BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Pengertian sistem terbagi menjadi dua yaitu :

Dilihat dari pendekatan yang menekankan pada prosedur dan dilihat dari pendekatan yang menekankan pada elemen / komponen.

Pengertian sistem yang menekankan pada prosedur didefinisikan oleh Jogiyanto adalah :

*"*suatu system adalah suatu jaringan kerja dari procedure – procedure yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.*"*(Jogiyanto,2001,1)

Pengertian sistem yang menekankan pada prosedur didefinisikan oleh *L. ACKOF* Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.

Syarat-syarat sistem :

1. Sistem harus dibentuk untuk menyelesaikan masalah.
2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan diantara elemen sistem.
4. Unsur dasar dari proses (arus informasi, energi dan material) lebih penting dari pada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen.

Sedangkan pengertian prosedur adalah :

“ Suatu urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan, dan bagaimana (*how*) mengerjakannya ” (3,1).

2.1.2 Elemen Sistem

Menurut Ladjamudin (2005) ada beberapa elemen yang membentuk sebuah sistem yaitu :

1. Tujuan

   Setiap sistem memiliki tujuan (*goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tidak terkendali

2. Masukan

   Masukan (*input*) sistem adalah segala sesuatu yang masuk kedalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud(tampak secara fisik) maupun yang tidak tampak.

3. Keluaran

   Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran,cetakan laporan dan sebagainya.

4. Proses

   Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan

produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah.

5. Mekanisme Pengembalian Umpan Balik

   Mekanisme pengendalian (control mechanism) diwujudkan dengan menggunakan umpan balik (feedback), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan ataupun proses.

2.1.3 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu. Karakteristik sistem tersebut adalah sebagai berikut :

1. Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi artinya saling bekerja sama membentuk satu kesatuan.

2. Batas Sistem

   Batas Sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya.

3. Lingkaran Luar Sistem

   Lingkungan luar (*environment*) adalah apapun diluar batas dari sistem yang mempengaruhi dari operasi sistem.

4. Penghubung Sistem

   Penghubung (*Interface*) merupakan media penghubung antara subsistem dengan subsistem lainnya yang memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem yang lainnya.

5. Masukan Sistem

   Masukan *(input)* adalah energy yang dimasukan kedalam sistem. Masukan dapat berupa masukan perawatan *(maintenance input)* dan masukan sinyal *(signal input)***.** Masukan perawatan yaitu energy yang dimasukan supaya sistem tersebut dapat beroperasi, sedangkan masukan sinyal yaitu energy yang diproses untuk mendapatkan keluar.

6. Keluaran Sistem

   Keluaran *(output)* adalah hasil dari energy yang diolah dan diklarifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolahan Sistem

   Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah masukan *(input)* menjadi keluaran *(output).*

8. Sasaran Sistem

   Suatu sistem mempunyai tujuan *(goal)* atau sasaran *(objective)***.** Kalau suatu sistem tidak mempunyaisasaran maka operasi sistem tidak aka nada manfaatnya. Sasaran dari sistem sangat menentukan sekali masukan yang akan dihasilkan oleh sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.1.4 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari berbagai sudut pandang diantaranya adalah sebagai berikut :

1. Sistem klasifikasi sebagai sistem abstrak dan sistem fisik.

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide - ide yang tidak tampak secara phisik seperti sistem hubungan antara manusia dan tuhan. Seangkan sistem phisik merupakan sistem yang ada secara fisik seperti sistem komputer.

2. Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan.

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, dan tidak dibuat oleh manusia seperti sistem perputaran bumi. Sedangkan sistem buatan manusia adalah sistem yang melibatkan interaksi antara manusia dan mesin bisa disebut *human machine system*. Sistem informasi akutansi merupakan contoh sistem tersebut karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem diklasifikasikan sebagai sistem tertentu dan sistem tertentu dan sistem tak tentu.

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem komputer adalah contoh dari sistem tertentu yangg tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan sedangkan sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probalitas.

4. Sistem diklasifikasikan sebagai sistem tertutup dan sistem terbuka.

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruhi dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya.

2.2 Konsep dasar Informasi

Informasi merupakan hal yang sangat penting di dalam sebuah sistem. Jika sebuah sistem mengolah informasi yang salah maka penerima informasi akan susah untuk mengambil keputusan masa kini atau masa yang akan datang.

