

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Sistem**

Dari segi etimologi, kata sistem sebenarnya berasal dari Bahasa Yunani yaitu *sistema* dan dalam Bahasa Inggris dikenal dengan *system*, yang mempunyai satu pengertian yaitu sehimpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan satu keseluruhan yang tidak terpisahkan.

Pada dasarnya, sistem dapat didefinisikan sebagai sekumpulan elemen-elemen yang saling berhubungan dan saling bergantung untuk mencapai suatu tujuan.

##### **2.1.1 Konsep Dasar Sistem**

Jogiyanto (469) mendefinisikan, bahwa suatu sistem sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan.

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem.

Richard F. Neuschel (469) mendefinisikan prosedur (*procedure*) sebagai berikut:

Suatu prosedur adalah suatu urutan-urutan operasi klerikal (tulis menulis), biasanya melibatkan beberapa orang didalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.

### 2.1.2 Bentuk Umum Sistem

Bentuk umum dari sistem terdiri dari atas masukan (*input*), pengolahan (*process*), dan keluaran (*output*). Dalam bentuk umum sistem ini biasa melakukan satu atau lebih masukan yang akan diproses dan menghasilkan keluaran sesuai dengan yang direncanakan sebelumnya. Gambaran umum mengenai sistem ditunjukkan pada gambar berikut ini:



Gambar 2.1

Gambaran Umum Sistem

### 2.1.3 Elemen Sistem

Suatu sistem terdiri dari sejumlah komponen atau elemen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen sistem atau elemen sistem dapat berupa:

- a. Elemen-elemen yang lebih kecil yang disebut subsistem. Misalnya, sistem komputer terdiri dari sub sistem perangkat keras, perangkat lunak, dan manusia.
- b. Elemen-elemen yang lebih besar yang disebut supersistem. Misalnya, bila perangkat keras adalah sistem yang memiliki sub sistem CPU, perangkat I/O dan memori, maka supra sistem perangkat keras adalah sistem komputer.

### 2.1.4 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen (*components*), batasan sistem (*system boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolahan (*process*), dan sasaran (*objectives*) atau tujuan (*goal*).

a. Komponen (*components*)

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

b. Batasan (*system boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar (*environments*)

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

d. Penghubung (*interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber data mengalir dari suatu subsistem ke subsistem lain. Keluaran dari subsistem akan menjadi masukan untuk subsistem lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berinteraksi dengan subsistem yang lainnya membentuk satu kesatuan.

e. Masukan (*input*)

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan sinyal dan masukan sinyal. Signal

maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang harus di proses untuk didapatkan keluaran.

f. Keluaran (*input*)

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.

g. Pengolahan (*process*)

Suatu sistem dapat mempunyai suatu bagian pengolah atau 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.

h. Sasaran (*objectives*)

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

### 2.1.5 Klasifikasi Sistem

Jogiyanto (469) menyatakan, bahwa sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

a. Sistem Abstrak dan Sistem Fisik

Sistem abstrak (*abstract system*) adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Contohnya sistem teologi, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sedangkan sistem secara fisik (*physical system*) merupakan sistem yang ada secara fisik. Contohnya sistem komputer, sistem akuntansi, sistem produksi dan lain sebagainya.

b. Sistem alamiah dan sistem buatan manusia

Sistem alamiah (*natural system*) adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia misalnya sistem perputaran bumi, sedangkan sistem buatan manusia (*human made system*) adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *man-machine system*. Contohnya sistem informasi.

c. Sistem terstruktur dan sistem tidak terstruktur

Sistem terstruktur beroperasi dengan tingkah laku yang sudah dapat diprediksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Contohnya sistem komputer yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan, sedangkan sistem tidak terstruktur adalah sistem yang kondisi masa depannya tidak dapat diproduksi karena mengandung unsur probabilitas.

d. Sistem tertutup dan sistem terbuka

Sistem tertutup (*closed system*) merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya dan bekerja secara otomatis tanpa adanya campur tangan dari pihak luarnya. Sedangkan sistem terbuka (*open system*) adalah sistem yang menghubungkan dan terpengaruh dengan lingkungan luarnya.

## 2.2 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. Kualitas informasi yang di harapkan tergantung 4 (empat) hal pokok yaitu:

a. Akurat

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

b. 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.

c. Relevan

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

d. Aman

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

### **2.2.1 Pengertian Informasi**

Informasi merupakan hasil suatu proses. Proses itu terdiri dari kegiatan-kegiatan mulai dari pengumpulan data, menyusun serta menghubungkannya, meringkas, mengambil inti sarinya, dan mengimplementasikannya agar menjadi sebuah informasi yang berguna.

McFadde (469) mendefinisikan, bahwa informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut.

Davis (469) mendefinisikan, bahwa informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.

### **2.2.2 Pengertian Sistem Informasi**

Alter (469) mendefinisikan, bahwa sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang

diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Dari definisi di atas mengenai sistem informasi dapat disimpulkan sebagai berikut:

- a. Suatu kumpulan prosedur yang memproses, mengumpulkan, menyimpan dan menyebabkan informasi untuk mendukung pengambilan keputusan dari suatu organisasi.
- b. Suatu sistem buatan manusia yang berisi komponen dan organisasi yang bertujuan untuk menyediakan informasi untuk mengendalikan dan merencanakan kegiatan perusahaan.
- c. Suatu sistem dari sebuah organisasi merupakan gabungan dari manusia, teknologi, fasilitas, media, prosedur dan pengendali yang bertujuan untuk mendapatkan jalur organisasi penting, memproses transaksi sehingga menghasilkan informasi untuk pengambilan keputusan oleh manajemen.
- d. Suatu sistem informasi mempunyai kegiatan-kegiatan sebagai berikut:
  - 1) Mengumpulkan data masukkan dari lingkungan luar organisasi.
  - 2) Melakukan proses terhadap data-data sehingga menghasilkan informasi yang berguna bagi perusahaan.
  - 3) Mendistribusikan informasi yang telah didapat kepada bagian organisasi yang memerlukan.
  - 4) Menyimpan data dan informasi dalam suatu media tertentu.

### **2.2.3 Komponen Sistem Informasi**

Dalam suatu sistem informasi terdapat komponen-komponen seperti:

- a. Perangkat keras (*hardware*)  
Mencangkup peranti-peranti fisik seperti komputer dan printer.
- b. Perangkat lunak (*software*) atau program  
Sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.
- c. Prosedur  
Sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.

d. Orang

Semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.

e. Basis data (*data base*)

Sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.

f. Jaringan komputer dan komunikasi data

Sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

## **2.3 Analisa dan Perancangan Sistem Berorientasi Obyek Dengan UML**

### **2.3.1 Unified Modeling Language (UML)**

*Unified Modeling Language* (UML) adalah salah satu kumpulan konveksi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. (Jeffery L. Whitten dkk 6:408).

Untuk membuat suatu model, UML mendefinisikan diagram-diagram berikut ini:

a. *Use Case Diagram*

b. *Class Diagram*

c. *Behaviour Diagram*

d. *Statechart Diagram*

e. *Activity Diagram*

f. *Interaction Diagram*

g. *Sequence Diagram*

h. *Collaboration Diagram*

i. *Component Diagram*

j. *Deployment Diagram*

Dalam menganalisa dan merancang sistem yang ditulis dalam tugas akhir ini, penulis tidak menggunakan semua diagram, hanya beberapa saja yang dibutuhkan seperti *use case diagram*, *activity diagram* dan *sequence diagram*.



### 2.3.2 Analisa Sistem Beorientasi Objek

#### a. Activity Diagram

Munawar (109) menyatakan bahwa :

Activity Diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak bias.

Simbol-simbol yang sering digunakan pada pembuatan activity diagram adalah sebagai berikut:

- 1) *Start point (initial node)*, tanda yang mengawali sebuah kegiatan yang diletakkan pada pojok kiri atas.



Simbol Start Point

- 2) *End point (activity final node)*, tanda yang mengakhiri sebuah kegiatan dalam activity diagram.



Simbol End Point

- 3) *Activities*, menggambarkan proses bisnis yang dikenal dengan activity state.



Simbol Activity

Jenis-jenis activities sebagai berikut:

- (a) *Black hole activities*, activity yang mempunyai masukan tetapi tidak ada keluaran. Biasanya digunakan di akhir sebuah activity diagram.



Simbol Black Hole Activities

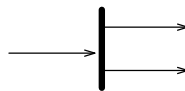
(b) *Miracle Activities*, activity yang tidak mempunyai masukan tetapi ada keluaran. Biasanya yang mengawali sebuah activity dalam activity diagram.



Simbol Miracle Activities

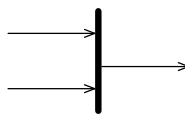
(c) *Parallel activities*, suatu activity yang berjalan secara berbarengan, yang terdiri dari:

(1) *Fork* (pencabangan), mempunyai satu transisi masuk dan dua atau lebih transaksi keluar.



Simbol Fork

(2) *Join* (penggabungan), mempunyai dua atau lebih transisi masuk dan hanya satu transisi keluar.



Simbol Join

(3) *Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true* atau *false*.



Simbol Decision Points

(4) *Guard* (kunci), sebuah kondisi atau keterangan sewaktu melewati sebuah transisi.

[ .... ]

Simbol Guard

(5) *Swimlane*, sebuah cara untuk mengelompokkan activity berdasarkan actor atau pengelompokkan activity dalam sebuah urutan yang sama.



Simbol Swimlane

#### b. Use Case Diagram

Use case diagram adalah diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem.

Use case diagram terdiri dari:

- 1) *Use case*, urutan langkah-langkah secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal.



Simbol Use Case

- 2) *Actor*, menggambarkan orang, system atau external entitas/stakeolder yang menyediakan atau menerima informasi dari sistem. Actor juga menggambarkan tugas atau peran dan bukan sebuah jabatan.



Simbol Actor

### 3) *Relationship*

Ada 4 jenis relasi yang bias timbul pada use case diagram, yaitu:

(a) Relasi antara *actor* dan *use case*, digambarkan dalam bentuk garis.

Relasi antara actor dengan use case disebut dengan asosiasi, asosiasi adalah sebuah relasi antara actor dengan *use case* dimana sebuah interaksi terjadi diantara mereka.



Relasi antara *actor* dan *use case*

(b) Relasi antara *use case* dan *use case*

Ada beberapa relasi antara *use case* dan *use case*, yaitu:

(1) *Include*, digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsional dari *use case* lainnya.

(2) *Extend*, digunakan untuk menunjukkan bahwa satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu yang dipenuhi.

#### c. Analisa Masukan

Analisa masukan adalah untuk mengetahui dokumen–dokumen apa saja yang digunakan sebagai masukan data pengolahan sistem penjualan tunai pada sistem yang berjalan.

#### d. