BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

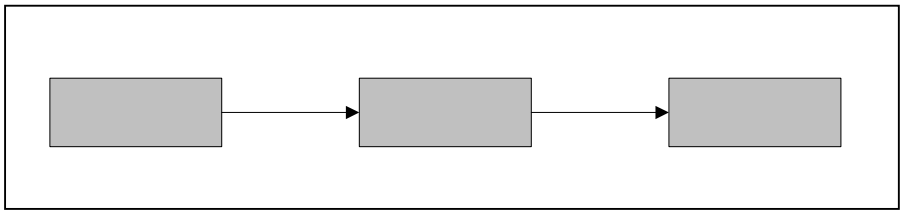
Pengertian sistem dapat dikelompokkan menjadi dua kelompok dalam mendenifisikan sistem yaitu penekanan pada prosedurnya dan menekankan pada komponen atau elemennya. Denifisi sistem itu sendiri yang menekankan pada komponen atau elemen memiliki pengertian yang lebih luas dari pada penekanan pada prosedur karena pengertian tersebut lebih diterima dikarenakan suatu sistem terdiri dari beberapa subsistem - subsistem.

Menurut [**Jog05**] sistem adalah kumpulan dari elemen – elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

Berdasarkan definisi diatas dapat dikemukakan bahwa sistem dibentuk dari bagian-bagian yang mempunyai tujuan yang sama dengan hubungan yang saling membutuhkan untuk mencapai tujuan tertentu. Pendefinisian sistem itu sendiri bisa di artikan berdasarkan penggunaan pendekatan komponen yang dilakukan oleh penggunanya.

2.1.1 Bentuk Dasar Sistem

Bentuk umum dari suatu sistem terdiri atas masukan (*input*), proses dan keluaran (*output*). Dalam bentuk umum sistem ini terdapat satu atau lebih masukan yang akan di proses dan akan menghasilkan suatu keluaran.



(Sumber **[Jog05]**)

Gambar 2.1 Bentuk Dasar Sistem

2.1.2 Karakteristik Sistem

[**Jog05**] Bahwa suatu sistem mempunyai karakteristik atau sifat – sifat tertentu, yaitu memiliki komponen – komponen (*components*), batas sistem (*boundary*), lingkungan sistem (*Environment*), penghubung (*Interface*), masukan (*Input*), keluaran (*Output*), pengolah (*Proses*), dan sasaran (*Objective*), dan tujuan (Goal).

1. Komponen sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem atau elemen –elemen sistem dapat berupa suatu subsistem atau bagain – bagian dari sistem. Setiap subsitem mempunyai sifat – sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengarui suatu sistem secara keseluruhan.

2. Batas Sistem (*boundary*)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem di pandang sebagai satu kesatuan. Batas suatu sistem menunjukan ruang lingkup dari sistem tersebut.

3. Lingkungan Sistem (*environment*)

   Linkungan luar dari sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan (harus dijaga dan merupakan energi dari sistem) dan dapat bersifat merugikan (harus ditahan dan dikendalikan).

4. Penghubung Sistem (*interface*)

   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya melalui penghubung, Dengan penghubung satu subsistem dapat berinteraksi yang lainnya membentuk satu kesatuan.

5. Masukan Sistem (*input*)

   Masukan (*input*) adalah energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (*signal input*). Maintenance input adalah energi yang masukan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem (*output*)

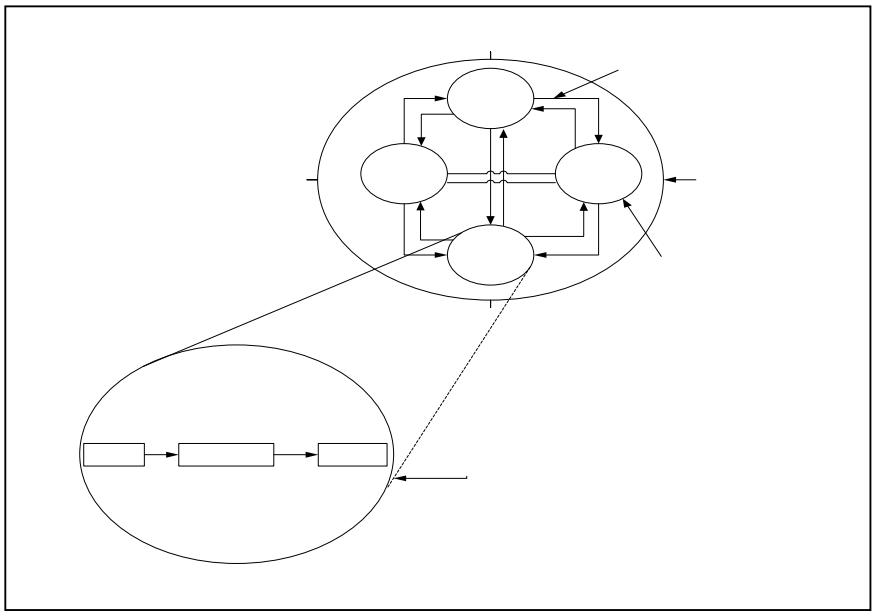
   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan subsistem yang lain atau kepada supersistem.

7. Pengolah Sistem

   Suatu sistem dapat mempunyai bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

   Suatu sistem pasti mempunya tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya Suatu sistem dikatakan berasil bila mengenai sasaran atau tujuan.



(Sumber: **[Jog05]**)

Gambar 2.2 Karakteristik Sistem

2.1.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya adalah sebagai berikut ini [**Jog05**].

1. Sistem diklasifikasikan berdasarkan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*).

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak nampak, misalnya sistem teologi. Sistem fisik adalah sistem yang ada secara fisik misalnya sistem komputer.

2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia.

   Sistem alamiah adalah sistem yang terjadi melalui proses alam dan tidak di buat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang di rancang oleh manusia yang melibatkan interaksi manusia dengan mesin yang disebut dengan *human-machine system* atau *man-machine system*.

3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*).

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya didekteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Misalnya sistem pada komputer. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat dipredisikan karena mengandung unsur probabilitas.

4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*).

   Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada

hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luar atau subsistem yang lainnya. Karena sistem sifatnya terbuka dan terpengaruh oleh lingkungan luarnya, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan secara otomatis dan terbuka hanya untuk pengaruh yang baik.

2.1.4 Pengertian Informasi

Informasi merupakan hasil dari data yang di proses yang berasal dari inputan suatu kejadian yang nyata yang berguna bagi pemakainya. Pengertian informasi di atas merupakan pengembangan teori yang di kemukakan oleh [**Jog05**] yang menyatakan bawah informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2.1.5 Konsep Dasar Informasi

Informasi merupakan hasil dari pengolahan yang di sajikan secara tepat dan akurat. Sumber dari informasi adalah data, data merupakan bentuk jamak dari bentuk tunggal atau data item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata kejadian-kejadian (*event*). Kejadian adalah suatu yang terjadi pada saat tertentu yang menyangkut perubahan nilai yang disebut transaksi. Sedangkan kesatuan nyata adalah berupa suatu objek yang nyata dan terjadi pada saat kejadian berlangsung. Data juga dapat diartikan suatu yang

perlu diolah terlebih dahulu untuk mendapatkan suatu informasi. Menurut [**Jog05**] kualitas informasi yang di harapkan tergantung 4 (empat) hal pokok yaitu:

1. Akurat

   Akurat mempunyai arti informasi yang dihasilkan harus bebas dari kesalahan–kesalahan, yang tidak biasa, tidak menyesatkan dan menceminkan maksudnya.

2. Tepat waktu

   Tepat waktu berarti informasi yang disampaikan ke penerima tidak terlambat, karena informasi adalah landasan untuk mengambil suatu keputusan. Untuk itu diperlukan suatu teknologi untuk dan mengirim dengan cepat dan tepat.

3. Relevan

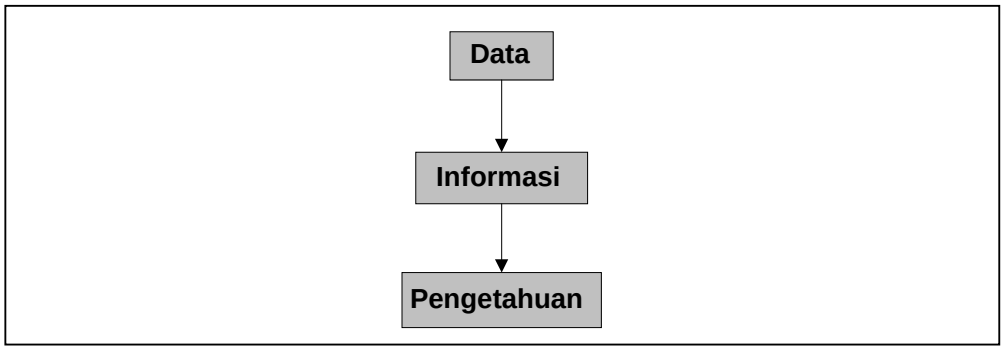
   Berarti informasi mempunyai manfaat dan berguna bagi pemakainya. Karena batas relevensi seseorang berbeda, maka informasi bisa dikatakan berguna jika benar – benar berguna dan dibutuhkan pemakainya.

4. Aman

   Aman berarti informasi harus terbebas dari penyadapan oleh pihak orang yang tidak berwenang dalam penggunaan informasi tersebut.

2.1.6 Hirarki Informasi

Informasi memiliki urutan atau tingkatan yang disebut sebagai hirarki makna informasi, yaitu tingkatan informasi dilihat dari kegunaan atau asas manfaatnya.



(Sumber: **[Jog05]**)

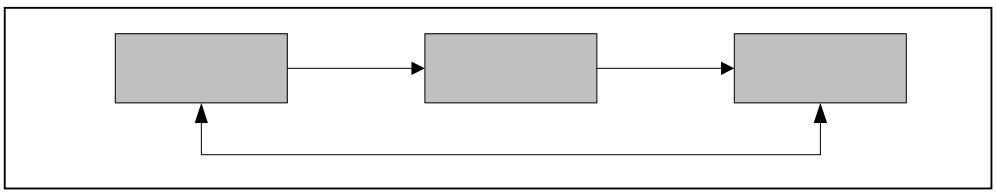
Gambar 2.3 Hirarki Informasi

Dari gambar diatas, menurut [**Jog05**] terjadi 3 (tiga) bagian utama, yaitu :

1. Data, yaitu gambaran dari suatu kejadian nyata.
2. Informasi, yaitu suatu hasil dari pengolahan data yang memiliki nilai guna bagi penerima yang membutuhkan.
3. Pengetahuan, yaitu nilai lebih yang didapat dari sebuat informasi.