2.2.1 Pengertian Informasi

Informasi adalah faktor yang terpenting dalam sistem untuk pengambilan suatu keputusan. Pengertian informasi menurut Jogiyanto adalah sebagai berikut :

“Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”(jogiyanto,2005,8)

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal data atau data item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian (Event) adalah sesuatu yang terjadi pada saat tertentu.

2.2.2 Kualitas Informasi

Kualitas informasi sangat dipengaruhi atau ditentukan oleh hal-hal sebagai berikut. Al-Bahra (2005:11).

1. Relevan (*relevancy*), yaitu sejauh mana tingkat relevansi informasi tersebut terhadap kenyataan kejadian masa lalu, masa sekarang dan kejadian yang akan datang.
2. Akurat (*accuracy*), yaitu suatu informasi dikatakan berkualitas jika seluruh kebutuhan informasi telah tersampaiakan serta pesan yang disampaiakan sudah lengkap sesuai dengan yang diinginkan oleh user.
3. Tepat Waktu (*timelines*), informasi yang datang pada penerima tidak boleh terlambat. Suatu informasi harus sesuai dengan keadaan saat itu. Keterlambatan suatu informasi bisa berakibat fatal bagi suatu organisasi atau pemakainya hal ini dikarenakan informasi merupakan landasan dalam pengambilan keputusan.
4. Ekonomis (*economy*), informasi yang dihasilkan harus mempunyai daya jual yang tinggi dan biaya operasional yang harus dikeluarkan untuk menghasilkan informasi tersebut harus minimal, informasi tersebut juga mapu memberikan dampak yang luas terhadap laju pertumbuhan ekonomi dan teknologi informasi.
5. Efisien (*efficiency*), informasi yang berkualitas harus memiliki kalimat yang sederhana dan mudah dimengerti, tapi bisa memberikan makna yang mendalam.
6. Dapat dipercaya (*reliability*), informasi yang didapat harus dari sumber yang bisa dipercaya. Sumber tersebut juga harus sudah teruji tingkat kejujurannya.

2.3 Sistem Informasi

2.3.1 Pengertian Sistem Informasi

Menurut Abdul Kadir (2005) dalam bukunya yang berjudul pengenalan sistem informasi, "bahwa sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokan, diproses menjadi informasi, dan didistribusikan kepada pemakai (Hall, 2001)". Selain itu juga sistem informasi merupakan kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (input) menjadi keluaran (informasi), guna mencapai sasaran – sasaran perusahaan (Wilkinson, 1992).

2.3.2 Komponen Sistem Informasi

Adapun beberapa elemen / komponen dalam sistem informasi dalam buku Al-bahra (2005:14) dapat diklasifikasikan sebagai berikut.

1. Hardware dan software yang berfungsi sebagai mesin.
2. People dan procedures yang merupakan manusia dan tatacara menggunakan mesin.
3. Data merupakan jembatan penghubung antara manuia dan mesin agar terjadi suatu proses pengolahan data.

2.4 Model Pengambilan Keputusan

2.4.1 FMADM (*Fuzzy Multiple Attribute Decission Making*)

Fuzzy Multiple Attribute Decision Making (FMADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari FMADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan. Pada dasarnya, ada 3 pendekatan untuk mencari nilai bobot atribut, yaitu pendekatan subyektif, pendekatan obyektif dan pendekatan integrasi antara subyektif dan obyektif. Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari para pengambil keputusan, sehingga beberapa faktor dalam proses perankingan alternatif bisa ditentukan secara bebas. Sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektifitas dari pengambil keputusan.

Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah FMADM. antara lain:

1. *Simple Additive Weighting Method* (SAW).
2. *Weighted Product* (WP).

3. ELECTRE
4. *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)
5. *Analytic Hierarchy Process* (AHP)

2.4.2 SAW (*Simple Additive Weighting*)

Metode SAW sering juga dikenal istilah metode penjumlahan terbobot.Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

$$r_{ij} = \begin{cases} \dfrac{X_{ij}}{\underset{i}{Max}\ X_{ij}} & Jika\ j\ adalah\ atribut\ keuntungan\ (Benefit) \\ \dfrac{\underset{i}{Min}\ X_{ij}}{X_{ij}} & Jika\ j\ adalah\ atribut\ biaya\ (Cost) \end{cases} \qquad (1)$$

