1 Definisi *Unified Modeling Language* (UML)

Berikut ini definisini *Unified Modeling Language (UML)* menurut para ahli :

1. Menurut (Hend, 2006) "*Unified Modeling Language* (UML) adalah bahasa yang telah telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak".
2. Menurut (Adi Nugroho : 2005) "*Unified Modeling Language* (UML) adalah alat bantu analis serta perancangan perangkat lunak berbasis objek".

Berdasarkan beberapa pendapat yang dikemukakan dapat ditarkkesimpulan bahwa "*Unified Modeling Language* (UML) adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi. Pemaduan beberapa notasi di beragam metodologi usaha bersama dari banyak pihak, di dukung oleh kakas- kakas yang di integrasikan lewat XML . Standar UML di kelola oleh OMG (Object Management Group)".

2.6.2 Langkah-langkah Penggunaan unified Modeling Language (UML)

Menurut (Afif Amrulla : 2002) "Langkah-langkah penggunaan *unified Modeling Language* (UML), sebagai berikut :

1. Buatlah daftar *business process* dari *level* tertinggi untuk mendefinisikan aktifitas dan proses yang mungkin muncul.

2. Pendekatan *usecase* untuk setiap *business process* untuk mendefinisikan dengan tepat fungsional yang harus disediakan oleh system, kemudian perhalus *usecase diagram* dan dilengkapi dengan *requirement*, *constraints* dan catatan lain.
3. Buatlah *deployment diagram* secara kasar untuk mendefinisikan arsitektur fisik sistem.
4. Definisikan *requirement* lain non fungsonal, *security* dan sebagai yang juga harus disediakan oleh system.
5. Berdasarkan *usecase diagram*, mulailah membuat *activity diagram*.
6. Defisinikan objek-objek level atas *package* atau *domain* dan buatlah *sequence* dan/atau *collaboration* untuk tiap alir pekerjaan, jika sebuah *use case* memungkinkan alir normal dan *error*, buat lagi satu diagram untuk masing-masing alir.
7. Buatlah rancangan *user interface model* yang menyediakan antarmuka bagi pengguna untuk menjalankan *skenario use case*.
8. Berdasarkan model-model yang sudah ada, buatlah *class diagram*. Setiap *package* atau *domain* dipecah menjadi *hirarki class* lengkap dengan *atribut* dan metodenya. Akan lebih baik jika untuk setiap *class* dibuat *unit test* untuk menguji *fungsionalitas class* dan interaksi dengan *class* lain.
9. Setelah *class diagram* itu dibuat, kita dapat melihat kemungkinan pengelompokkan *class* menjadi komponen-komponen karena itu

buatlah *component diagram* pada tahap ini. Juga, definisikan *test integrasi* untuk setiap komponen meyakinkan ia bereaksi dengan baik.

10. Perhalus *deployment diagram* yang sudah dibuat. Detailkan kemampuan dan *requirement* piranti lunak, sistem operasi, jaringan dan sebagainya. Petakan komponen ke dalam node.
11. Memulailah membangun system. Ada dua pendekatan yang tepat digunakan :
    a. Pendekatan *use case* dengan *mengassign* setiap *usecase* kepada tim pengembang tertentu untuk mengembangkan unit kode yang lengkap dengan *test.*
    b. Pendekatan *komponen* yaitu *mengassign* setiap komponen kepada tim pengembang tertentu.
12. Lakukan uji modul dan uj integrasi serta perbaiki model beserta *code*nya. Model harus selalu sesuai dengan *code* yang aktual.
13. Perangkat lunak siap dirilis.

2.6.3 Fokus *Unified Modeling Language* (UML)

Menurut (Adi Nugroho: 2005) "Dalam kerangka spesifikasi, *Unified Modeling Languag (UML)* menyediakan model-model yang tepat, tidak mendua arti (*ambigu*) serta lengkap. Secara khusus *Unified Modeling Language* (UML) menspesifikasikan langkah-langkah penting dalam pengambilan keputusan analisis, perancangan serta implementasi dalam sistem yang sangat bernuansa perangkat lunak(*software intensive system*). Dalam hal ini, *Unified Modeling Language* (UML) bukanlah merupakan

bahasa pemrograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemrograman, sehingga adalah mungkin melakukan pemetaan (*mapping*) langsung dari model-model yang dibuat dengan *Unified Modeling Language* (UML) dengan bahasa-bahasa pemrograman berorientasi objek, sepert Java, Borland Delphi, Visual Basic, C++, dan lain-lain.

