

BAB II

LANDASAN TEORI

1.1 Konsep Dasar Sistem

Konsep dasar sistem ada dua pendekatan yaitu penekanan pada prosedurnya dan penekanan pada komponennya.

1.1.1 Defenisi Sistem

Menurut Jogiyanto (2005) Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

1.1.2 Karakteristik Sistem

Jogiyanto (2005: 3) mengemukakan sistem mempunyai karakteristik atau sifat-sifat tertentu, yakni :

1. Komponen

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

2. Batasan sistem.

Batasan sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem.

Lingkungan luar (*evinronment*) dari suatu sistem adalah apapun diluar batas sistem yang mempengaruhi operasi. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

Lingkungan luar yang menguntungkan berupa energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

1.1.3 Komponen Sistem

Komponen sistem memiliki 5 macam :

1. Perangkat keras (CPU, disk, printer, tape).
2. Perangkat lunak (sistem operasi, sistem database, program pengontrol komunikasi, program aplikasi).
3. Personil (yang mengoperasikan sistem, menyediakan masukan, mengkonsumsi keluaran dan melakukan aktivitas manual yang mendukung sistem).
4. Data (yang harus tersimpan dalam sistem selama jangka waktu tertentu).
5. Prosedur (instruksi dan kebijakan untuk mengoperasikan sistem).

1.2 Konsep Dasar Informasi

Didalam organisasi sangat penting dalam mengelola sumberdaya-sumberdaya utama seperti buruh, dan bahan mentah, tapi saat ini informasi juga merupakan sumberdaya yang tidak kalah pentingnya harus dikelola. Para pembuat keputusan memahami bahwa informasi tidak hanya sekedar produk sampingan bisnis yang sedang berjalan, namun juga sebagai bahan pengisi bisnis dan menjadi faktor kritis dalam menentukan kesuksesan atau kegagalan suatu usaha.

Informasi ibarat darah yang mengalir di dalam tubuh suatu organisasi. Sehingga informasi merupakan salah satu bentuk sumber daya utama dalam suatu

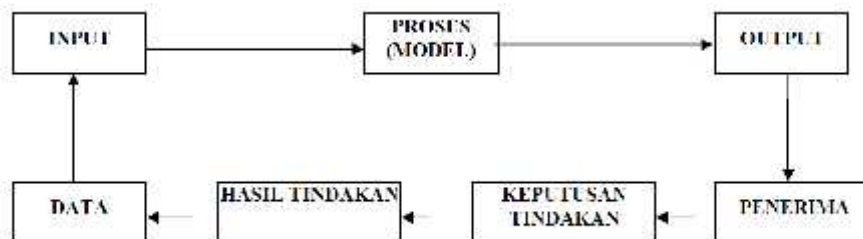
organisasi yang digunakan oleh manager untuk mengendalikan perusahaan dalam mencapai tujuan.

1.2.1 Defenisi Informasi

Menurut Jogianto, (2005) Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penggunanya. Informasi juga dapat diartikan sebagai data yang telah diolah menjadi bentuk yang mempunyai arti dan manfaat bagi manusia. Sedangkan data adalah aliran fakta mentah yang menunjukkan peristiwa yang telah terjadi dalam organisasi dan lingkungan fisik sebelum diorganisir dan ditata menjadi suatu bentuk yang bisa dipahamidan digunakan.

1.2.2 Siklus Informasi

Siklus **informasi adalah** gambaran secara umum mengenai proses terhadap data sehingga menjadi informasi yang bermanfaat bagi pengguna. Informasi yang menghasilkan informasi berikutnya. Demikian seterusnya proses pengolahan data menjadi informasi.



Gambar 2.1 : Siklus Informasi

Menurut Tata Sutabri, (2004) Data merupakan bentuk mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data ditangkap sebagai *input*, diproses melalui suatu model membentuk informasi. Pemakai kemudian menerima informasi tersebut sebagai landasan untuk membuat suatu keputusan dan melakukan tindakan operasional yang akan membuat sejumlah data baru.

Data baru tersebut selanjutnya menjadi *input* pada proses berikutnya, begitu seterusnya sehingga membentuk suatu siklus informasi/*Information Cycle*.

