BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Menurut Jogiyanto Hartono (2004:683) mendefinisikan bahwa :

Suatu sistem dapat terdiri dari sistem – sistem, bagian (*subsistems*) . Masing – masing subsistem dapat terdiri dari komponen – komponen. Subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

2.1.1. Karakteristik Sistem

Menurut Jogiyanto Hartono (2004:684) menyatakan karakteristik dari sistem adalah, sebagai berikut :

1. Komponen Sistem

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

2. Batasan Sistem

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang

sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan dari sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. lingkungan luar yang mengutungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung merupakan media perantara antara satu subsistem dengan sistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Output dari satu subsistem akan menjadi input untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berinteraksi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa *maintenance* input dan sinyal input . *Maintenance* input adalah energi yang dimasukkan supaya sistem

tersebut dapat beroperasi. Sinyal input adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem

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

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atas sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan- bahan yang lain menjadi keluaran berupa barang jadi.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.1.2. Klasifikasi Sistem

Menurut Jogiyanto Hartono (2004:687) menyatakan Sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya adalah sebagai berikut:

1. Sistem abstrak dan sistem fisik.

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologi, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sedangkan, Sistem fisik adalah sistem yang ada secara fisik. Misalnya sistem komputer, sistem operasi, sistem penjualan, dan lain sebagainya.

2. Sistem alamiah dan sistem buatan manusia.

   Sistem alamiah adalah sistem yang terjadi karena proses alam tidak dibuat oleh manusia (ditentukan dan tunduk kepada kehendak sang pencipta alam). Misalnya sistem perputaran bumi, sistem pergantian siang dan malam, sistem kehidupan umat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sedangkan, Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut dengan *human- machinesistem* atau ada yang menyebut dengan *man-machine sistem*. Sistem informasi merupakan contoh *man-machine sistem*. Karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

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

   Sistem Tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian- bagiannya dapat dideteksi dengan pasti, sehingga keluaran dan sistem tersebut dapat diramalkan dan relatif stabil atau konstan dalam jangka waktu yang lama. Contoh: Sistem komputer.

   Sedangkan, Sistem Tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem tertutup (*closesistem*) dan sistem terbuka (*opensistem*).

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkunganluarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya walaupun sebenarnya bersifat *relatively closed* sistem (secara relatif tertutup, tidak benar-benar tertutup). Sedangkan, Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lain.

2.1.3. Elemen Sistem

Menurut Jogianto dalam bukunya suatu sistem mempunyai elemen-elemen penyusun diantaranya :

1. Tujuan

   Sasaran atau tujuan dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem, karena suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuan.

2. Batasan

   Dalam mencapai tujuan dari sistem dibutuhkan batasan-batasan suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Kontrol

   Merupakan pengawasan dari pelaksanaan pencapaian tujuan sistem yang dapat berupa asal masukan, frekuensi pemasukan data, jenis pemasukan data, dan lain-lain.

4. Input

   Merupakan bagian dari sistem yang bertugas untuk menerima data masukan dimana data dapat berupa asal masukan, frekuensi pemasukan data, jenis pemasukan data, dan lain-lain.

5. Proses

   Merupakan bagian yang mengolah data menjadi informasi sesuai dengan keinginan pemakai.

6. Output

   Merupakan keluaran atau tujuan dari sistem yang didapat dari input dan proses yang dilakukan, misalnya laporan.

7. Umpan Balik

   Dalam sebuah sistem yang baik dibutuhkan adanya umpan balik yang tujuannya sebagai perbaikan dan pemeliharaan.

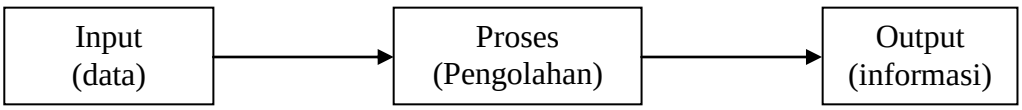
2.2. Konsep Dasar Informasi

Menurut Raymond Mcleod : " Informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang " . Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang sering terjadi adalah transaksi perubahan dari suatu nilai yang disebut transaksi. Kesatuan nyata adalah berupa suatu objek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol- simbol semacam huruf, angka, bentuk suara, sinyak, gambar, dsb. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sabagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus.

2.2.1. Siklus Informasi

Untuk memperoleh informasi yang bermanfaat bagi penerimanya, maka perlu dijelaskan bagaimana siklus yang terjadi atau dibutuhkan dalam menghasilkan informasi. Siklus informasi atau siklus pengolahan data adalah sebagai berikut. Al-Bahra (2005:11).



Gambar 2.1 Siklus Informasi

[Sumber : Al-bahra bin Ladjamudin. 2005. *Analisis dan Desain Sistem Informasi.* Graha Ilmu. Yogyakarta.]

2.2.2. Kualitas Informasi

Menurut Jogiyanto Hartono (2004:696) menyatakan Kualitas informasi tergantung dari 3 hal, yaitu informasi harus akurat, tepat waktu dan relevan :

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

2.2.3. Nilai Informasi

Menurut Jogiyanto Hartono (2004:696) Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Kegunaan informasi adalah untuk mengurangi hal ketidakpastian di dalam proses pengambilan keputusan tentang sesuatu keadaan.