Pemetaan (*mapping*) *Unified Modeling Language* (UML) bersifat dua arah, yaitu:

a. Generasi kode bahasa pemrograman tertentu dari *Unified Modeling Language* (UML) *forward engineering.*
b. Generasi kode belum sesuai dengan kebutuhandan harapan pengguna, pengembang dapat melakukan langkah balik bersifat iterative dari implementasi ke *Unified Modeling Language* (UML) hingga didapat system/peranti lunak yang sesuai dengan harapan pengguna dan pengembang".

2.6.4 Bangunan Dasar Metodologi *Unified Modeling Language* (UML)

Menurut (Adi Nugroho: 2005). "Bangunan dasar metodologi *Unified Modeling Language* (UML) menggunakan tiga banguna dasar untuk mendeskripsikan sistem/perangkat lunak yang akan dikembangkan yaitu:

1. Sesuatu (*things*)

Ada 4 (empat) *things* dalam *Unified Modeling Language* (UML), yaitu:

a. *Struktur things*

Merupakan bagian yang relatif statis dalam model *Unified Modeling Language* (UML). Bagian yang relatif statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.

b. *Behavioral things*

Merupakan bagian yang dinamis pada model *Unified Modeling Language* (UML), biasanya merupakan kata kerja dari model *Unified Modeling Language* (UML), yang mencerminkan perilaku sepanjang ruang dan waktu.

c. *Grouping things*

Merupakan bagian pengorganisasian dalam *Unified Modeling Language* (UML). Dalam penggambaran model yang rumit kadang diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model dan subsistem-subsistem.

d. *Annotational things*

Merupakan bagian yang memperjelaskan model *Unified Modeling Language* (UML) dan dapat berupa komentar-komentar yang

menjelaskan fungsi serta cirri-ciri setiap element dalam model *Unified Modeling Language* (UML).

2. Relasi (*Relationship*)

Ada 4 (empat) macam relationship dalam *Unified Modeling Language* (UML), yaitu:

a. Ketergantungan

Merupakan hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*indenpendent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*indenpendent*).

b. Asosiasi

Merupakan apa yang menghubungkan antara objek satu dengan objek yang lainnya, bagaimana hubungan suatu objek dengan objek lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.

c. Generalisasi

Merupakan hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk (*ancestor*). Arah dari atas kebawah dari objek induk ke objek anak dinamakan spesialisasi, sedangkan arah berlawanan sebaliknya dari arah bawah keatas dinamakan generalisasi.

d. Realisasi

Merupakan operasi yang benar-benar dilakukan oleh suatu objek.

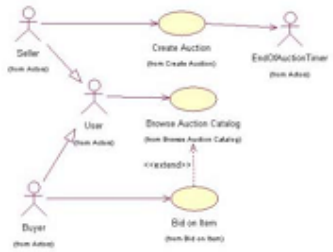
3. *Diagram*

Berikut adalah macam diagram dalam *Unified Modeling Language* (UML), yaitu:

a. Use Case Diagram

Diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta di harapkan pengguna.

Contoh *Usecase diagram* :



Gambar 2.3 Usecase Diagram

Sumber:http://www.cset.oit.edu/~yangs/CST412/Handout/UseCaseDiagram.jpg

b. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut

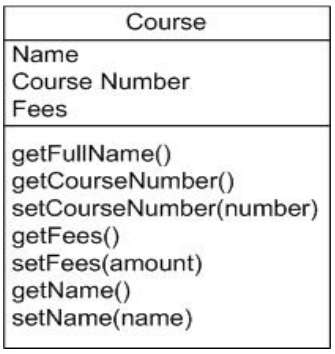
(metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a) Private, tidak dapat dipanggil dari luar class yang bersangkutan
b) Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
c) Public, dapat dipanggil oleh siapa saja.