2.1.7 Siklus Hidup Informasi

Dari penjelasan sebelumnya telah di kemukakan bahwa informasi adalah merupakan hasil dari pengolahan data. Supaya informasi yang didapat sesuai dengan yang diinginkan dan dapat melangsungkan hidupnya, maka diperlukan *feedback* atau umpan balik dari sistem yang berjalan.



(Sumber: **[Jog05]**)

Gambar 2. 4 Siklus Hidup Informasi

2.1.8 Sistem Informasi

Sistem yang ada berfungsi sebagai penghasil suatu keluaran, baik berupa informasi maupun berupa objek / benda. Untuk keluaran berupa informasi, maka sistem tersebut dikatakan sebagai sistem informasi.

2.1.9 Pengertian Sistem Informasi

Sistem informasi adalah sekumpulan komponen-komponen sistem yang berada didalam suatu ruang lingkup organisasi, saling berinteraksi untuk menghasilkan sebuah informasi yang bertujuan untuk pihak manajemen tertentu dan untuk mencapai tujuan tertentu.

Menurut [**Jog05**] faktor – faktor yang menentukan kehandalan dari suatu sistem informasi atau informasi dapat dikatakan baik jika memenuhi kriteria-kriteria sebagai berikut :

1. Keunggulan (*usefulness*)

   Yaitu suatu sistem yang harus dapat menghasilkan informasi yang tepat dan relevan untuk mengambil keputusan manajemen dan personil operasi dalam organisasi.

2. Ekonomis

   Kemampuan sistem yang mempengaruhi sistem harus bernilai manfaat minimal, sebesar biayanya.

3. Kehandalan (*Reliability*)

   Keluaran dari sistem harus mempunyai tingkat ketelitian tinggi dan sistem tersebut harus beroperasi secara efektif.

4. Pelayanan (*Customer Service*)

   Yakni suatu sistem memberikan pelayanan yang baik dan efisien kepada para pengguna sistem pada saat berhubungan dengan organisasi.

5. Kapasitas (*Capacity*)

   Setiap sistem harus mempunyai kapasitas yang memadai untuk menangani setiap periode sesuai yang dibutuhkan.

6. Sederhana dalam kemudahan (*Simplicity*)

   Sistem tersebut lebih sederhana (umum) sehingga struktur dan operasinya dapat dengan mudah dimengerti dan prosedure mudah diikuti.

7. Fleksibel (*Fleksibility*)

   Sistem informasi ini harus dapat digunakan dalam kondisi sebagaimana yang diinginkan oleh organisasi tersebut atau pengguna tertentu.

2.1.10 Komponen Sistem Informasi

Istilah dalam komponen sistem informasi adalah blok bangunan (*building block*) yang dapat di bagi menjadi enam blok [**Jog05**], yaitu :

1. Blok masukan (*input block*)

   Blok input merupakan data–data yang masuk ke dalam sistem informasi, yang dapat berupa dokumen-dokumen dasar yang dapat diolah menjadi suatu informasi tertentu.

2. Blok model (*model block*)

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan mengolah data input untuk menghasilkan suatu informasi yang dibutuhkan.

3. Blok keluaran (*output block*)

   Merupakan informasi yang menghasilkan sekumpulan data yang nantinya akan disimpan berupa data cetak laporan.

4. Blok teknologi (*technologi block*)

   Blok teknologi merupakan penunjang utama dalam berlangsunganya sistem informasi. Yang memiliki beberapa komponen yaitu alat memasukan data (*input device*), alat untuk menyimpan dan mengakses data (*storege device*), alat untuk menghasilkan dan mengirimkan keluaran (*output divice*) dan alat untuk membantuk pengendalian sistem secara keseluruan (*control device*). Teknologi informasi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*humanware* atau *brainare*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok basis data (*database block*)

   Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan oleh perangkat lunak untuk memanipulasinya. Data perlu di simpan dan perlu di organisasi sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

6. Blok kendali (*control block*)

   Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal–hal yang dapat merusak sistem dapat di cegah bila terlanjur terjadi kesalahan-kesalahan kerusakan dalam penggunaan sistem.