1.2.3 Kualitas Informasi

Kualitas Informasi ; tergantung dari 3 hal, yaitu :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.
2. Tetap pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

1.3 Konsep Dasar Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

1.3.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building blok*), yang terdiri dari komponen input, komponen model, komponen output, komponen teknologi, komponen hardware, komponen software, komponen basis data, dan komponen kontrol. Semua komponen tersebut

saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Komponen *input*

Input mewakili data yang masuk kedalam sistem informasi. *Input* disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Komponen model

Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen *output*

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen teknologi

Teknologi merupakan "*tool box*" dalam sistem informasi, Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen *hardware*

Hardware berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi. Yang berfungsi sebagai tempat untuk menampung database atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen *software*

Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.

7. Komponen basis data

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*).

8. Komponen kontrol

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

1.3.2 Jenis-Jenis Sistem Informasi

Sistem informasi dikembangkan untuk tujuan yang berbeda-beda, tergantung pada kebutuhan bisnis.

1. *Transaction Processing Systems* (TPS)

TPS adalah sistem informasi yang terkomputerisasi yang dikembangkan untuk memproses data dalam jumlah besar untuk transaksi bisnis rutin seperti daftar gaji dan inventarisasi. TPS berfungsi pada level organisasi yang memungkinkan organisasi bisa berinteraksi dengan lingkungan eksternal. Data yang dihasilkan oleh TPS dapat dilihat atau digunakan oleh manajer.

2. Sistem Informasi Manajemen (SIM)

SIM tidak menggantikan TPS, tetapi mendukung spektrum tugas-tugas organisasional yang lebih luas dari TPS, termasuk analisis keputusan dan

pembuat keputusan. SIM menghasilkan informasi yang digunakan untuk membuat keputusan, dan juga dapat membantu menyatukan beberapa fungsi informasi bisnis yang sudah terkomputerisasi (basis data).

1.4 Analisa Sistem

Menurut Yogyanto (1995) analisis sistem adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

Menurut Kristanto (2003) analisis sistem adalah suatu proses mengumpulkan dan menginterpretasikan kenyataan-kenyataan yang ada, mendiagnosa persoalan dan menggunakan keduanya untuk memperbaiki sistem.

Langkah-langkah dasar yang harus dilakukan oleh analisis sistem adalah :

1. *Identify*, mengidentifikasi masalah
2. *Understand*, memahami kerja sistem yang ada
3. *Analyze*, menganalisis sistem
4. *Report*, membuat laporan hasil analisis

1.5 Perancangan Sistem

Perancangan sistem adalah merancang atau mendesain suatu sistem yang baik, yang isinya adalah langkah-langkah operasi dalam proses pengolahan data dan prosedur untuk mendukung operasi sistem.

Menurut Jogiyanto. HM,(1991). Perancangan sistem dapat diartikan sebagai berikut :

1. Tahap setelah analisis dari siklus pengembangan sistem
2. Pendefinisian dari kebutuhan-kebutuhan fungsional
3. Persiapan untuk rancang bangun implementasi
4. Menggambarkan bagaimana suatu sistem dibentuk

5. Yang dapat berupa penggambaran perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
6. Termasuk menyangkut mengkonfigurasi dari komponen perangkat keras dari suatu sistem.

1.6 Basis Data

Basis Data adalah sekumpulan data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah struktur tertentu, tersimpan di hardware komputer dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu.

Komponen sebuah sistem basis data antara lain :

1. *Hardware*, Biasanya berupa perangkat computer standar, media penyimpan sekunder dan media komunikasi untuk system jaringan.
2. *Operating System*, Yakni merupakan perangkat lunak yang memfungsikan, mengendalikan seluruh sumber daya dan melakukan operasi dasar dalam system komputer. Harus sesuai dengan DBMS yang digunakan.
3. *Database*, Yakni basis data yang mewakili system tertentu untuk dikelola. Sebuah sistem basis data bias terdiri dari lebih dari satu basis data.
4. *DBMS (Database Management System)*, erang katlunak yang digunakan untuk mengelola basis data. Contoh kelas sederhana: dBase, Foxbase, Rbase, MS. Access, MS. Foxpro, Borland Paradox. Contoh kelas kompleks: Borland-Interbase, MS. SQL Server, Oracle, Informix, Sybase.

Sebuah *DBMS (Database Management System)* umumnya memiliki sejumlah komponen fungsional (modul) seperti :

- a) *File Manager*, yang mengelola ruang dalam disk dan struktur data yang dipakai untuk merepresentasikan informasi yang tersimpan dalam disk.

- b) Database Manager, yang menyediakan *interface* antara data low-level yang ada di basis data dengan program aplikasi dan *query* yang diberikan ke sistem.
 - c) *Query Processor*, yang menterjemahkan perintah-perintah dalam *query language* ke perintah low-level yang dapat dimengerti oleh database manager.
 - d) *Data Modeling LanguagePrecompiler*, yang mengkonversi perintah DML yang ditambahkan dalam sebuah program aplikasi kepemanggil prosedur normal dalam bahasa induk.
 - e) *Data Definition LanguageCompiler*, yang mengkonversi perintah-perintah DDL ke dalam sekumpulan tabel yang mengandung metadata. Tabel-tabel ini kemudian disimpan dalam kamus data.
5. User (Pengguna Sistem Basis Data), Orang-orang yang berinteraksi dengan sistem basis data, mulaidari yang merancangsampai yang menggunakan di tingkataakhir.
6. *Optional Software*, Perangkat lunak pelengkap yang mendukung. Bersifat opsional.

1.7 Manajemen Aset

Manajemen aset merupakan suatu ilmu yang dibutuhkan bagi setiap entitas/instansi/organisasi. Dengan penerapan manajemen aset maka setiap entitas bisa memanfaatkan aset yang mereka miliki dengan optimal/sesuai dengan tupoksinya. Untuk mengoptimalkan aset tersebut maka alangkah lebih baik dijelaskan terlebih dahulu pengertian dari manajemen, aset, dan manajemen aset.

1.7.1 Definisi Manajemen Aset

Menurut Siregar (2004) “Manajemen Aset merupakan salah satu profesi atau keahlian yang belum sepenuhnya berkembang dan populer di lingkungan pemerintahan maupun di satuan kerja atau instansi”.

1.7.2 Konsep Manajemen Aset

Menurut Siregar (2004), di dunia internasional manajemen aset telah berkembang cukup pesat, namun di Indonesia hal ini khususnya dalam konteks pengelolaan aset pemerintah daerah sepenuhnya belum dipahami oleh para pengelola daerah. Manajemen aset pemerintah daerah dapat dibagi dalam empat tahap kerja yang meliputi; inventarisasi aset, legal audit, penilaian aset, dan optimalisasi aset, di mana kelima tahapan tersebut adalah saling berhubungan dan terintegrasi satu dengan yang lainnya.



Gambar 2.2 Tahapan Manajemen Aset

1. Inventarisasi aset.

Inventarisasi Aset merupakan kegiatan yang terdiri dari dua aspek, yaitu inventarisasi fisik dan yuridis/legal. Aspek fisik terdiri atas bentuk, luas, lokasi, volume/jumlah, jenis, alamat dan lain-lain. Sedangkan aspek yuridis adalah status penguasaan, masalah legal yang dimiliki, batas akhir penguasaan. Proses kerja yang dilakukan adalah pendataan, kodifikasi/*labelling*, pengelompokkan dan pembukuan/administrasi sesuai dengan tujuan manajemen aset.

2. Legal audit.

Demikian menyangkut legal audit sebagai lingkup kerja manajemen aset yang berupa inventarisasi status penguasaan aset, sistem dan prosedur penguasaan atau pengalihan aset. Selanjutnya identifikasi dan mencari solusi atas

permasalahan legal, dan strategi untuk memecahkan berbagai permasalahan legal yang terkait dengan penguasaan dan pengalihan aset. Masalah yang sering dihadapi dalam legal audit, menyangkut status penguasaan yang lemah, aset dikuasai pihak lain, pemindahan aset yang tidak termonitor dan lain lain.