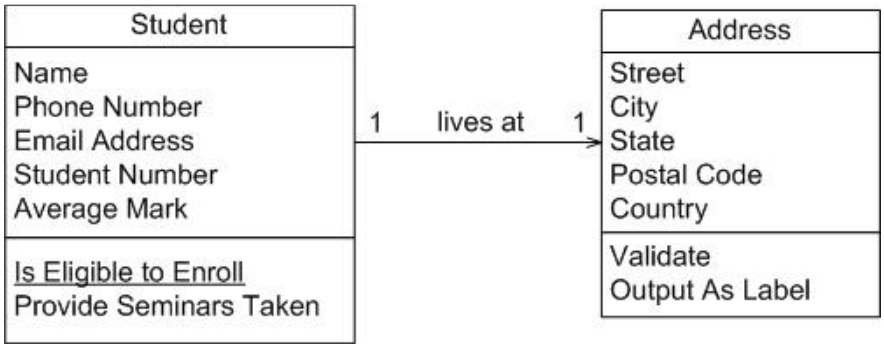


Gambar 2.4 Class Diagram
Sumber: http://www.agilemodeling.com/artifacts/classDiagram.htm

Class dapat merupakan implementasi dari sebuah interface, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah

class. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

Sesuai dengan perkembangan *class model*, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.



Gambar 2.5 Relasi *Class* Diagram

Sumber : http://www.agilemodeling.com/artifacts/classDiagram.htm

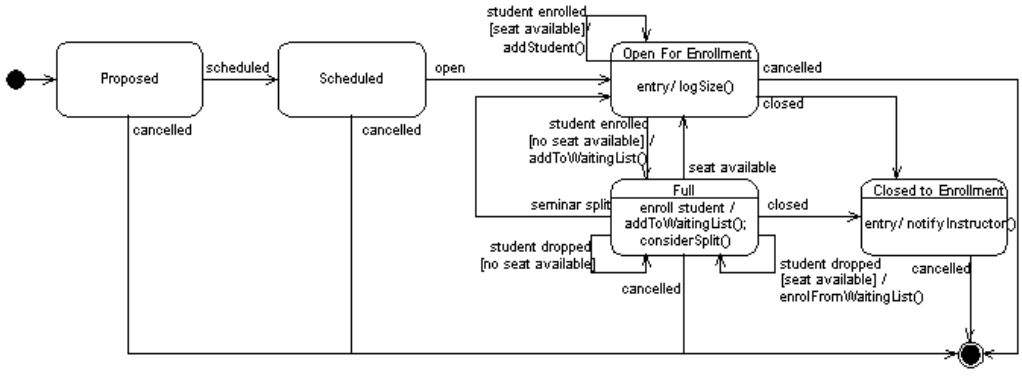
Hubungan Antar *Class*

1) Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2) Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
3) Pewarisan, yaitu hubungan hirarkis antar class. *Class* dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

4) Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

c. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statech diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statech diagram). Dalam UML, state digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar state umumnya memiliki kondisi guard yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. Action yang dilakukan sebagai akibat dari event tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.



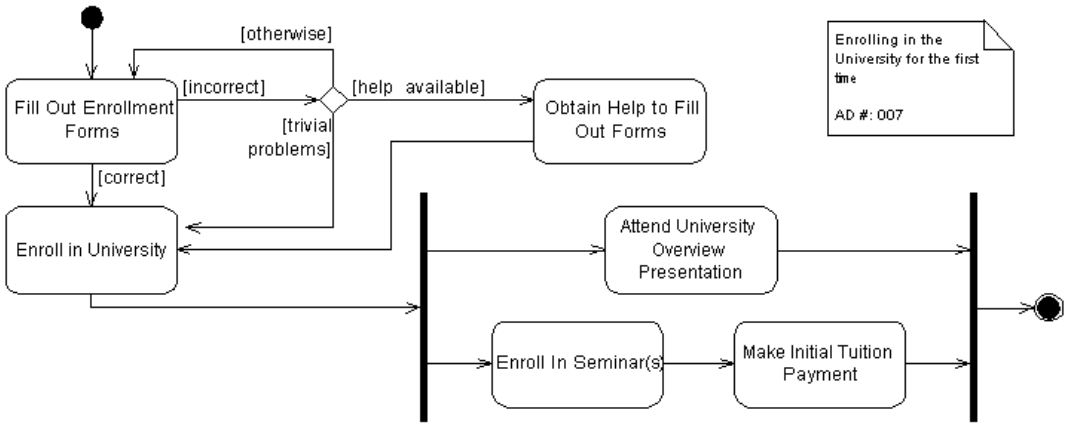
Gambar 2.6 Statechart Diagram
Sumber : http://www.agilemodeling.com/style/stateChartDiagram.htm

d. Activity Diagram

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

swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Contoh activity diagram tanpa swimlane:



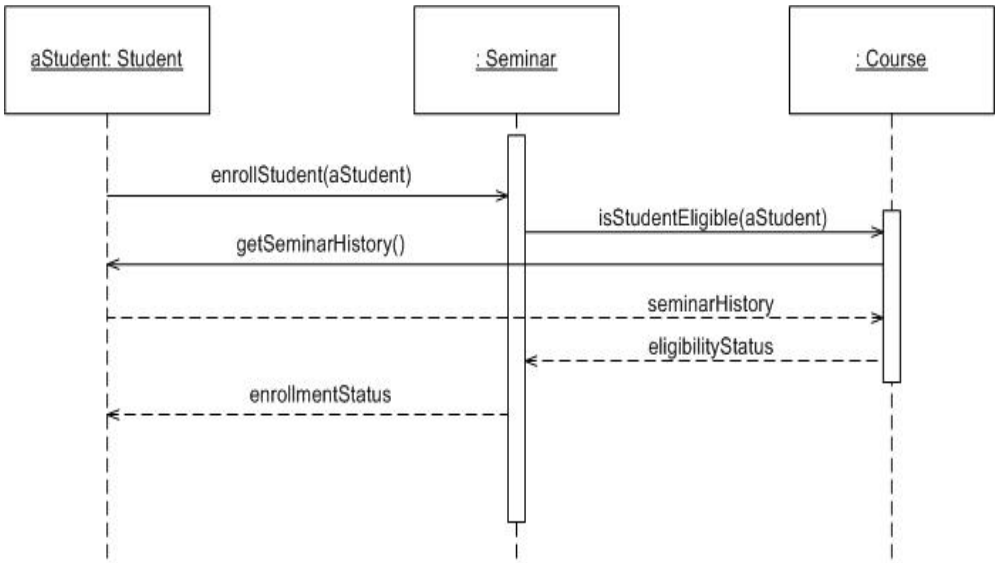
Gambar 2.7 Diagram Activity

Sumber: http://www.agilemodeling.com/style/activityDiagram.htm

e. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message

digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek *boundary*, *controller* dan *persistent entity*.



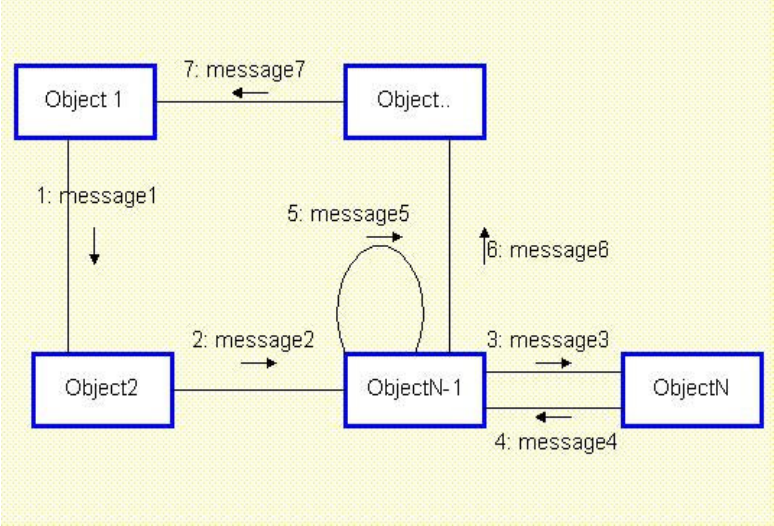
Gambar 2.8 Sequence Diagram

Sumber: http://www.agilemodeling.com/artifacts/sequenceDiagram.htm

f. Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki *sequence number*, di mana

message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.



Gambar 2.9 Collaboration Diagram

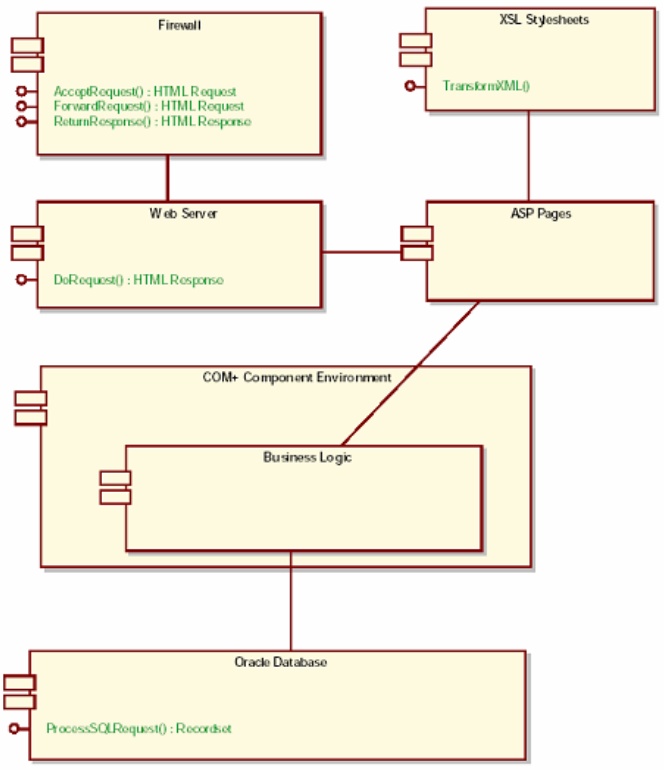
Sumber : http://it.toolbox.com/blogs/enterprise-solutions/defining-collaboration-diagrams-21140

g. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) di antaranya. Komponen piranti lunak adalah modul berisi code, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada compile time, link time, maupun run time. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu

kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

Contoh component diagram:



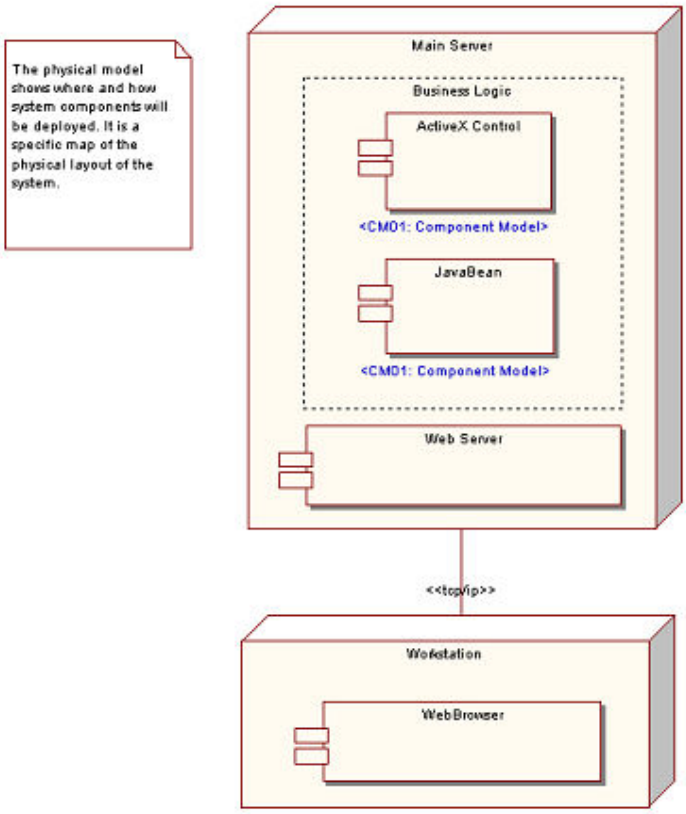
Gambar 2.10 Component Diagram
Sumber : http://ikc.cbn.net.id/umum/yanti-uml.php

h. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal . Sebuah node adalah server, workstation, atau piranti keras lain yang digunakan untuk

men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

Contoh deployment diagram :



Gambar 2.11 Deployment Diagram
Sumber : http://ikc.cbn.net.id/umum/yanti-uml.php

Tool Yang Mendukung UML

Saat ini banyak sekali tool pendesainan yang mendukung UML, baik itu tool komersial maupun opensource. Beberapa diantaranya adalah:

1. Rational Rose (www.rational.com)
2. Together (www.togethersoft.com)

3. MagicDraw (www.nomagic.com/magicdrawuml)
4. Visual Object Modeller (www.visualobject.com)

2.7 Bahasa Pemprograman Yang di Gunakan

Bahasa pemprograman yang digunakan dalam penulisan skripsi ini adalah PHP, HTML, Mysql.

2.7.1 PHP (*Personal Home Page*)

PHP merupakan bahasa pemprograman *server side* yang di desain khusus untuk aplikasi web yang di tambahkan kedalam HTML . Sifat server side berarti pengerjaan skrip akan dilakukan di server, baru kemudian hasilnya dikirim ke browser.

Kelebihan PHP dari bahasa pemrograman lain

(http://id.wikipedia.org/wiki/*php*/ 10 September 2009**)**

a) Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaanya.

b) Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan configurasi yang relatif mudah.

c) Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.

d) Dalam sisi pemahamanan, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.

e) PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara

runtime melalui console serta juga dapat menjalankan perintah-perintah sistem.

f) Skrip tidak dapat di intip dengan menggunakan fasilitas view HTML source.