2.3. Konsep Dasar Sistem Informasi

Definisi dari Sistem Informasi sendiri ialah Suatu Sistem terintegrasi yang mampu menyediakan Informasi yang bermanfaat bagi penggunanya atau

Sebuah Sistem terintegrasi atau Sistem manusia-mesin, untuk menyediakan Informasi untuk mendukung operasi, manajemen dalam Sistemini memanfaatkan perangkat keras dan perangkat suatu organisasi. lunak komputer, prosedur manual,model manajemen dan basis data.

Sedangkan Menurut Robert A. Leitch ; Sistem Informasi adalah suatu Sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.3.1. Komponen Sistem Informasi

Sistem Informasi dapat terdiri dari komponen – komponen yang disebut dengan istilah blok bangunan (*building block*) . Sebagai suatu sistem, keenam blok tersebut masing – masing asling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

Berikut penjelasan dari masing – masing blok bangunan, yaitu :

1. Blok Input

   Input mewakili data yang masuk kedalam sistem informasi, termasuk metode dan media untuk memperoleh data yang dimasukan, yang dapat berupa dokumen dasar.

2. Blok Model

   Kombinasi prosedur, logika, dan model matemetik yang akan memanipulasi data input dan data yang tersimpan di

basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diingiinkan.

3. Blok Output

   Produk dari sistem informasi adalah keluaran berupa informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem

4. Blok Teknologi

   Teknologi merupakan kotak alat (*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 menyeluruh.

5. Blok Basis Data

   Merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan diperangkat keras komputer dan perangkat lunak digunakan untuk memanipulasinya. Data didalam basis data perlu diorganisasikan sedemikian rupa supaya informsi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data dapat diakses atau dapat dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*database management sistem*).

6. Blok Kendali

   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.

2.3.2. Perancangan Sistem Informasi

Menurut Jogiyanto (2005 : 23), perancangan sistem adalah tahap yang dilakukan setelah melakukan analisis sistem, pendefinisian kebutuhan–kebutuhan sistem yang akan dibangun, dan persiapan untuk merancang bangun implementasi sistem dengan menggambarkan sistem yang akan dibangun. Perancangan sistem dimulai dengan memahami sistem yang sedang berjalan dan kriteria–kriteria sistem yang akan dibangun biasanya menggunakan pemodelan secara terstruktur yang digambarkan oleh grafik atau diagram. Hal–hal yang harus diperhatikan dalam merancang dan membangun sebuah sistem menurut Jogiyanto (2005 : 25) adalah :

1. Kebutuhan perusahaan, organisasi, atau lembaga, yaitu dengan cara memahami bidang yang akan dikembangkan,sasaran yang dibidik serta media yang akan digunakan.
2. Kebutuhan operator, yaitu kebutuhan operator untuk memperoleh sistem yang mudah dipahami dan dioperasikan serta tampilan yang interaktif. Kebutuhan operator dapat diperoleh melalui proses wawancara atau kuisioner.

3. Kebutuhan pemakai, yaitu keinginan – keinginan dari pemakai sistem informasi seperti jaminan, keamanan, standarisasi tampilan, kecepatan akses, dan kemudahan dalam pengoperasian.
4. Kebutuhan teknis, yaitu meliputi arsitektur dan konfigurasi sistem. Secara teknis peralatan dan teknologi yang digunakan, termasuk pertimbangan penggunaan peralatan yang tidak standar, seperti peralatan multimedia, kebutuhan interface, database, dan perangkat lunaknya.

2.3.3. Pengembangan Sistem Informasi

Menurut Jogiyanto (2005 : 35) Menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem yang lama perlu diperbaiki atau diganti. Menurut Jogiyanto (2005 : 36) Pengembangan sistem dapat berarti penyusunan suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada.

1. Adanya permasalahan-permasalahan yang timbul pada sistem yang lama. Permasalahan yang timbul dapat berupa :
    1. Ketidakberesan

        Ketidakberesan pada sistem yang lama menyebabkan sistem yang lama tidak dapat beroperasi sesuai dengan yang diharapkan.

2. Pertumbuhan Organisasi

Pertumbuhan organisasi yang menyebabkan harus disusunnya sistem yang baru. Pertumbuhan organisasi diantaranya adalah kebutuhan informasi yang semakin luas. *Volume* pengolahan data semakin meningkat, perubahan prinsip akuntasi yang baru. Karena adanya perubahan ini, maka menyebabkan sistem yang lama tidak efektif lagi sehingga sistem yang lama sudah tidak dapat memenuhi lagi semua kebutuhan informasi yang dibutuhkan oleh manajemen.