Keterangan :

r_{ij} = Nilai rating kinerja ternormalisasi

X_{ij} = Nilai atribut yang dimiliki dari setiap kriteria

Max X_{ij} = Nilai terbesar dari setiap kriteria
i

Min X_{ij} = Nilai terkecil dari setiap kriteria
I

Benefit = Jika nilai terbesar adalah terbaik

Cost = Jika nilai terkecil adalah terbaik

dimana r*ij* adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj; i=1,2,...,m dan j=1,2,...,n. Nilai preferensi untuk setiap alternatif (Vi) diberikan sebagai:

$$V_i = \sum_{j=1}^{n} W_j \, r_{ij} \qquad (2)$$

Keterangan :

V_i = rangking untuk setiap alternative

w_j = nilai bobot dari setiap kriteria

r_{ij} = nilai rating kinerja ternormalisasi

Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih.

2.4.3 Langkah Penyelesaian

Dalam penelitian ini menggunakan model FMADM dengan metode SAW. Adapun langkah-langkahnya adalah:

1. Memberikan nilai setiap alternatif (Ai) pada setiap kriteria (Cj) yang sudah ditentukan, dimana nilai i=1,2,…m dan j=1,2,…n.
2. Memberikan nilai bobot (W) yang juga didapatkan berdasarkan nilai *crisp*.
3. Melakukan normalisasi matriks dengan cara menghitung nilai rating kinerja ternormalisasi (r_{ij}) dari alternatif A_i pada atribut C_j berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan/*benefit*=MAKSIMUM atau atribut biaya/*cost*=MINIMUM). Apabila berupa artibut keuntungan maka nilai crisp (X_{ij}) dari setiap kolom atribut dibagi dengan nilai crisp MAX (MAX X_{ij}) dari tiap kolom, sedangkan untuk atribut biaya, nilai crisp MIN (MIN X_{ij}) dari tiap kolom atribut dibagi dengan nilai crisp (X_{ij}) setiap kolom.
4. Melakukan proses perankingan untuk setiap alternatif (V_i) dengan cara mengalikan nilai bobor (w_i) dengan nilai rating kinerja ternormalisasi (r_{ij}).

2.5 Pengertian Beasiswa

Beasiswa adalah pemberian berupa bantuan keuangan yang diberikan kepada perorangan yang bertujuan untuk digunakan demi keberlangsungan pendidikan yang ditempuh. Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan. Pemberian beasiswa dapat dikategorikan pada pemberian cuma-cuma ataupun pemberian dengan ikatan kerja (biasa disebut ikatan dinas) setelah selesainya pendidikan. Lama ikatan dinas ini berbeda-beda, tergantung pada lembaga yang memberikan beasiswa tersebut.

2.6 Pengertian Basis Data

Basis data dapat didefinisikan dalam sejumlah sudut pandang. Menurut Fathansyah dalam bukunya yang berjudul Basis Data adalah (2004:2) :

1. Basis data merupakan himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Basis data merupakan kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian dan tanpa pengulangan (*redundansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Basis Data merupakan kumpulan *file*/tabel/arsip yang saling berhubunganyang disimpan dalam media penyimpanan elektronis.

Menurut Adi Nugroho (2004:5) basis data adalah kumpulan terorganisasi dari data-data yang berhubungan sedemikian rupa sehingga mudah disimpan, dimanipulasi serta dipanggil oleh pengguna. Dari pengertian diatas dapat disimpulkan bahwa database merupakan media penyimpanan data berupa *file*, tabel ataupun arsip-arsip yang diperlukan untuk memenuhi berbagai kebutuhan.

Adapun kegunaan basis data (*database*) yaitu digunakan untuk menjalankanfungsi pengelolaan suatu sistem informasi dan mengatasi permasalahan yang timbul pada penyimpanan data, diantaranya *redudansi* (penggandaan), *multiple user* (banyak pemakai), *security* (keamanan), dan *integritas* (kesatuan) data. 20 Sedangkan tujuan basis data menurut James martin (1975) terbagi menjadi dua kelompok (Edhy Sutanta 2004:2), yaitu :

1. Tujuan Primer

   Tujuan primer dimaksudkan sebagai tujuan utama yang ingin dicapai dalam usaha perancangan dan pengembangan basis data.

2. Tujuan Sekunder

   Tujuan tambahan yang dimaksudkan untuk mencapai tujuan primer.