2.2 Pendekatan sistem berorientasi objek

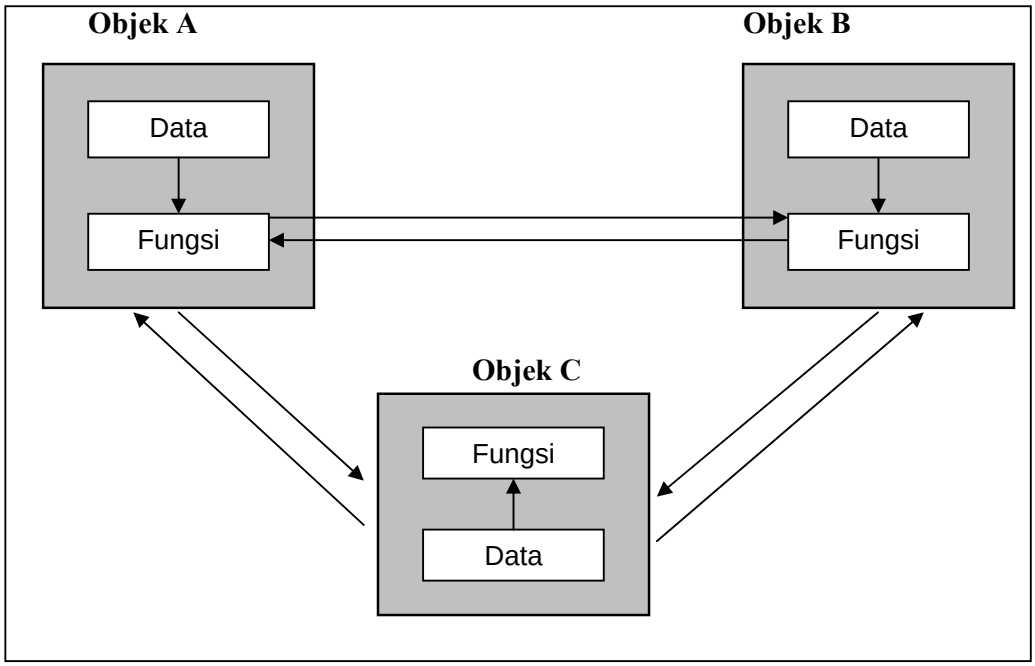
[**Ayi06**] Konsep dasar dari pengembangan berorientasi objek adalah Abstraksi, Enkapsulasi, Pewarisan (*inheritance*), *Reusability*, *Generalisasi* dan *Spesialisasi*, Komunikasi antar objek dan *Polymorphism.*

1) *Abstraksi*, prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.
2) *Enkapsulasi*, pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki objek, untuk menyembunyikan implementasi dari objek sehingga objek lain tidak mengetahui cara kerjanya.
3) Pewarisan (*inheritance*), mekanisme yang memungkinkan suatu objek/kelas mewarisi sebagian atau seluruh definisi dari objek/kelas lain sebagai bagian dari dirinya.
4) *Reusability*, pemanfaatan kembali objek yang telah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek itu.
5) *Generalisasi* dan *spesialisasi*, menunjukan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

6) Komunikasi antar objek, dilakukan lewat pesan (*message*) yang dikirim antar satu objek ke lainnya.
7) *Polymorphism*, kemampuan suatu suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

Menurut **[Adi05]** ada beberapa karakterisitik yang menjadi ciri-ciri dari pendekatan berorientasi objek adalah :

1) Pendekatan lebih pada data dan bukannya pada prosedur/fungsi.
2) Program besar dibagi pada apa yang dinamakan objek-objek.
3) Sturktur data dirancang dan menjadi karakteristik dari objek-objek.
4) Fungsi-fungsi yang mengoperasikan data tergabung dalam suatu objek yang sama.
5) Data tersembunyi dan terlidung dari prosedur/fungsi yang ada di luar.
6) Objek-objek dapat saling berkomunikasi dengan saling mengirim *message* (pesan) satu sama lain.
7) Pendekatan adalah dari bawah ke atas (*bottom up approach*).



(Sumber: **[Adi05]**)

Gambar 2.5 Pengorganisasian Data Serta Fungsi Pendekatan Berorientasi Objek

Pendekatan beorientasi objek membuat data terbungkus pada setiap fungsi/prosedur dan melindunginya terhadap perubahan yang tidak dikehendaki dari fungsi yang berada di luar.

2.2.1 Unified Modelling Language (UML)

Ada banyak sekali definisi *Unified Modeling Language* (UML) yang pada dasarnya memiliki maksud yang sama diantaranya :

[Fow05] *Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek.

[RJB99] *The unified modeling Language is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifact of a software system.*

[Lar04] *The Unified Modeling Language is a visual language for specifying, constructing and documenting the artifacts of systems.*

UML merupakan sebuah bahasa pemodelan visual yang digunakan untuk menspesifikasikan, membangun, memvisualisasikan dan mendokumentasikan artifak sistem perangkat lunak dalam paradigma pengembangan berorientasi objek. *Unified Modeling Language* (UML) menggabungkan Metode Booch, *Object Modelling Technique* (OMT), dan *Object Oriented Software Engineering* (OOSE). UML merupakan standar yang relative terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas sistem berorientasi objek.

UML bukanlah metode akan tetapi bahasa pemodelan. Terdapat perbedaan antara metode dan bahasa pemodelan. Metode adalah cara eksplisit yang menstrukturkan berfikir dan aksi seseorang. Metode memberitahukan ke pemakai mengenai apa yang dilakukan, bagaimana melakukan, kapan melakukan dan kenapa dilakukan (maksud aktivitas spesifik). Metode-metode menghasilkan model-model dan model-model ini digunakan untuk mendeskripsikan sesuatu dan mengkomunikasikan hasil-hasil dari penggunaan metode. Perbedaan utama antara

metode dan bahasa pemodelan adalah bahasa pemodelan tidak menyatakan proses atau instruksi untuk melakukan apa, bagaimana melakukan, kapan melakukan dan kenapa dilakukan.

Model diekspresikan dalam bahasa pemodelan. Bahasa pemodelan berisi notasi, yaitu simbol-simbol yang digunakan di model dan aturan-aturan yang menuntun bagaimana menggunakannya. Aturan-aturan ini termasuk sintaks, semantiks dan pragmatik.

2.2.2 Model-Model Diagram UML

UML terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang disebut sebuah model. Diagram diklasifikasikan berdasarkan tiga jenis diagram sebagai berikut.