PHP dirancang sedemikian khusus untuk membentuk web dinamis artinya dapat membentuk suatu tampilan berdasarkan permintaan terkini.

Pada saat ini PHP cukup popular sebagai peranti pemprograman web karena dapat berfungsi pada server-server yang berbasis unix, linux, windows dan mancitosh dan untuk mencoba kita tidak perlu menggunakan computer berkelas server, cukup dengan menggunakan computer biasa kita bisa mempelajari dan mempraktekkan PHP.

9 Tipe data yang di miliki PHP :

1. Integer
2. Double
3. Boolean
4. String
5. Object
6. Array
7. Null
8. Nill
9. Resource

Program HELLO WORLD yang dituliskan PHP adalah sebagai berikut :

```
<?php
echo "<b>Hello World</b>" ;
echo "<b>Halo Dunia</b>" ;
?>
```

2.7.2 HTML(Hypertext Markup Languange) Dan CSS (Cascading Style Sheet)

(http://id.wikipedia.org/wiki/*HTML* / 10 September 2009)

HyperText Markup Language (HTML) adalah sebuah *bahasa markup* yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah browser Internet. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML *(Standard Generalized Markup Language),* HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C).

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan browser web seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan browser.

HTML dokumen tersebut mirip dengan dokumen teks biasa, hanya dalam dokumen ini sebuah teks bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Sebagai contoh jika ingin membuat teks ditampilkan menjadi tebal seperti: **TAMPIL TEBAL**, maka penulisannya dilakukan dengan cara: TAMPIL TEBAL. Tanda digunakan untuk mengaktifkan instruksi cetak tebal, diikuti oleh teks yang ingin ditebalkan, dan diakhiri dengan tanda untuk menonaktifkan cetak tebal tersebut.

Secara garis besar, terdapat 4 jenis elemen dari HTML:

1. *Structural*. tanda yang menentukan level atau tingkatan dari sebuah teks (contoh, <h1>Golf</h1> akan memerintahkan browser untuk menampilkan "Golf" sebagai teks tebal besar yang menunjukkan sebagai Heading 1.
2. *Presentational*. tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut (contoh, boldface akan menampilkan **bold**. Tanda presentational saat ini sudah mulai digantikan oleh CSS dan tidak direkomendasikan untuk mengatur tampilan teks.
3. *Hypertext*. tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala ke dokumen lain (contoh, Wikipedia akan menampilkan Wikipedia sebagai sebuah hyperlink ke URL tertentu).

4. Elemen *widget* yang membuat objek-objek lain seperti tombol (<button>), list (<li>), dan garis horizontal (<hr>).

Selain markup *presentational* , markup yang lin tidak menentukan bagaimana tampilan dari sebuah teks. Namun untuk saat ini, penggunaan tag HTML untuk menentukan tampilan telah dianjurkan untuk mulai ditinggalkan dan sebagai gantinya digunakan *Cascading Style Sheets.*

Contoh dokumen HTML sederhana

```
<!DOCTYPE html>
<html>
<head>
<title>Halo HTML</title>
</head>
<body>
        <p>Halo Dunia!</p>
</body>
</html>
```

Yang bisa dilakukan dengan HTML yaitu :

1. Mengontrol tampilan dari web dan content nya
2. Mempublikasikan dokumen secara online sehingga bisa di akses dari seluruh dunia.
3. Membuat online form yang bisa digunakan untuk menangani pendaftaran, transaksi secara online.

4. Menambahkan objek2 seperti image, audi, video dan juga java applet dalam dokumen dan HTML.

Sedangkan CSS merupakan *feature* yang sangat penting dalam *dynamic* HTML. Meskipun bukan merupakan suatu keharusan dalam membuat web,akan tetapi penggunaan *style sheet* merupakan kelebihan tersendiri.

Suatu *style sheet* merupakan tempat dimana dapat mengontrol dan memanage *style-style* yang ada. *Style sheet* mendeskripsikan bagaimana tampilan document HTML yang menggunakannya,juga bisa membuat efek-efek spesial di web anda dengan menggunakan *style sheet*. Sebagai contoh anda bisa membuat *style sheet* yang mendefenisikan style untuk <H1> dengan *style bold* dan *italic* dan berwarna biru. Atau pada tag <P> yang akan di tampilkan dengan warna biru dan menggunakan font gothic dan masih banyak lagi yang bisa dilakukan dengan style sheet. Secara teoritis bisa menggunakan style sheet technologi denga HTML. Akan tetapi pada prakteknya hanya *Cascading Style* Sheet (CSS) tehnology yang support pada hampir semua web browser .karena css telah di standarakan oleh *World Wide Web consortium* (w3c) untuk digunakan di web browser.