   Keuntungan dari basis data adalah sebagai berikut :

   a. Kerangkapan data dapat diminimalkan.
   b. *Inkonsistensi* data dapat dihindari.
   c. Data dalam basis data dapat digunakan secara bersama (*multiuser*).
   d. Standarisasi dapat dilakukan.
   e. Pembatasan untuk keamanan data dapat diterapkan.
   f. Integritas data dapat terpelihara.
   g. Perbedaan kebutuhan data dapat diseimbangkan.

Operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data meliputi:

1. Pembuatan basis data baru (*create database*).
2. Penghapusan basis data (*drop database*).
3. Pembuatan *file* atau tabel dari suatu basis data (*create table*).
4. Penghapusan *file* atau tabel dari suatu basis data (*drop table*).
5. Penambahan atau pengisian data baru ke sebuah *file* atau tabel di sebuah basis data (*insert*).
6. Pengambilan data dari sebuah *file* atau tabel (*retrieve* atau *search*).

7. Pengubahan data dari sebuah *file* atau tabel (*update*).
8. Penghapusan data dari sebuah *file* atau tabel (*delete*).

2.7 Pengenalan Object Oriented

Menurut Suhendar dan Hariman (2002:10) Object oriented merupakan paradigma baru dalam rekayasa software yang didasarkan pada objek dan kelas. Object oriented memandang software bagian per bagian, dan menggambarkan suatu bagian dalam satu objek. Satu objek dalam sebuah model merupakan suatu fokus selama dalam proses analisis, desain dan implementasi dengan menekankan pada state, prilaku (behavior), dan interaksi objek-objek dalam model tersebut.

2.7.1 OOAD *(Object Oriented Analysis and Design)*

Object oriented analysis adalah metode analisis yang memeriksa keperluan requirements (syarat/keperluan yang harus dipenuhi suatu sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan. Object oriented design adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

2.7.2 Konsep dasar dalam *Object Oriented Analysis and Design (*OOAD)

1. Abstraksi

   Abstraction adalah prinsip mengabaikan sejumlah aspek dari suatu objek yang tidak relevan dengan tujuan tertentu untuk lebih memfokuskan pada objek tersebut secara utuh.

2. Objek

   Objek (object) adalah benda, secara fisik ataupun konseptual. Hardware,software, dokumen dan manusia adalah beberapa contoh dari objek. Sebuah objek memiliki keadaan sesaat (state) dan prilaku (behavior). State adalah kondisi objek yang menggambarkan objek tersebut, sedangkan behaviour adalah suatu definisi tindakan dan reaksi suatu objek. Atribut adalah nilai internal suatu objek yang mencerminkan antara lain karakteristik objek, kondisi sesaat, koneksi dengan objek lain, dan identitas.

   Behavior suatu objek mendefinisikan bagaimana sebuah objek bertindak dan member reaksi. Behavior ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan dalam objek itu sendiri. Behavior dari suatu objek dicerminkan oleh interface, service, dan method dari objek tersebut.

3. Kelas

   Kelas (class) adalah definisi umum untuk himpunan objek sejenis. Kelas menetapkan spesifikasi prilaku (behavior) dan atribut objek-objek tersebut. Class adalah abstraksi dari entitas dalam dunia nyata. Objek adalah contoh (instance) dari sebuah kelas.

4. Pemodulan (encapsulation)

   Encapsulation adalah prinsip yang digunakan ketika membangun keseluruhan struktur program, setiap komponen dari program harus dibungkus atau tersembunyi dalam satu desain keputusan.

5. Penurunan (inheritance)

   Inheritance adalah mekanisme untuk mengekspresikan kesamaan diantara kelas, memudahkan definisi dari kelas yang sama.

6. Kebanyak Rupaan (polymorphism)

   Polymorphism menunjukan bahwa ada banyak objek yang berasal dari kelas yang berbeda dapat bereaksi pada pesan yang sama.

7. Association

   Asosiasi adalah gabungan atau hubungan dari beberap ide. Merupakan landasan untuk, menentukan kompleksitas hubungan antara kelas objek.

2.8 UML (Unified Modelling Language)

2.8.1 Pengertian UML

Unified Modelling Language (UML) adalah sebuah “bahasa” yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna

tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch , metodologi coad , metodologi OOSE , metodologi OMT , metodologi shlaer-mellor , metodologi wirfs-brock , dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management.