3. Penilaian aset.

Kesatuan kerja lanjutan dari manajemen aset, yaitu berupa kegiatan penilaian aset sebagai upaya penilaian atas aset yang dikuasai pemerintah daerah dan biasanya kegiatan ini dilakukan oleh konsultan penilaian independent. Hasil dari nilai tersebut akan dapat dimanfaatkan untuk mengetahui nilai kekayaan maupun informasi untuk penetapan harga bagi aset yang ingin dijual.

4. Optimalisasi aset.

Selanjutnya optimalisasi aset merupakan kegiatan untuk mengoptimalkan potensi fisik, lokasi, nilai, jumlah/volume, legal dan ekonomi yang dimiliki aset tersebut. Dalam kegiatan ini aset-aset yang dikuasai Pemda diidentifikasi dan dikelompokkan atas aset yang memiliki potensi dan yang tidak memiliki potensi. Aset yang memiliki potensi dapat dikelompokkan berdasarkan sektor-sektor unggulan yang dapat menjadi tumpuan dalam strategi pengembangan ekonomi nasional, baik dalam jangka pendek, menengah, maupun jangka panjang. Untuk menentukan hal tersebut harus terukur dan transparan, sedangkan aset yang tidak dapat dioptimalkan, harus dicari faktor penyebabnya, apakah faktor permasalahan legal, fisik, nilai ekonomi yang rendah ataupun faktor lainnya, sehingga setiap aset nantinya memberikan nilai tersendiri. Hasil akhir dari tahapan ini adalah rekomendasi yang berupa sasaran, strategi dan program untuk mengoptimalkan aset yang dikuasai.

1.8 Strategi Manajemen Aset

Strategi Manajemen Aset berguna untuk mengidentifikasi pendekatan yang paling memadai untuk memenuhi kebutuhan pemberian pelayanan.

1. Rencana pengadaan, yang menjelaskan aset- aset yang dibutuhkan atau diganti dalam periode perencanaan dan yang menyusun sumber dan biaya pendanaan untuk pengadaan.

2. Rencana operasional menjelaskan kebijakan penggunaan aset yang telah ada dan mungkin mencakup hal-hal seperti jam operasi, pemakaian, keamanan, manajemen energi dan pembersihan.
3. Rencana pemeliharaan menyusun standar atas aset-aset yang akan dipelihara, bagaimana standar akan dicapai, dan bagaimana pelayanan pemeliharaan akan diberikan.
4. Rencana penghapusan akan menjelaskan seluruh aset yang akan dihapuskan dalam periode perencanaan, metode penghapusan yang dipilih dan hasil yang diharapkan dari penghapusan.

2.8.1 Ruang Lingkup Barang Milik Negara

Barang milik negara dapat meliputi dari hal-hal yang telah di tentukan oleh Sekretariat Jendral Kementerian Agama Republik Indonesia, diantaranya adalah sebagai berikut :

1. Barang yang dibeli atau diperoleh atas beban APBN
2. Barang yang berasal dari perolehan lainnya yang sah

Maksud dari barang yang diperoleh dari yanglainnya yang sah adalah sebagai berikut :

- a. Hibah, sumbangan atau yang sejenis
- b. Pelaksanaan perjanjian atau kontrak
- c. Berdasarkan ketentuan undang-undang
- d. Berdasarkan keputusan pengadilan yang telah memperoleh kekuatan hukum tetap .

2.8.2 Ruang Lingkup Pengelolaan Barang Milik Negara

Berdasarkan pada peraturanPP No.6/2006, pengaturan pengelolaan Barang Milik Negara meliputi hal-hal sebagai berikut :

1. Perancangan dan penganggaran
2. Pengadaan
3. Penggunaan
4. Pemanfaatan

5. Pemeliharaan
6. Penilaian
7. Penghapusan
8. Pemindahtanganan
9. Penatausahaan
10. Pengawasan atau pengendalian

1.9 SDLC (System Development Life Cycle)

SDLC (*System Development Life Cycle*) merupakan siklus yang menggambarkan perangkat lunak yang dibangun. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap:

1. Rencana (*planning*),
2. Analisis (*analysis*),
3. Desain (*design*),
4. Implementasi (*implementation*),
5. Uji coba (*testing*) dan
6. Pengelolaan (*maintenance*).

Dalam rekayasa perangkat lunak, konsep SDLC mendasari berbagai jenis [metodologi pengembangan perangkat lunak](#). Metodologi-metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem informasi, yaitu [proses pengembangan perangkat lunak](#). Terdapat 3 jenis metode siklus hidup sistem yang paling banyak digunakan, yakni:

1. [Siklus hidup sistem tradisional](#) (*traditional system life cycle*), Berupa definisi masalah, Studi kelayakan, Analisa, Desain, dan Implementasi.
2. [Siklus hidup menggunakan protoyping](#) (*life cycle using prototyping*), Merupakan salah satu metode [siklus hidup sistem](#) yang didasarkan pada konsep model bekerja (*working model*)
3. [Siklus hidup sistem orientasi objek](#) (*object-oriented system life cycle*), Dasar dari konsep obyek oriented adalah obyek, class, asosiasi dan

inheritance menjadi kemudi dalam proses pembuatan dari tahap paling awal dan menyediakan dasar tunggal bagi daur hidup sistem.

2.10 Pendekatan Berorientasi Objek

Perancangan berorientasi objek adalah suatu teknik atau cara pendekatan baru dalam melihat permasalahan dan sistem (sistem perangkat lunak, Sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Pengertian “berorientasi objek” berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya.

2.10.1 Karakteristik dari Objek

Objek ialah Identitas. Berarti bahwa data diukur mempunyai nilai tertentu yang membedakan entitas disebut Objek. Objek dapat kongkrit, seperti halnya arsip dalam sistem, atau konseptual seperti kebijakan penjadwalan dalam multiprocessing pada sistem operasi. Setiap objek mempunyai sifat yang melekat pada identitasnya. Dua objek dapat berbeda walaupun bila semua nilai atributnya identik.

Kelas Objek ialah gambaran sekumpulan Objek yang terbagi dalam atribut, operasi, metode, hubungan, dan makna yang sama. Suatu kegiatan mengumpulkan data (atribut) dan perilaku (operasi) yang mempunyai struktur data sama ke dalam satu grup. Kelas Objek merupakan wadah bagi Objek. Dapat digunakan untuk menciptakan Objek. Objek mewakili fakta / keterangan dari sebuah kelas.

2.10.2 Metodologi Berorientasi Objek

Metodologi pengembangan sistem berorientasi objek mempunyai 3 karakteristik utama :

1. Encapsulation

Encapsulation merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi

dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri.

2. *Inheritance*

Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek dari objek induk diturunkan kepada anak objek, demikian seterusnya. *Inheritance* mempunyai arti bahwa atribut dan operasi yang dimiliki bersama di antara kelas yang mempunyai hubungan secara hirarki. Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya. Kelas Objek dapat didefinisikan atribut dan service dari kelas Objek lainnya. *Inheritance* menggambarkan generalisasi sebuah kelas.

3. *Polymorphism*

Polimorfisme yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda. Kemampuan objek-objek yang berbeda untuk melakukan metode yang pantas dalam merespon message yang sama. Seleksi dari metode yang sesuai bergantung pada kelas yang seharusnya menciptakan Objek.

2.11 *Object Oriented Analysis & Design (OOAD)*

OOAD adalah metode analisis yang memeriksa requirements dari sudut pandang kelas kelas dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek system atau subsistem. OOAD merupakan cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar

dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas.

2.11.1 Konsep Dasar OOAD (*Object Oriented Analysis & Design*)

OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu Analisis Berorientasi Objek (OOA) dan Desain Berorientasi Objek (OOD). Analisis Berorientasi Objek adalah metode analisis yang memeriksa *requirement* (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. Sedangkan Desain Berorientasi Objek adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem. Tools Perancangan Berorientasi Objek

Adapun *tools-tools* yang digunakan dalam perancangan berorientasi objek sebagai berikut :

1. *Object Oriented Analysis* (OOA) dan *Object Oriented Design* (OOD) dari Peter Coad dan Edward Yourdon [1990].
2. UML (*Unified Modeling Language*) dari James Rumbaugh, Grady Booch dan Ivar Jacobson [1997].