2.7.3 Java Script

(http://id.wikipedia.org/wiki/*JavaScript* / 10 September 2009)

JavaScript adalah nama implementasi *Netscape Communications Corporation* untuk ECMAScript standar, suatu bahasa skrip yang didasarkan pada konsep pemrograman berbasis prototipe. Bahasa ini

terutama terkenal karena penggunaannya di situs web (sebagai JavaScript sisi klien) dan juga digunakan untuk menyediakan akses skrip untuk objek yang dibenamkan (*embedded*) di aplikasi lain.

Walaupun memiliki nama serupa, JavaScript hanya sedikit berhubungan dengan bahasa pemrograman Java, dengan kesamaan utamanya adalah penggunaan sintaks C. Secara semantik, JavaScript memiliki lebih banyak kesamaan dengan bahasa pemrograman Self.

Skrip JavaScript yang dimasukkan di dalam berkas HTML ataupun XHTML harus dimasukkan di antara tag <script>...</script>. Berikut ini adalah contohnya (yang akan menampilkan sebuah *dialog box* berisi Halo Dunia! ketika sebuah tombol diklik oleh pengguna):

```
<input type="button" value="Tekan di sini" onclick="halo();">
<script type="text/javascript">
 function halo() {
 alert( "Halo Dunia!" );
  }
</script>
```

2.7.4 Ajax

Asynchronous JavaScript And XML, atau disingkat AJAX, adalah suatu teknik pemrograman berbasis web untuk menciptakan aplikasi web interaktif. Tujuannya adalah untuk memindahkan sebagian besar interaksi pada komputer web surfer, melakukan pertukaran data dengan server di belakang layar, sehingga halaman web tidak harus dibaca ulang secara

keseluruhan setiap kali seorang pengguna melakukan perubahan. Hal ini akan meningkatkan interaktivitas, kecepatan, dan usability. Ajax merupakan kombinasi dari:

1. XHTML (atau HTML) dan CSS untuk bahasa mark up dan tampilan.
2. DOM yang diakses dengan client side scripting language, khususnya implementasi ECMAScript JavaScript dan JScript, untuk menampilkan secara dinamis dan berinteraksi dengan informasi yang ditampilkan seperti
3. Objek XMLHttpRequest untuk melakukan pertukaran data asinkronus dengan web server. Pada beberapa framework dan kasus Ajax, objek IFrame lebih dipilih daripada XMLHttpRequest untuk melakukan pertukaran data dengan web server.
4. XML umumnya digunakan sebagai format untuk pengiriman data, walaupun format lain juga memungkinkan, seperti HTML, plain text, JSON dan EBML. Seperti halnya DHTML, LAMP, atau SPA, Ajax bukanlah teknologi spesifik, melainkan merupakan gabungan dari teknologi yang dipakai bersamaan. Bahkan, teknologi turunan/komposit yang berdasarkan Ajax, seperti AFLAX sudah mulai bermunculan

2.8 Perangkat Lunak Yang Digunakan

2.8.1. Database MySQL

http://id.wikipedia.org/wiki/*MySQL*/ 10 September 2009

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang

multithread, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-

program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain :

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. *Open Source*

   MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser* MySQL

   Dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance* tuning

   MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. *Command And functions*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam query.

7. *Security*

   MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

8. *Scalability And limits*

   MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT)

10. *Interface*

    MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

11. *Clients and tools*

    MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

12. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

2.8.2. JQuery

(http://id.wikipedia.org/wiki/*Jquery*/ 10 September 2009)

jQuery adalah pustaka JavaScript kecil bersumber terbuka yang menekankan pada interaksi antara JavaScript dan HTML. Pustaka ini dirilis pada Januari 2006 di BarCamp NYC oleh John Resig dan berlisensi ganda di bawah Lisensi MIT dan GPL.

Microsoft dan Nokia telah mengumumkan akan mengemas jQuery di platform mereka. Microsoft awalnya mengadopsinya dalam Visual Studio untuk digunakan dalam ASP.NET AJAX dan ASP.NET MVC Framework, sedangkan Nokia akan mengintegrasikannya dalam kerangka Web Run-Time mereka.