Tabel 2.1
Klasifikasi Jenis Diagram UML

Structural Diagram	1. *Class Diagram* 2. *Composite Structure Diagram* 3. *Object Diagram* 4. *Component Diagram* 5. *Deployment Diagram* 6. *Package Diagram*
Behavioral Diagram	1. *Use Case Diagram* 2. *Activity Diagram* 3. *State Machine Diagram*
Interaction Diagram	1. *Sequence Diagram* 2. *Communication Diagram* 3. *Interaction Overview Diagram* 4. *Timing Diagram*

(Sumber: **[Fow05]**)

2.2.2.1 Diagram *Use Case*

Diagram *use case* adalah deskripsi fungsi dari sebuah sistem dari perspektif/sudut pandang para pengguna sistem. *Use case* mendefinisikan "apa" yang dilakukan oleh sistem dan elemen-elemennya, bukan "bagaimana" sistem dan elemen-elemennya saling berinteraksi. *Use case* bekerja dengan menggunakan "*scenario*", yaitu deskripsi urutan-urutan langkah yang menerangkan apa yang dilakukan pengguna terhadap sistem maupun sebaliknya. Diagram *use case* mengidentifikasikan fungsionalitas yang dipunyai oleh sistem (*use case*), user yang berinteraksi dengan sistem (*actor*) dan asosiasi/keterhubungan antara user dengan fungsionalitas sistem. Komponen notasi dasar yang dipunyai oleh diagram *use case* adalah *actor*, *use-case*, dan *association*. Berikut adalah notasi yang terdapat pada diagram *use case*. Diagram *use case* memiliki sebuah model khusus yang terbatas untuk kondisi tertentu yang disebut *stereotype*. Untuk menunjukkan *stereotype* digunakan symbol "<<"

diawalnya dan ditutup dengan “>>” diakhirnya. Terdapat dua *stereotype* paling sering digunakan dalam *use case diagram* yaitu <<extend>> dan <<include>>. <<extend>> digunakan untuk menunjukkan bahwa satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu dipenuhi. Sedangkan <<include>> digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.

2.2.2.2 Diagram *Activity*

Diagram *activity* digunakan untuk mendokumentasikan alur kerja pada sebuah sistem, yang dimulai dari pandangan *business level* hingga ke *operational level*. Pada dasarnya, diagram *activity* merupakan variasi dari diagram *state machine*. Diagram *activity* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah diagram *activity* bisa mendukung perilaku *parallel* sedangkan *flowchart* tidak bisa.

2.2.2.3 Diagram *Sequence*

Diagram *Sequence* mendokumentasikan komunikasi/interaksi antar kelas-kelas. Diagram ini menunjukkan sejumlah objek dan *message* (pesan) – yang diletakkan diantara objek-objek didalam *use case*. Perlu diingat bahwa di dalam diagram ini, kelas-kelas dan aktor-aktor diletakkan dibagian atas diagram dengan urutan dari kiri ke kanan dengan garis *lifeline* yang diletakkan secara vertikal terhadap kelas dan aktor.

2.2.2.4 Diagram *Class*

Diagram *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat pada mereka. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

2.2.2.5 Diagram *Component*

Komponen perangkat lunak adalah bagian fisik dari sebuah sistem yang menetap di komputer. komponen merupakan implementasi software dari sebuah *class*. Komponen bisa berupa tabel, *file* data, *file* exe, *file* DLL, dokumen, dll.

Diagram c*omponent* mengandung komponen, *interface* dan *relationship*. Komponen diagram ini digunakan pada saat pengembang ingin memecah sistem menjadi komponen-komponen dan ingin menampilkan hubungan-hubungan mereka dengan antarmuka atau pemecahan komponen menjadi struktur yang lebih rendah. Secara umum dapat disimpulkan bahwa *component* diagram yang digunakan untuk menjelaskan kebergantungan antar beragam komponen-komponen *software* seperti misalnya kebergantungan antara *file-file executable* dengan *file-file* sumbernya (*source file*) dll.

2.2.2.6 Diagram *Deployment*

Diagram *deployment* menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian

hardware yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut. Diagram *deployment* dapat digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem.

2.2.2.7 Diagram *State(State Machine Diagram)*

State Machine Diagram menelusuri individu-individu objek melalui keseluruhan daur hidupnya, menspesifikasi semua urutan yang mungkin dari pesan-pesan yang akan diterima objek tersebut bersama-sama dengan tanggapan atas pesan-pesan tersebut. Diagram *State* menggambarkan transisi dan perubahan keadaan suatu objek dalam sistem sebagai akibat dari stimuli yang diterima. Pada umumnya diagram ini menggambarkan *class* tertentu. *State* diagram membantu analis, perancang dan pengembang untuk memahami perilaku objek dalam sistem.

2.2.2.8 Diagram *Composite* (*Composite Structure Diagram)*

Composite Structure Diagram adalah diagram untuk menunjukkan dekomposisi secara hierarkis sebuah *class* ke sebuah struktur internal. Hal ini memungkinkan untuk memecah objek yang kompleks menjadi bagian-bagian yang kecil.

2.2.2.9 Diagram *Object*

Object diagram merupakan sebuah gambaran tentang objek-objek dalam sebuah sistem pada satu titik waktu. Karena lebih menonjolkan perintah-perintah

daripada *class*, *object* diagram lebih sering disebut sebagai sebuah diagram perintah.