2. Untuk meraih kesempatan-kesempatan

Teknologi informasi telah berkembang dengan cepatnya. Perangkat keras komputer, perangkat lunak dan teknologi komunikasi telah begitu cepat berkembang. Organisasi mulai merasakan bahwa teknologi informasi ini perlu digunakan untuk meningkatkan penyediaan informasi sehingga dapat mendukung dalam proses pengambilan keputusan yang akan dilakukan oleh manajemen. Dalam keadaan pasar bersaing, kecepatan informasi atau efisiensi sangat menentukan berhasil atau tidaknya strategi dan rencana-rencana yang telah disusun untuk meraih kesempatan- kesempatan yang asa. Bila pesaing dapat memanfaatkan teknologi ini, maka kesempatan-kesempatan akan jatuh ke tangan pesaing. Kesempatan-kesempatan ini dapat

berupa peluang-peluang pasar, pelayanan yang meningkat kepada pelanggan dan lain sebagainya.

3. Adanya instruksi-instruksi

Penyusunan sistem yang baru dapat juga terjadi karena adanya instruksi-instrukai dari atas pimpinan atau dari luar organisasi, seperti peraturan pemerintah dan sebagainya. Dengan dikembangkannya sistem yang baru, maka diharapkan akan terjadi peningkatan-peningkatan ini, yaitu sebagai berikut :

1. *Performance* (kinerja)

   Peningkatan terhadap kinerja (hasil kerja) sistem yang baru sehingga menjadi lebih efektif. Kinerja dapat diukur dari :

   1. *Throughput*, yaitu jumlah dari pekerjaan yang dapat dilakukan suatu saat tertentu.
   2. *Response time*, yaitu rata-rata waktu yang tertunda diantara dua pekerjaan tersebut.

2. *Economy* (ekonomis)

   Peningkatan terhadap manfaat-manfaat atau keuntungan-keuntungan atau penurunan-penurunan biaya yang terjadi

3. *Control* (Pengendalian)

   Peningkatan terhadap pengendalian untuk mendeteksi dan memperbaiki kesalahan-kesalahan serta keuntungan-keuntungan yang akan terjadi.

4. *Effeciency* (efisiensi)

   Peningkatan terhadap efesiensi operasi,yaitu bagai mana sumber daya digunakan dengan pemborosan yang paling minimum.

5. *Service* (pelayanan)

   Peningkatan terhadap pelayanan yang diberikan oleh sistem.

2.4. Metodologi Berorientasi Objek

Metodologi ini diperkenalksan sekitar tahun 1990 sebagai pelengkap untuk pemrograman yang terlebih dahulu telah mengadopsi metode berorientasi objek. Beberapa alat dan teknik yang dapat digunakan antara lain *dynamic* dan *static object-oriented model*, *state transisition diagram* dan *case scenario*.

Fokus utama metodologi ini pada objek, dengan melihat suatu sistem terdiri dari objek yang saling berhubungan dengan beberapa cara untuk mencapai suatu tujuan. Objek dapat digambarkan sebagai benda, orang, tempat dan sebaginya yang mempunyai atribut dan *method*. Objek dalam *software analysis & design* adalah sesuatu berupa konsep (*concept*), benda (*thing*), dan sesuatu yang membedakannya dengan lingkungannya. Secara sederhana objek adalah mobil, manusia, alarm dan lain-lainnya. Tapi objek dapat pula merupakan sesuatu yang abstrak yang hidup didalam sistem seperti tabel, *database, event, sistem messages*. Objek dikenali dari keadaannya dan juga operasinya. Sebagai contoh sebuah mobil dikenali dari warnanya, bentuknya, sedangkan manusia dari suaranya. Ciri-ciri ini yang akan membedakan objek tersebut dari objek lainnya.

Object Oriented Programming merupakan cara atau metode baru dalam membuat program. Langkah membuat program dengan teknik OOP, pertama kali mengetahui objeknya, kemudian atribut (*field/data*) dan perilakunya (fungsi). Berbeda dengan pemrograman prosedural (konvensional), membuat program memakai aspek *input*, proses, dan *output*. OOP adalah cara berpikir, pandangan, atau paradigma baru untuk membuat program/merancang sistem dengan memperhatikan objek, cirri objek, dan perilakunya. OOP ini sangat berbeda dengan program prosedural yang fokusnya aspek input, proses, dan output.

2.5. Pengertian *Object Oriented Analysis and Design*

Menurut Mathiassen ”*Object-Oriented Analysis and Design* (OOAD) adalah metode untuk menganalisa dan merancang sistem dengan pendekatan berorientasi object”. *Object* diartikan sebagai suatu entitas yang memiliki identitas, *state*, dan *behavior*.

Pada analisa, identitas sebuah *object* menjelaskan bagaimana seorang *user* membedakannya dari *object* lain, dan *behavior object* digambarkan melalui event yang dilakukannya. Sedangkan pada perancangan, identitas sebuah *object* digambarkan dengan cara bagaimana *object* lain mengenalinya sehingga dapat diakses, dan *behavior object* digambarkan dengan *operation* yang dapat dilakukan *object* tersebut yang dapat mempengaruhi *object* lain dalam sistem.

2.5.1. Prinsip Umum OOAD

1. *Model the context* : Sistem yang bermanfaat sesuai dengan konteks OOAD. *Emphasize the architecture* : Merupakan arsitektur yang mudah dipahami yang memfasilitasi kolaborasi antara *designer* dan *programmer*.
2. Reuse Patters : Dibangun berdasarkan gagasan-gagasan yang kuat dan komponen *pretested* memperbaiki kualitas sistem dan produktivitas dari proses *development*.
3. *Tailor the method to suit specific projects* : Setiap usaha *devlopment* masing-masing mempunyai tantangan yang unik. OOA&D harus disesuaikan dengan kebutuhankebutuhan yang khusus dari situasi analisis dan desain yang diberikan.

2.5.2. Keuntungan OOAD

1. Menurut konsep umum yang dapat digunakan untuk memodelkan hampir semua fenomena dan dapat dinyatakan dalam bahasa umum (*natural language*).
2. Memberikan informasi yang jelas tentang *context* dari sistem.
3. Mengurangi biaya *maintainance*.
4. Memudahkan untuk mencari hal yang akan diubah.
5. Membuat perubahan menjadi lokal tidak berpengaruh pada modul yang lain.

2.5.3. Karakteristik OOAD

Dalam pendekatan berorientasi objek ada 4 pilar utama yang harus dipahamai dalam pendekatan berorientasi objek yaitu karakteristik. Karakteristik (ciri) suatu program termasuk OOAD/OOP, apabila terdapat abstraksi, pembungkusan (*encapsulation*), *polymorphisme*, dan turunan (*inheritance*).

1. *Abstraction*

   Kemampuan untuk menjadikan dalam bentuk yang lebih sederhana. Hal ini juga dikenal dalam metodologi pendekatan struktur yaitu dekomposisi seperti menyerderhanakan suatu sistem dalam bentuk *Context* Diagram.

2. *Encapsulation*

   Merupakan suatu karakteristik OOAD dimana program terbungkus (jadi satu) data dan perilaku, artinya lebih memperhatikan aspek internal daripada aspek eksternal. Contoh: dalam program terdapat tombol *button close* didalamnya ada *method sistem.exit*(0) untuk keluar dari sistem java. Berbeda dengan metodologi terdahulu, metodologi ini menggabungkan atribut dan fungsi / proses kedalam suatu objek yang disebut dengan *encapsulation*. Setiap objek dapat "menyembunyikan" kompleksitasnya dan berhubungan dengan objek lain dengan mengirim "pesan / *message*" yang dapat dikenal dan diproses oleh objek penerima contoh Pada dunia nyata, seorang ibu rumah tangga menanak nasi dengan menggunakan *rice cooker*, ibu

tersebut menggunakannya hanya dengan menekan tombol. Tanpa harus tahu bagaimana proses itu sebenarnya terjadi. Disini terdapat penyembunyian informasi milik *rice cooker*, sehingga tidak perlu diketahui seorang ibu. Dengan demikian menanak nasi oleh si ibu menjadi sesuatu yang menjadi dasar bagi konsep *information hiding*.

3. *Polymorphism*e

Dengan kata lain suatu mekanisme yang memungkinkan suatu objek memiliki semua atau sebagian definisi dari objek induk. Menurut Bambang Hariyanto (2007:67) *Polymorphism* berasal dari kata Poly yang artinya banyak dan morph yang artinya bentuk. Jadi *polymorphism* adalah kemampuan suatu atribut atau *method* dapat berubah dalam berbagai bentuk dalam implementasi. Contoh Pada objek mobil, walaupun minibus dan truk merupakan jenis objek mobil yang sama, namun memiliki juga perbedaan. Misalnya suara truk lebih keras dari pada minibus, hal ini juga berlaku pada objek anak (*child*) melakukan metoda yang sama dengan algoritma berbeda dari objek induknya. Hal ini yang disebut *polymorphism*, teknik atau konsep dasar lainnya adalah ruang lingkup/pembatasan. Artinya setiap objek mempunyai ruang lingkup kelas, atribut, dan metode yang dibatasi.

4. *Inheritance*

Merupakan suatu karakteristik OOAD di mana suatu kelas (*parent/base class*) dapat diturunkan ke kelas lain (*child/derived class*), sehingga kelas anak dapat memiliki data atau perilaku kelas orangtuanya. Contoh dengan beberapa buah mobil yang mempunyai kegunaan yang berbeda-beda. Ada mobil bak terbuka seperti truk, bak tertutup seperti sedan dan minibus. Walaupun demikian objek-objek ini memiliki kesamaan yaitu teridentifikasi sebagai objek mobil, objek ini dikatakan objek induk (*parent)*. Sedangkan minibus objek anak (*child*), berarti semua operasi yang berlaku pada mobil berlaku pada minibus.

2.5.4. Ciri-ciri *Object Oriented Programming* (OOP)

Menurut Bambang Hariyanto dalam bukunya yang berjudul Rekayasa Perangkat Lunak Berorientasi Objek, bahwa *Object Oriented* memiliki ciri-ciri sebagai berikut: Bambang Hariyanto (2007:67)

1. Objek

   Bentuk baik yang nyata atau tidak, seperti manusia, hewan, benda, konsep, aliran, dan lain-lain. Objek merupakan inisiasi (turunan langsung) dari suatu kelas.