2.9 Definisi Online

Secara umum, sesuatu dikatakan online adalah terkoneksi/terhubung dengan suatu jaringan ataupun sistem yang lebih besar. Beberapa arti kata lainnya yang lebih spesifik adalah sebagai berikut :

1. Dalam percakapan umum, jaringan/*network* yang lebih besar dalam konteks ini biasanya lebih mengarah pada *internet*, sehingga '*online*' menjelaskan status bahwa ia dapat diakses melalui *internet*.

2. Secara lebih spesifik dalam sebuah sistem yang terkait pada ukuran dalam satu aktifitas tertentu, sebuah elemen dari sistem tersebut dikatakan *online* jika elemen tersebut beroperasional. Sebagai contoh, sebuah instalasi pembangkit listrik dikatakan *online* jika ia dapat menyediakan listrik pada jaringan elekrik.
3. Dalam telekomunikasi, istilah *online* memiliki arti lain yang lebih spesifik. Suatu alat diasosiasikan dalam sebuah sistem yang lebih besar dkatakan *online* bila berada dalam kontrol langsung dari sistem tersebut. Dalam arti jika ia bersedia saat akan digunakan oleh sistem (*on-demand*), tanpa membutuhkan intervensi manusia, namun tidak bisa beroperasi secara mandiri di luar sistem tersebut.

2.10 Internet

http://www.total.or.id/info.php?kk=*internet*/ 29maret 2009

Singkatan dari *Interconnection Networking*. *The network of the networks*. Diartikan sebagai a *global network of computer networks* atau sebuah jaringan komputer dalam skala global/mendunia. Jaringan komputer ini berskala internasional yang dapat membuat masing-masing komputer saling berkomunikasi. *Network* ini membentuk jaringan inter-koneksi (Inter-connected network) yang terhubung melalui protokol TCP/IP. Dikembangkan dan diuji coba pertama kali pada tahun 1969 oleh US Department of Defense dalam proyek ARPAnet. dalam arti akan membentuk suatu.

2.11 WWW (*Word Wide Web*)

(http://ibliz-kecil.blogspot.com/2008/08/*pengertian-www*.html/) 10 September 2009

WWW (World Wide Web) adalah sebuah bagian dari internet yang sangat dikenal dalam dunia internet, dengan adanya WWW seorang dapat menampilkan sebuah halaman virtual yang disebut website.

Jika dilihat dari proses kerjanya WWW dapat dibagi menjadi beberapa komponen yaitu:

1. Protocol
2. Address
3. HTML (Hypertext Markup Language)
4. HTTP (Hypertext Transfer Protocol)
5. FTP (File Transfer Protocol)
7. Mailto
8. News
9. Telnet (Tele Networking)

Secara teknis, web adalah sebuah sistem dimana informasi dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah *internet webserver* dipresentasikan dalam bentuk *hypertext*. Informasi di web dalam bentuk teks umumnya ditulis dalam format HTML (*Hypertext Markup Language*). Informasi lainnya disajikan dalam bentuk *grafis* (dalam format GIF, JPG, PNG), *suara* (dalam format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World). Web dapat

diakses oleh perangkat lunak *web client* yang secara populer disebut sebagai *browser*. *Browser* membaca halaman-halaman web yang tersimpan dalam *webserver* melalui protokol yang disebut HTTP (*Hypertext Transfer Protocol*). Dewasa ini, tersedia beragam perangkat lunak *browser*. Beberapa diantaranya cukup populer dan digunakan secara meluas, contohnya seperti Microsoft Internet Explorer, Netscape Navigator, maupun Opera, namun ada juga beberapa produk browser yang kurang dikenal dan hanya digunakan di lingkungan yang terbatas. Sebagai dokumen *hypertext*, dokumen-dokumen di web dapat memiliki *link (sambungan)* dengan dokumen lain, baik yang tersimpan dalam *webserver* yang sama maupun di *webserver* lainnya. Link memudahkan para pengakses web berpindah dari satu halaman ke halaman lainnya, dan "berkelana" dari satu server ke server lain. Kegiatan penelusuran halaman web ini biasa diistilahkan sebagai *browsing*, ada juga yang menyebutnya sebagai *surfing* (berselancar).

http://this-live.blogspot.com/2008/11/*pengertian-www*.html/ 10 September 2009