2.8.2 Diagram-Diagram dalam UML

1. *Use case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2. *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 .

1. *Private,* tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.

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*.

Hubungan Antar *Class* :

a. 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*.
b. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
c. 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.
d. 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.

3. *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 *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart 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.

4. *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.

5. *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*.

6. *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.

7. *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.

8. *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.

2.9 Perangkat Lunak Pendukung

2.9.1 DBMS (*Database Management System)*

Sistem manajemen database atau database management system (DBMS) adalah merupakan suatu sistem software yang memungkinkan seorang user dapat mendefinisikan, membuat, dan memelihara serta menyediakan akses terkontrol terhadap data. Database sendiri adalah sekumpulan data yang berhubungan dengan secara logika dan memiliki beberapa arti yang saling berpautan. DBMS yang penulis gunakan dalam proyek ini adalah MySQL.

2.9.1.1 MySQL

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 di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, di mana 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 di bawah lisensi GPL (General Public License). Di mana 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), di bawah 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 atau unsigned, integer, float, double, char, text, date, timestamp dan lain-lain.

6. Command dan 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 dan limits

   MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta lima 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. Localisation

    MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari 20 bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Interface

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

12. Clients dan tools

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

13. Struktur tabel

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

2.9.2 PHP (*Hypertext Preprocessor*)

2.9.2.1 Pengertian PHP

PHP (akronim dari PHP: Hypertext Preprocessor) adalah bahasa pemrograman yang berfungsi untuk membuat website dinamis maupun aplikasi web.

Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan database, file dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah website. Blog, Toko Online, CMS, Forum, dan Website Social Networking adalah contoh aplikasi web yang bisa dibuat oleh PHP. PHP adalah bahasa scripting, bukan bahasa tag-based seperti HTML. PHP termasuk bahasa yang cross-platform, ini artinya PHP bisa berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran ".php".

2.9.2.2 Sejarah PHP

PHP ditulis (diciptakan) oleh Rasmus Lerdorf, seorang software engineer asal Greenland sekitar tahun 1995. Pada awalnya PHP digunakan Rasmus hanya sebagai pencatat jumlah pengunjung pada website pribadi beliau. Karena itu bahasa tersebut dinamakan Personal Home Page (PHP) Tools. Tetapi karena perkembangan nya yang cukup disukai oleh komunitas nya, maka beliau pun merilis bahasa PHP tersebut ke publik dengan lisensi open-source. Saat ini, PHP adalah server-side scripting yang paling banyak digunakan di website-website di seluruh dunia, dengan versi sudah mencapai versi 5 dan statistiknya terus bertambah (www.php.net/usage.php).

2.9.2.3 Syarat Untuk Menjalankan PHP

Untuk dapat berjalan, PHP membutuhkan web server, yang bertugas untuk memproses file-file php dan mengirimkan hasil pemrosesan untuk ditampilkan di

browser client. Oleh karena itu, PHP termasuk server-side scripting (script yang diproses di sisi server). Web server sendiri adalah software yang diinstall pada komputer lokal ataupun komputer lain yang berada di jaringan intranet / internet yang berfungsi untuk melayani permintaan-permintaan web dari client. Web server yang paling banyak digunakan saat ini untuk PHP adalah "Apache" (www.apache.org). Selain Apache, PHP juga memerlukan PHP binary (www.php.net) yang bisa dikonfigurasikan sebagai modul Apache atau pun sebagai aplikasi CGI. Untuk media penyimpanan datanya (database server), PHP biasa menggunakan "MySQL" (www.mysql.com).

Untuk menginstall dan mengkonfigurasi ketiga software tersebut (Apache, MySQL, PHP) agar dapat berjalan dan saling terhubung, memang cukup sulit. Maka dari itu dibuatlah paket software LAMP, XAMPP, MAMP, WAMP, dll yang tinggal kita install dalam satu kali installasi. Dalam satu kali installasi, sudah mencakup ketiga software tersebut dan sudah dikonfigurasi untuk keperluan lingkungan pengembangan aplikasi web. Sehingga, programmer web hanya tinggal menulis program PHP dan langsung menjalankan / mengetest program yang ditulis tersebut melalui web browser. Untuk mendapatkan paket software web server tersebut silakan download dari website yang bersangkutan (untuk XAMPP: www.apachefriends.org, dan untuk WampServer: www.wampserver.com/en/).

2.9.3 Adobe Dreamweaver CS 4

Menurut C. Widyo Hermawan, Sri Sulistiyani, Leo Agung dan Suci Nurasih (2009:1) Adobe Dreamweaver CS4 merupakan produk software Adobe yang digunakan sebagai HTML editor professional untuk mendesain web secara visual dan dapat juga digunakan untuk mengelola situs atau halaman web. Selain itu Adobe Dreamweaver CS4 memberikan keleluasaan kepada anda untuk menggunakan sebagai media penulisan pemrograman web. Dalam perkembangannya Dreamweaver banyak digunakan para web desainer maupun web programmer. Fasilitas optimal dalam jendela design yang tersedia menjadikan program ini sebuah produk unggulan dalam memberikan kemudahan dalam mendesain web, tidak terkecuali bagi para web desainer pemula.

Kemampuan Dreamweaver untuk berinteraksi dengan beberapa bahasa pemrograman seperti : PHP, ASP, JavaScript, dan sebagainya, juga merupakan fasilitas pendukung maksimal kepada para desainer web yang menyertakan bahasa pemrograman web dalam pekerjaannya. Adobe Dreamweaver Creative Suite 4 (CS4), merupakan web komersial dikenal juga editor yang memungkinkan Anda untuk merancang, membangun dan mengelola website a kompleks. Editor adalah yang berarti bahwa Anda dapat membuat halaman web Anda secara visual dan apa pun yang Anda lihat pada layar saat desain adalah apa yang akan Anda dapatkan ketika situs Anda dimuat dalam web browser normal.

2.10 Pengenalan Jaringan Komputer

Berdasarkan kriterianya, jaringan komputer dibedakan menjadi 4 yaitu:

1. Berdasarkan distribusi sumber informasi/data.
   a. Jaringan terpusat

      Jaringan ini terdiri dari komputer klient dan server yang mana komputer klient yang berfungsi sebagai perantara untuk mengakses sumber informasi/data yang berasal dari satu komputer server.

   b. Jaringan terdistribusi

      Merupakan perpaduan beberapa jaringan terpusat sehingga terdapat beberapa komputer server yang saling berhubungan dengan klient membentuk sistem jaringan tertentu.

2. Berdasarkan jangkauan geografis dibedakan menjadi:
   a. Jaringan LAN

      Merupakan jaringan yang menghubungkan 2 komputer atau lebih dalam cakupan seperti laboratorium, kantor, serta dalam 1 warnet.

   b. Jaringan MAN

      Merupakan jaringan yang mencakup satu kota besar beserta daerah setempat. Contohnya jaringan telepon lokal, sistem telepon seluler, serta jaringan relay beberapa ISP internet.

c. Jaringan WAN

Merupakan jaringan dengan cakupan seluruh dunia. Contohnya jaringan PT. Telkom, PT. Indosat, serta jaringan GSM Seluler seperti Satelindo, Telkomsel, dan masih banyak lagi.

3. Berdasarkan peranan dan hubungan tiap komputer dalam memproses data.

a. Jaringan Client-Server

Pada jaringan ini terdapat 1 atau beberapa komputer server dan komputer client. Komputer yang akan menjadi komputer server maupun menjadi komputer client dan diubah-ubah melalui *software* jaringan pada protokolnya. Komputer client sebagai perantara untuk dapat mengakses data pada komputer server sedangkan komputer server menyediakan informasi yang diperlukan oleh komputer client.

b. Jaringan Peer-to-peer

Pada jaringan ini tidak ada komputer client maupun komputer server karena semua komputer dapat melakukan pengiriman maupun penerimaan informasi sehingga semua komputer berfungsi sebagai client sekaligus sebagai server.

4. Berdasarkan media transmisi data

a. Jaringan Berkabel (*Wired Network*)

Pada jaringan ini, untuk menghubungkan satu komputer dengan komputer lain diperlukan penghubung berupa kabel jaringan. Kabel jaringan

berfungsi dalam mengirim informasi dalam bentuk sinyal listrik antar komputer jaringan.

b. Jaringan Nirkabel (*Wireless Network*)

Merupakan jaringan dengan medium berupa gelombang elektromagnetik. Pada jaringan ini tidak diperlukan kabel untuk menghubungkan antar komputer karena menggunakan gelombang elektromagnetik yang akan mengirimkan sinyal informasi antar komputer jaringan.