2. Kelas

   Kumpulan objek yang memiliki kemiripan perilaku (*method*), cirri atau karakteristik (*property*). Contoh objek orang

dari kelas manusia, potongan sebagai berikut: Manusia orang1=*new* manusia(“Parto”);

3. Metode

   Perilaku dari objek atau kelas tertentu. Merupakan perwujudan aksi atau tindakan dari dunia nyata di dalam pemrograman komputer.

4. Konstruktor

   Suatu fungsi yang dideklarasikan atau didefinisikan di dalam kelas, konstruktor harus mempunyai nama yang sama dengan fungsinya. Konstruktor dijalankan bersamaan dengan terciptanya kelas tersebut.

5. De-konstruktor

   Fungsi yang dideklarasikan dalam kelas, nama sama dengan nama fungsinya. Tetapi dijalankan bersamaan dengan dimusnahkannya kelas tersebut.

6. Karakteristik / properties

   Ciri yang dimiliki oleh suatu objek, karakteristik ini juga sebagai pembeda objek satu dengan objek lainnya dalam kelas yang sama (konsep individu).

7. Variabel

   Tempat menampung data sementara, dalam pemrograman objek biasanya disebut data, sedangkan dalam pemrograman prosedural sering disebut dengan variabel.

8. Data

   Istilah lain dari variabel dalam OOP. Dalam pemrograman java bisa juga disebut *field*, data *member* atau *instance variable*.

9. Hak akses (*access attribute*)

   Hak akses digunakan untuk dapat menentukan data *member* mana yang dapat digunakan oleh kelas lain, dan mana yang tidak dapat digunakan. Hak akses ini sangat penting dalam membuat program turunan kelas.

   1. *Publi*c

      Data *member* atau *variable* dapat diakses dari kelas mana saja

   2. *Protected*

      Dapat mengakses data member dari kelas dalam *package* yang sama dan subkelasnya

   3. *Privat*e

      Kelas yang data membernya memakai *private* hanya dapat digunakan oleh kelas bersangkutan, tidak dapat digunakan kelas lain

   4. Tidak disebutkan

      Data member dapat diakses dari kelas dalam *package* yang sama.

2.6. Bahasa Pemodelan UML (*Unified Model Language*)

Bahasa Pemodelan UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut.

2.6.1. Pengertian UML

UML adalah bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak tergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha dari banyak pihak, didukung oleh kakas-kakas yang integrasikan lewat XML (XMI). UML dikelolah oleh OMG (*Objek Management Group*).

2.6.2. Diagram-Diagram UML

UML mendefinisikan diagram-diagram berikut ini :

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.

2) *Activity diagram*

Activity diagram 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 *behavior internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

3) *Sequence diagram*

Sequence diagram digunakan untuk memberikan gambaran detail dari setiap *use case* diagram yang telah dibuat sebelumnya. Setiap *objects* yang terlibat dalam sebuah *use case* digambarkan dengan garis putus-putus vertical, kemudian message yang dikirim oleh *object* digambarkan dengan garis horizontal secara kronologis dari atas ke bawah.

4) *Collaboration diagram*

Collaboration diagram 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.

5) *Class diagram*

Class diagram menggambarkan *class* dan hubungan antar *class* di dalam sistem. *Class* diagram dibangun berdasarkan *use case diagram, sequence diagram* yang telah dibuat sebelumnya.

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

6) *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk keter gantungan (*dependency*) diantaranya.

7) *Deployment/physical Diagram*

Deployment atau *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 .

2.7. Pengertian Jaringan Komputer

Menurut Budhi Irawan (2005:19) menjelaskan bahwa jaringan komputer dapat dibedakan berdasarkan cakupan geografisnya. Adapun Tiga katagori utama jaringan komputer yaitu :

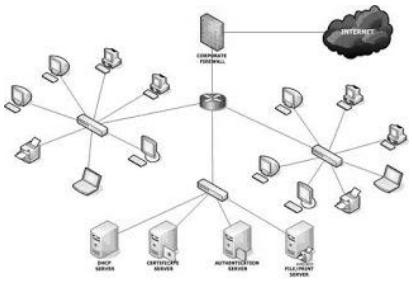
2.7.1. Tipe Jaringan Komputer

Jaringan dapat dibedakan berdasarkan cakupan ada empat kategori utama jaringan komputer, yaitu :

1. LAN (*Local Area Network*)

LAN digunakan untuk menghubungkan komputer yang berada didalam suatu area yang kecil, misalnya didalam suatu gedung perkantoran atau kampus. Jarak antar komputer yang dihubungkan bisa mencapai 5 sampai 10 km. Suatu LAN biasanya bekerja pada kecepatan mulai 10 mbps sampai 100 mbps. LAN menjadi popular karena memungkinkan banyak pengguna untuk memakai sumber daya secara bersama – sama. Contoh dari sumber daya yang

dapat digunakan misalnya suatu *main frame*, *file server*, *printer*, dan sebagainya.

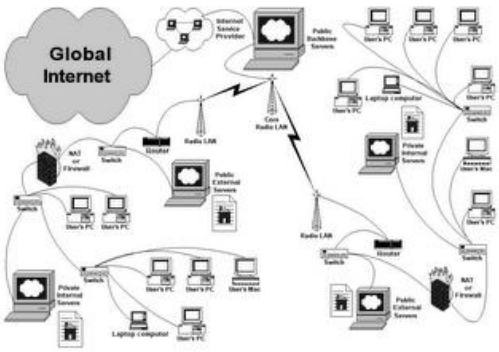


Gambar 2.2 Jaringan LAN (*Lokal Area Network*)

[Sumber :http://dudleysenanayakacc.blogspot.com/2010/lan]

2. MAN (*Metropolitan Area*)

MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN mneghubungkan LAN – LAN yang lokasinya berjauhan. Jangkauan MAN bisa mencapai 10 km samapai beberapa rausan km. Suatu MAN biasanya bekerja pada kecepatan 1,5 sampai 150 Mbps.

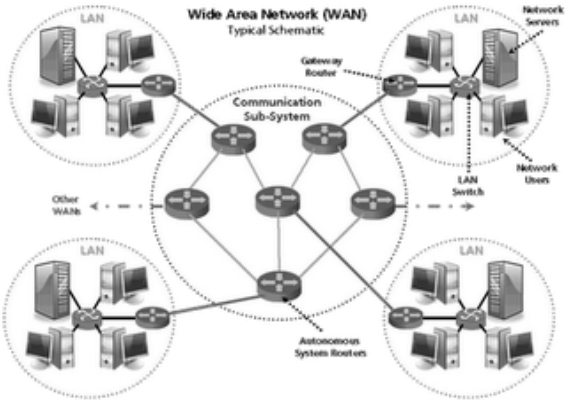


Gambar 2.3 Jaringan MAN (*Metropolitan Area*)

[Sumber :http://dudleysenanayakacc.blogspot.com/2010/man]

3. WAN (*Wide Area Network*)

WAN dirancang untuk menghubungkan komputer – komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari satu kota ke kota lain di dalam suatu negara. Cakupan WAN bisa meliputi 100 km sampai 1.000 km dan kecepatan antar kota bisa bervariasi antara 1,5 Mbps sampai 2,4 Gbps. Dalam WAN, biaya biasanya jaringan WAN dimiliki dan dioperasikan sebagai suatu jaringan publik.



Gambar 2.4 Jaringan WAN (*Wide Area Network*)

[Sumber :http://dudleysenanayakacc.blogspot.com/2010/wan]

4. GAN (*Global Area Network*)

GAN merupakan suatu jaringan yang menghubungkan negara – negara di seluruh dunia.Kecepatan GAN bervariasi mulai dari 1,5 sampai dengan 100Gbps dan cakupannya mencapai ribuan kilometer. Contoh yang sangat baik dari GAN ini adalah internet.



Gambar 2.5 Jaringan GAN (*Global Area Network*)

[Sumber :http://dudleysenanayakacc.blogspot.com/2010/gan]

2.7.2. Topologi Jaringan

Menurut Budhi Irawan (2005:25) topologi secara fisik dari suatu jaringan local adalah merujuk kepada konfigurasi kabel, computer, dan perangkat lainnya. Tipe-tipe utama topologi fisik Berikut ini akan di jelaskan mengenai topologi fisik yang di gunakan di dalam jarinagn local diantaranya :

1. *Linier Bus* (Garis Lurus)

   Topologi *linier Bus* (Garis Lusrus) terdiri dari jalur satu jalur kabel utama dimana pada masing-masing ujungnya diberikan sebuah terminator. Semua *nodes* pada jaringan (*file server*, *workstatio*n, dan perangkat lainnya) terkoneksi sebuah kabel utama *(backbone*). Jaringn-jaringan *ethernet* dan *local talk* menggunakan topologi *linier* ini.

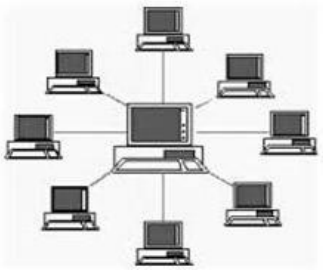


Gambar 2.6 Topologi Bus

[Sumber :http://prima.kurniawan.students-blog.undip.ac.id/2009/topologi-jaringan-komputer]

2. *Star* (Bintang)

   Topologi model ini dirancang, yang mana setiap nodes (*server*, *workstation*, dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator*.



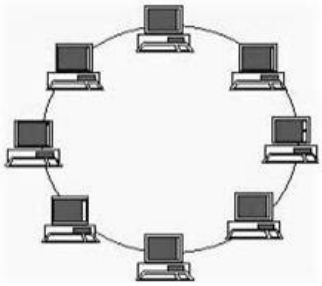
Gambar 2.7 Topologi Star

[Sumber :http://prima.kurniawan.students-blog.undip.ac.id/2009/topologi-jaringan-komputer]

3. *Ring* (Cincin)

   Topologi *Ring* (Cincin) menggunakan tehnik konfigurasi yang sama dengan topologi *star* tetapi pada topologi ini terlihat bahwa jalur media transmisis menyerupai suatu lingkungan

tertutup menyerupai cincin (lingkaran), sehingga diberi nama topologi bintang dalam lingkaran atau *star-wired ring*.

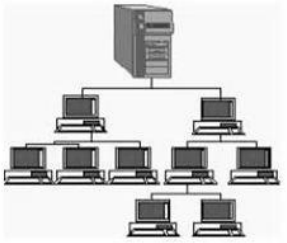


Gambar 2.8 Topologi Ring

[Sumber :http://prima.kurniawan.students-blog.undip.ac.id/2009/topologi-jaringan-komputer]

4. *Tree* (Pohon)

   Topologi model ini merupakan perpaduan antara topologi *Linear Bus* dan *Star*, yang mana terdiri dari kelompok – kelompok dari *workstation* dengan konfigurasi *star* yang terkoneksi ke kabel utama yang menggunakan topologi *Liniear Bus*. Topologi ini memungkinkan untuk pengembangan jaringan yang telah ada dan memungkinkan untuk menkonfigurasi jaringan sesuai dengan kebutuhan.



Gambar 2.9 Topologi Tree

[Sumber :http://prima.kurniawan.students-blog.undip.ac.id/2009/topologi-jaringan-komputer]

2.7.3. Pengertian *Client Server*

Menurut Budhi Irawan (2005:30) model hubungan *client server* memungkinkan jaringan untuk mesentralisasi fungsi dan aplikasi kepada satu atau dua *dedicated file server*. Sebuah *file sever* menjadi jantung dari keseluruhan sistem, memungkinkan untuk mengakses sumberdaya, dan menyediakan keamanan. *Workstation* yang berdiri sendiri dapat mengambil sumber daya yang ada pada *file server*. Model hubungan ini, menyediakan mekanisme untuk mengintegrasikan seluruh komponen yang ada di jaringan dan memungkinkan banyak pengunaan secara bersama- sama memakai sumber daya pada *file server*.

1. Kelebihan model hubungan *Client Server*.

   Terpusat (sumber daya dan keamanan data dikontrol melalui server).

   1. Skalabilitas
   2. Fleksibel

Teknologi baru dengan mudah terintregrasi kedalam sistem keseluruhan komponen (*client / network /server*) dapat bekerja bersama.

2. Kekuranagn model hubungan *Client server*.
    1. Mahal
    2. Membutuhkan investasi untuk *didecated file server*
    3. Perbaikan (jaringan besar membutuhkan seorang staff untuk mengatur agar sistem berjalan dengan efisien) Beketergantungan Ketika *server* jauh, mengakibatkan keseluruhan operasi pada *network* akan jatuh pula.

2.8. Perangkat Lunak Pendukung

Perangkat lunak pendukung adalah salah satu alat yang digunakan dari proses pengkodingan sampai implementasi, yaitu menggunakan Netbeans 7.1.1 dan MySQL.

2.8.1. Netbeans 7.1.1

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh dan memiliki hampir 100 mitra dan terus bertambah. *Sun Microsistems* mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Saat ini terdapat dua produk : *NetBeans* IDE dan *NetBeans Platform* JAVA.

Sejarah NetBeans Pengembangan *NetBeans* diawali dari *Xelifi*, sebuah proyek mahasiswa tahun 1997 di bawah bimbingan Fakultas matematika dan

Fisika Universitas Charles, Praha. Sebuah perusahaan kemudian dibentuk untuk proyek tersebut dan menghasilkan versi komersial *NetBeans* IDE hingga kemudian dibeli oleh *Sun Microsistem* pada tahun 1999. *Sun* kemudian menjadikan *NetBeans open source* pada bulan Juni tahun 2000. Sejak itu komunitas *NetBeans* terus berkembang. *NetBeans* mengacu pada dua hal, yakni *platform* untuk pengembangan aplikasi *desktop* java, dan sebuah *Integrated Development Kit* (IDE) yang dibangun menggunakan platform *NetBeans*.

Platfom *NetBeans* memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak moduler yang disebut 'modul'. Sebuah modul adalah suatu arsip Java (*Java archive*) yang memuat kelas-kelas Java untuk berinteraksi dengan *NetBeans Open API* dan *file manifestasi* yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis platform *NetBeans* dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan powerfull .*Netbeans* IDE

NetBeans IDE adalah IDE *open source* yang ditulis sepenuhnya dengan bahasa Java menggunakan platform *NetBeans*. *NetBeans* IDE mendukung pengambangan semua tipe aplikasi Java (J2SE, web, EJB, dan aplikasi mobile). Fitur lainnya adalah sistem proyek berbasis Ant, control versi, dan refactoring .

Modularitas adalah semua fungsi IDE disediakan oleh modul-modul. Tiap modul menyediakan fungsi yang didefinisikan dengan baik, seperti dukungan untuk bahasa pemrograman Java, editing, atau dukungan bagi CVS. *NetBeans* memuat semua modul yang diperlukan dalam pengembangan Java dalam sekali

download, memungkinkan pengguna untuk mulai bekerja sesegera mungkin. Modul-modul juga mengijinkan *NetBeans* untuk dikembangkan. Fitur-fitur baru, seperti dukungan bahsa pemrograman lain, dapat ditambahkan dengan menginstal modul tambahan, sebagai contoh Sun Studio, Sun Java Studio Enterprise, dan Sun Java Studio Creator dari Sun Microsistem semuanya berbasis NetBeans IDE.

2.8.2. MySQL

Menurut Abdul Kadir (2009:15) mendefinisikan bahwa MySQL merupakan software yang tergolong database server dan bersifat *Open Source*. *Open Source* menyatakan bahwa software ini di lengkapi oleh source *code* (kode yang di pakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode dapat di jalankan secara langsung di dalam sistem operasi dan bias diperoleh dengan cara mengunduh di internet secara gratis. Hal lainnya adalah MySQL juga bersifat multiplatform. MySQL dapat dijalankan pada berbagai sistem operasi.

2.9. JAVA

Definisi bahasa Java Menurut Sun Microsistem adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada *computer standalone* ataupun pada lingkungan jaringan.

Menurut definisi lainnya, Java adalah bahasa pemrograman yang *open source* dan berorientasi objek (OOP) serta bersifat *portable* sehingga bias dijalankan diberbagai sistem operasi.

2.9.1 Java Adalah Bahasa Pemrogramman

Java adalah bahasa pemrogramman yang bias digunakan untuk membuat aplikasi desktop, website, mobile dan lain-lain. Pada dasarnya semua aplikasi java bersifat *portable* yang artinya bias dijalankan dihampir semua sistem operasi, tapi dengan syarat di sistem operasi tersebut wajib terpasang *Java Runtime Environment* (JRE).

2.9.2 Edisi Bahasa Java

Bahasa Java selalu dikembangkan hingga secara garis besar bahasa Java memiliki 3 edisi, yaitu :

1. **J2SE (*Java 2 Standart Edition*)**

   Edisi ini adalah standar dari pemrograman Java. Edisi J2SE ini meliputi aplikasi desktop dan pada lingkungan *web browser* (*Applet*). Edisi ini mendukung pemrograman *database*, perancangan I/O (*input/output*), perancangan *interface, network programming* dan sebagainnya.

2. **J2ME (*Java 2 Micro Edition*)**

   Edisi ini adalah edisi Java yang diperuntukan untuk membuat aplikasi bersekala *micro*, seperti *handphone*, PDA, *pagers*, *smart card*, dsb. Banyak juga yang mendefinisikan bahwa edisi ini adalah edisi *mobile* dari bahasa pemrograman Java.

3. **J2EE (*Java 2 Enterprise Edition*)**

   Edisi ini adalah edisi Java yang paling komplit, karena diperuntukan untuk membangun aplikasi bersekala *enterprise*

(besar). J2EE berisikan paket-paket J2SE ditambah paket khusus *enterprise* seperti JavaBeans, Servlets,JSP dan XML.

2.10 Pengertian Pemesanan

Pemesanan adalah suatu aktifitas yang dilakukan oleh konsumen sebelum membeli. Untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem pemesanan yang baik.

Tujuan Pemesanan yaitu :

1. Memaksimalkan pelayanan bagi kosnsumen
2. Meminimumkan investasi pada persediaan
3. Perencanaan kapasitas
4. Pengesahan produksi dan pengendalian produksi

2.11 Pengertian Penjadwalan

Penjadwalan dapat didefinisikan sebagai pengaturan pengalokasian sumber daya dalam jangka waktu tertentu untuk melakukan serangkaian tugas yang telah dirancangkan.

Penjadwalan adalah aturan atau proses pengorganisasian pemilihan waktu, penggunaan tempat atau sumber - sumber untuk mengerjakan semua aktifitas yang diperlukan yang memenuhi kendala aktifitas dan sumber daya.

2.12 Pengertian Tour

Tour adalah perjalanan keliling yang memakan waktu lebih dari 3 hari yang diselenggarakan oleh suatu Biro Perjalanan Wisata dengan acara peninjauan di beberapa tempat yang menarik (Drs.Oka.Yeti : 26).

2.13 Pengertian Paket Wisata

Paket Wisata (*tour package*) yang berarti suatu rencana perjalanan wisata tersusun secara tetap dengan biaya yang sudah ditentukan di dalam paket wisata, biaya tersebut mencakup penginapan, transportasi, *sightseeing tour*, transfer yang semua tertera di dalam paket tersebut. Oleh karena itu paket wisata tersebut sangat tergantung terhadap perekonomian di suatu Negara dan harga kebutuhan yang berlaku pada waktu itu.

Oleh karena itu Paket Wisata (*Tour package*) ialah suatu program perjalanan wisata yang telah disusun dan diramu oleh penyelenggara secara tetap, dengan kondisi, harga, tempat tempat kunjungan, penginapan, transportasi, *sightseeing*, atraksi wisata dalam perjalanan telah dicantumkan dalam program. Biasanya harga *tour package* ini jauh lebih murah daripada disusun sendiri dan *package tour* ini biasanya mempunyai masa berlaku (*Limited Time*). Suatu produk wisata yang diciptakan oleh Biro Perjalanan Wisata yang telah tersusun dengan baik dengan cara menggabungkan beberapa unsur/ komponen jasa wisata beserta harga yang dilaksanakan secara tetap dan teratur disebut sebuah paket wisata.