2.2.2.10 Diagram *Package*

Package diagram adalah sebuah bentuk pengelompokan yang memungkinkan untuk mengambil setiap bentuk di UML dan mengelompokkan elemen-elemen dalam tingkatan unit yang lebih tinggi.

2.2.2.11 Diagram *Communication*

Communication diagram sejenis dengan diagram interaksi, yang lebih menekankan pada link data diantara bermacam-macam partisipan pada interaksi tersebut.

2.2.2.12 Diagram *Interaction Overview*

Interaction Overview Diagram adalah pencangkokan secara bersama antara activity diagram dengan sequence diagram. *Interaction Overview* Diagram dapat dianggap sebagai *activity* diagram dimana semua aktivitas diganti dengan sedikit *sequence* diagram, atau bisa juga dianggap sebagai *sequence* diagram yang dirincikan dengan notasi *activity* diagram yang digunakan untuk menunjukkan aliran pengawasan.

2.2.2.13 Diagram *Timing*

Timing Diagram adalah bentuk lain dari *interaction* diagram, dimana fokus utamanya lebih ke waktu. Timing diagram sangat berdaya guna dalam menunjukkan faktor pembatas waktu diantara perubahan *state* pada objek yang berbeda.

2.2.3 Perangkat Lunak Pendukung Pemodelan UML

Hasil-hasil pemodelan yang dikomunikasikan dengan pelanggan agar lebih baik dan lebih profesional jika diformat dengan baik menggunakan *software* (*case tool*) yang bisa menyajikan gambar-gambar hasil pemodelan dengan rapi. *Software* yang digunakan adalah *Enterprise Architect* 7.0 dari *Sparx System*. Enterprise Architect 7.0 adalah sebuah perangkat lunak yang digunakan untuk membuat dokumentasi, mendesain sebuah sistem/perangkat lunak yang menggunakan UML.

2.2.3.1 Sekilas Tentang *Enterprise Architect 7.0*

Salah satu *case tool* yang cukup lengkap untuk melakukan pemodelan dengan UML adalah *Enterprise Architect* dari *Sparx System*. Hampir seluruh jenis diagram pemodelan UML 2.0 terdapat dalam *Enterprise Architect*. *Layout software* juga mirip dengan *layout* dari sebuah IDE, dimana *toolbox* ditempatkan di sebelah kiri, *project explorer* di sebelah kanan, dan *content* ditempatkan di tengah. Kemiripan *layout* ini tentunya memudahkan pengguna terutama *programmer* untuk cepat familiar dengan *software* ini.

Kelebihan-kelebihan software ini di antaranya adalah: sangat lengkap, fleksibilitas penggunaan, transformasi dari satu model ke model lain, korelasi satu model dengan model lain, *auto generate* model ke dalam dua platform yang saat ini sedang terkenal: .NET dan Java, *reverse engineering* dari kode sumber ke model, dan fitur-fitur pendukung lain seperti *auto generate* RTF dan HTML report, serta transformasi dari sebuah model menjadi image: JPG, GIF, atau BMP.

Sedangkan kekurangannya adalah perintah-perintah atau pilihan-pilihan yang terlalu kompleks dan tidak mudah untuk segera dipahami. Untuk bisa memahami seluruh notasi dan istilah-istilah yang dipakai, terlebih dahulu harus mengetahui konsep dasar UML.

2.3 Pengertian Perwalian dan Registrasi

Dibawah ini akan dijelaskan pengertian mengenai perwalian dan registrasi yang berlaku dalam Jurusan Manajemen Informatika Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia (UNIKOM) Bandung.

2.3.1 Perwalian dan Dosen Wali

Perwalian adalah kegiatan konsultasi antara Dosen Wali dengan Mahasiswa yang dilaksanakan di Kampus UNIKOM. Sedangkan Dosen Wali adalah dosen pembimbing akademik dari suatu jurusan. Tugas yang dimiliki dosen wali diantaranya sebagai berikut:

1) Memeriksa kelengkapan syarat-syarat administratif dan keuangan yang harus dipenuhi oleh mahasiswa yang dibimbingnya agar dapat dengan

lancar mengikuti program pendidikan dalam semester yang sedang berlangsung.

2) Mengarahkan mahasiswa dalam mengisi program studinya, antara lain dalam mengisi Formulir Rencana Studi (FRS) atau Formulir Perubahan Rencana Studi atau memberikan pertimbangan mengenai mata kuliah apa saja yang seyogyanya diambil untuk semester yang sedang berjalan.
3) Memberikan pertimbangan kepada mahasiswa mengenai berapa banyak SKS yang seharusnya diambil mahasiswa dengan memperhatikan IP dan IPK yang telah dicapai.
4) Menandatangani Kartu Rencana Studi (KRS) yang telah disusun oleh mahasiswa dan disetujui dosen wali.
5) Menandatangani Kartu Hasil Studi (KHS) atas nilai-nilai yang diperoleh mahasiswa. Mengikuti perkembangan studi setiap mahasiswa bimbingannya sehingga dapat diantisipasi sedini mungkin masalah dalam studi mereka.
6) Memberikan konsultasi kepada mahasiswa bimbingannya yang menghadapi kesulitan dalam menyelesaikan studinya dan kalau memerlukan bimbingan dan konseling yang lebih intensif dapat meneruskannya kepada Biro Bimbingan dan Konseling UNIKOM.