2.11.2 Kelebihan dan Kekurangan Perancangan Berorientasi Objek

Adapun kelebihan dan kekurangan perancangan berorientasi objek sebagai berikut :

1. Kelebihan Perancangan Berorientasi Objek :
 - a. Dibandingkan dengan metode SSAD, OOAD lebih mudah digunakan dalam pembangunan sistem.
 - b. Dibandingkan dengan SSAD, waktu pengembangan, level organisasi, ketangguhan, dan penggunaan kembali (*reuse*) kode program lebih tinggi dibandingkan dengan metode OOAD (Sommerville, 2000).

- c. Tidak ada pemisahan antara fase desain dan analisis, sehingga meningkatkan komunikasi antara user dan developer dari awal hingga akhir pembangunan sistem.
 - d. Analis dan programmer tidak dibatasi dengan batasan implementasi sistem, jadi desain dapat diformulasikan yang dapat dikonfirmasi dengan berbagai lingkungan eksekusi.
 - e. Relasi obyek dengan entitas (thing) umumnya dapat di mapping dengan baik seperti kondisi pada dunia nyata dan keterkaitan dalam sistem. Hal ini memudahkan dalam memahami desain (Sommerville, 2000).
 - f. Memungkinkan adanya perubahan dan kepercayaan diri yang tinggi terhadap kebenaran software yang membantu untuk mengurangi resiko pada pembangunan sistem yang kompleks (Booch, 2007).
 - g. *Encapsulation* data dan method, memungkinkan penggunaan kembali pada proyek lain, hal ini akan memperjelas proses desain, pemrograman dan reduksi harga.
 - h. OOAD memungkinkan adanya standarisasi obyek yang akan memudahkan memahami desain dan mengurangi resiko pelaksanaan proyek.
 - i. Dekomposisi obyek, memungkinkan seorang analis untuk memecah masalah menjadi pecahan-pecahan masalah dan bagian-bagian yang dimanage secara terpisah. Kode program dapat dikerjakan bersama-sama. Metode ini memungkinkan pembangunan software dengan cepat, sehingga dapat segera masuk ke pasaran dan kompetitif. Sistem yang dihasilkan sangat fleksibel dan mudah dalam memelihara.
2. Kekurangan Perancangan Berorientasi Objek
- a. Pada awal desain OOAD, sistem mungkin akan sangat simple.
 - b. Pada OOAD lebih fokus pada coding dibandingkan dengan SSAD.
 - c. Pada OOAD tidak menekankan pada kinerja team seperti pada SSAD.
 - d. Pada OOAD tidak mudah untuk mendefinisikan class dan obyek yang dibutuhkan sistem.

- e. Sering kali pemrograman berorientasi obyek digunakan untuk melakukan analisis terhadap fungsional sistem, sementara metode OOAD tidak berbasis pada fungsional sistem.
- f. OOAD merupakan jenis manajemen proyek yang tergolong baru, yang berbeda dengan metode analisis dengan metode terstruktur. Konsekuensinya adalah, team developer butuh waktu yang lebih lama untuk berpindah ke OOAD, karena mereka sudah menggunakan SSAD dalam waktu yang lama (Hantos, 2005).
- g. Metodologi pengembangan sistem dengan OOAD menggunakan konsep reuse. Reuse merupakan salah satu keuntungan utama yang menjadi alasan digunakannya OOAD. Namun demikian, tanpa prosedur yang eksplisit terhadap reuse, akan sangat sulit untuk menerapkan konsep ini pada skala besar (Hantos, 2005).

2.12 Unified Modeling Language (UML)

UML (Unified Modeling Language) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan system non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.12.1 Diagram-Diagram UML

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu. Adapun jenis diagram antara lain :

1. Use Case Diagram

Use case adalah abstraksi dari interaksi antara system dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata user. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

2. *Class Diagram*

Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa class diagram. Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

3. *Component Diagram*

Component software merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada di benak para analis. Komponen merupakan implementasi software dari sebuah atau lebih class. Komponen dapat berupa source code, komponen biner, atau *executable component*. Sebuah komponen berisi informasi tentang logic class atau class yang diimplementasikan sehingga membuat pemetaan dari logical view ke component view. Sehingga component diagram merepresentasikan dunia riil yaitu component software yang mengandung *component, interface dan relationship*.

4. *Deployment Diagram*

Menggambarkan tata letak sebuah system secara fisik, menampilkan bagian-bagian software yang berjalan pada bagian-bagian hardware, menunjukkan hubungan komputer dengan perangkat (nodes) satu sama lain dan jenis hubungannya. Di dalam nodes, *executeablecomponent* dan object yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

5. *State Diagram*

Menggambarkan semua state (kondisi) yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. *State class* tidak digambarkan untuk semua class, hanya

yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.

6. ***Sequence Diagram***

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

7. ***Collaboration Diagram***

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan object dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

8. ***Activity Diagram***

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

2.13 Pengertian Antarmuka (*Interface*)

Pengertian antarmuka (*interface*) adalah salah satu layanan yang disediakan sistem operasi sebagai sarana interaksi antara pengguna dengan sistem operasi. Antarmuka adalah komponen sistem operasi yang bersentuhan langsung dengan pengguna. Terdapat dua jenis antarmuka, yaitu *Command Line Interface*(CLI) dan *Graphical User Interface*(GUI).

2.14 Konsep Pengembangan Website

Ada beberapa hal yang perlu diketahui ketika pertama kali ingin membuat website, hal ini bertujuan agar website yang dibuat nampak profesional. Berikut beberapa aspek yang perlu diperhatikan dalam membuat website :

1. *Usability* : Umumnya pengguna ingin mendapatkan informasi secara tepat dan cepat meskipun tampilan website-nya biasa saja. Karena meskipun memiliki website yang dinamis yang nampak profesional dan bagus, namun jika perlu waktu yang lama untuk mendownload artikel atau penggunaan navigasinya rumit, jangan heran kalau pengguna akan “hengkang” dari website kita.
2. *Navigation*: Navigasi berfungsi untuk membantu penggunaan dalam menjelajah website kita untuk mencari informasi yang diinginkan. Navigasi yang bagus mencerminkan struktur website yang sangat baik.
3. *Writing Concept* : Bagi pengguna internet membaca teks yang banyak sekali merupakan hal yang akan membuang waktu. Riset membuktikan bahwa umumnya pengguna akan lebih cepat membaca di layar monitor dibandingkan membaca diatas kertas. Oleh karena itu, buatlah content dengan kalimat sesederhana mungkin.
4. *Simplicity* : Kesederhanaan disini maksudnya bukan dalam arti tampilan sebuah situs, namun lebih mengarah ke teknik penulisannya.
5. *Accessibility* : Ketika membuat website, pastikan website mudah digunakan oleh browser utama yang sering digunakan banyak orang, ujilah website untuk memeriksa kinerja pada berbagai sistem operasi dengan versi browser yang berbeda.
6. *Color Blindness* : Sekitar satu dari belasan pengunjung mungkin tidak mampu menggunakan website kita berkaitan dengan format kebutaan warna. Oleh karena itu selalu pertimbangkan pewarnaan pada website, salah satunya dengan mempertahankan tingkat kontras antara teks dan latar belakang.

2.15 Pemrograman Web

Sebuah situs web adalah sebutan bagi sekelompok halaman web (*web page*), yang umumnya merupakan bagian dari suatu nama domain (*domain name*) atau subdomain di *World Wide Web* (WWW) di Internet. WWW terdiri dari seluruh situs web yang tersedia untuk publik.

Tidak semua situs web dapat dibuka/akses dengan gratis. Beberapa situs web memerlukan pembayaran agar dapat menjadi pelanggan, misalnya situs-situs berita, layanan surat elektronik, dan lain-lain. Webside dibagi menjadi 2 macam yaitu :

1. **Website statis** pada umumnya menggunakan script html yang mana untuk file yang dieksekusi dari browser menggunakan ekstensi akhiran ".html"
2. **Website dinamis** sering yang kita jumpai menggunakan script php, namun ada juga yang menggunakan jsp, asp, *cold fusion*, dan lain-lain.

2.15.1 PHP (*Personal Home Page*)

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: *Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Beberapa kelebihan PHP

- a. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
- c. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
- d. Dalam sisi pemahaman, 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 system.

2.15.2 Web Database

MySQL adalah sebuah perangkat lunak iasm 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.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. 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*, 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. Jenis Kolom. MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (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. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.