Tabel 2.2 Perhitungan Jumlah SKS yang dapat diambil

Indeks Prestasi	**Jumlah SKS yang dapat diambil**
>= 3.00	19 - 24
< 3.00	19

2.3.2 Pengisian KRS (Kartu Rencana Studi)

Setiap awal semester, mahasiswa wajib mengajukan rencana kegiatan akademik dengan mengisi KRS, Termasuk mahasiswa yang hanya sedang mengerjakan Skripsi/Tugas Akhir. Pengajuan rencana kegiatan akademik dilakukan mahasiswa secara langsung dengan memasukkan mata kuliah yang akan diambil ke dalam Kartu Rencana Studi (KRS) pada saat perwalian.

2.3.3 Indeks Prestasi (IP)

Indeks Prestasi adalah penilaian keberhasilan seorang mahasiswa yang dinyatakan dengan nilai kredit rata-rata yang merupakan satuan nilai akhir yang menggambarkan mutu penyelesaian suatu program studi. Indeks Prestasi dihitung pada setiap akhir semester. Indeks Prestasi mahasiswa dihitung berdasarkan pembobotan pada Indeks Nilai sebagai berikut :

Tabel 2.3 Indeks Nilai dan Bobot

Indeks	Bobot
A	4
B	3
C	2
D	1
E	0

Perhitungan Indeks Prestasi (IP) sebagai berikut :

$$IP = \frac{\Sigma(\text{Bobot X SKS})}{\Sigma \text{ SKS}}$$

Gambar 2.6 Perhitungan Indeks Prestasi

Indeks Prestasi Semester (IPS) , dihitung dari nilai mata kuliah untuk semua mata kuliah yang diambil pada satu semester, dengan diberi bobot SKS untuk tiap mata kuliah yang bersangkutan.

2.3.4 Indeks Prestasi Kumulatif (IPK)

Indeks Prestasi Kumulatif (IPK) merupakan angka yang menunjukkan Prestasi Akademik atau kemajuan belajar mahasiswa secara kumulatif yang dicapai mulai dari semester pertama sampai dengan semester paling akhir untuk semua mata kuliah yang ditempuh.

IPK digunakan sebagai kriteria dalam pemberian Sanksi Akademik dan evaluasi studi pada akhir program. Seorang mahasiswa dapat dinyatakan lulus

dari Program Strata-I, Diploma-III atau Diploma-I apabila telah memiliki IPK > 2.00 tanpa Nilai E.

Perhitungan Indeks Prestasi Kumulatif, sebagai berikut :

$$IPK = \frac{\Sigma \text{ (Bobot X SKS) seluruh semester}}{\Sigma \text{ SKS seluruh semester}}$$

Gambar 2.7 Perhitungan Indeks Prestasi Kumulatif

2.3.5 Registrasi Ulang Mahasiswa

Registrasi Ulang adalah proses pendaftaran kembali mahasiswa yang akan aktif pada setiap awal pergantian semester. Setiap mahasiswa wajib melakukan Registrasi Ulang sesuai dengan ketentuan-ketentuan yang berlaku untuk memenuhi persyaratan akademik dan administrasi keuangan. Registrasi Ulang dilakukan pada sekertariat jurusan masing-masing sesuai dengan jadwal yang telah ditentukan oleh kampus.

Mahasiswa yang tidak melaksanakan Registrasi Ulang dapat mengajukan cuti akademik atau akan dianggap sebagai mahasiswa non aktif pada semester yang bersangkutan dan kepadanya dikenakan sanksi :

1) Tidak berhak memperoleh pelayanan akademik.
2) Mahasiswa yang tidak Registrasi Ulang selama dua semester berturut-turut tanpa alasan yang jelas dianggap mengundurkan diri dari Universitas Komputer Indonesia.

3) Jika Mahasiswa yang dimaksud point 2 bermaksud melakukan Registrasi Ulang, persyaratan administrasi keuangannya sesuai dengan mahasiswa baru.

2.4 Objek-objek yang berhubungan dalam pembuatan *web*

Dalam pembuatan *web* terdapat banyak sekali objek-objek yang saling berhubungan mulai dari tempat desain tampilan, tempat untuk meletakkan *script* programnya, hingga fasilitas yang menghubungkan *web* dengan *internet*. Objek-objek tersebut diantara lain internet, PHP, MySQL, HTML dan program aplikasi Macromedia Dreamweaver 8.

2.4.1 Internet

Pengertian internet secara umum adalah sebuah jaringan komputer yang terdiri dari berbagai macam ukuran jaringan komputer diseluruh dunia mulai dari sebuah PC, Jaringan lokal berskala kecil, jaringan kelas menengah hingga jaringan – jaringan utama yang menjadi tulang punggung internet.

2.4.1.1 Sejarah Internet

Sekitar tahun 1963, RAND *Coorporation*, organisasi pengendali perang dingin Amerika Serikat menghadapi masalah yang sangat pelik yaitu bagaimana Amerika Serikat tetap berkomunikasi dengan lancar setelah perang nuklir terjadi. Pemecahannya adalah dengan membangun sebuah jaringan yang menghubungkan semua tempat strategis diseluruh Amerika dan tetap memberikan *Request For*

Proposal (RFP) kepada UCLA (*University of California Los Angles*). Mereka yang terlibat antara lain Vinton Crf, Stefen Croackern, Jon Postel dan Robert Branden.

Pada tahun 1964 dikeluarkan proposal RAND yang intinya adalah bahwa jaringan yang akan dibentuk tidak berpusat pada suatu dan tetap berfungsi sekalipun dalam keadaan hancur. Proposal RAND ini juga diilhami munculnya "NET" diawal tahun 1990-an yaitu suatu teknologi dimana pesan disampaikan dari tempat asalnya ketempat tujuan dengan memfragmentasikan pesan tersebut menjadi bagian – bagian kecil yang disebut paket ditempat asalnya dan dirangkai kembali menjadi pesan semula ditempat tujuan. Teknologi ini dikenal sebagai *packet switching network.*

Tahun 1969 empat buah IMP (*Interface Message Processor*) dikirim kepada 4 perguruan tinggi yakni UCLA, SRI *(Standard Reseach Institute),* UCSB (University of California

2.4.2 HTML

Penulisan HTML biasanya diawali dengan simbol <…> dan diakhiri dengan simbol </…>. Berikut struktur dasar HTML:

```
<HTML>
<HEAD>
<TITLE>……</TITLE>
</HEAD>
<BODY>
……
</BODY>
</HTML>
```

Berikut adalah penjelasan struktur dasar HTML di atas:

a. Pasangan tag <HTML> dan </HTML> menandakan bahwa kode yang terdapat di dalamnya adalah kode HTML, sehingga browser akan menterjemahkannya sebagai dokumen HTML.
b. Bagian yang terdapat dalam <HTML> dan </HTML> umunya terbagi atas kepala dan badan.
c. Bagian kepala ditandai dengan pasangan <HEAD> dan </HEAD>. Di dalam tag <HEAD> dan </HEAD> dapat dimasukkan tag <TITLE> dan </TITLE> yang isinya adalah judul yang akan muncul pada title bar web browser.
d. Pada bagian badan yang ditandai dengan <BODY> dan </BODY> dapat dimasukkan atau diisi dengan berbagai macam teks, gambar, atau apapun yang akan ditampilkan pada web.

Adapun cara penulisan nama tag, bisa memakai huruf kecil seperti <html> ataupun dapat juga ditulis dengan huruf kapital <HTML>, atau bisa juga ditulis menggunakan kombinasi antara huruf kecil dab huruf kapital <Html>. Dengan kata lain penulisan dianggap sama.

2.4.3 Sekilas tentang *Personal Home Page* (PHP)

PHP adalah *script* yang dijalankan diserver. Jadi konsepnya berbeda dengan JavaScript yang dijalankan di klien. Hal menarik yang didukung oleh PHP tetapi tidak didukung oleh JavaScript adalah kenyataan bahwa PHP bisa

digunakan untuk mengakses berbagai macam *database* seperti access, oracle, MySQL, dan lain-lain.

Jika bermaksud untuk membuat halaman web yang menggunakan PHP, harus mengaktifkan terlebih dahulu web server. Tentu saja bukan web server yang terhubung dengan internet, untuk mengujinya dapat digunakan sebuah komputer yang bertindak sebagai server dan sekaligus sebagai klien.

2.4.3.1 Kelebihan PHP

1. Bahasa pemrograman php merupakan sebuah bahasa *script* yang tidak memerlukan kompilasi dalam penggunaannya
2. Server web yang mendukung file php dapat ditemukan dimana – mana, dimulai dari IIS hingga Apache serta konfigurasi yang relatif mudah.
3. Dalam sisi pemahaman, php adalah bahasa *scripting* yang paling mudah karena referensinya banyak
4. Dalam sisi pengembangan, lebih mudah karena banyak milis – milis dan *developer* yang siap membantu dalam pengembangannya.

2.4.4 Apache

Pada awal mulanya, Apache merupakan perangkat lunak sumber terbuka yang menjadi alternatif dari *server web Netscape* (sekarang dikenal sebagai *Sun Java System Web Server*).

Asal mula nama Apache berasal ketika sebuah *server web* populer yang dikembangkan pada awal 1995 yang bernama *NCSA HTTPd* 1.3 memiliki

sejumlah perubahan besar terhadap kode sumbernya *(patch)*. Karena begitu banyaknya *patch* pada perangkat lunak tersebut sehingga disebut sebuah *server* yang memiliki banyak *patch ("a patchy" server).* Tetapi pada halaman *FAQ* situs web resminya, disebutkan bahwa *"Apache"* dipilih untuk menghormati suku asli Indian Amerika *Apache (Indé),* yang dikenal karena keahlian dan strategi perangnya. Versi 2 dari Apache ditulis dari awal tanpa mengandung kode sumber dari NCSA.

Selain itu Apache juga berguna untuk menjadi server lokal di komputer user. Apabila membuat web di dalam komputer, maka dibutuhkan apache agar web yang telah dibuat dapat dibuka di dalam browser.

2.4.5 MySQL

Inilah salah satu database server yang paling populer dan sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya, dan mampu menampung sampai bermiliaran data di dalamnya.

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

2.4.6 Macromedia Dreamweaver 8

Dreamweaver 8 merupakan salah satu software unggulan macromedia yang dapat membantu dalam pembuatan web. Macromedia Dreamweaver 8 merupakan pengembangan dari *web editor* macromedia Dreamweaver mx.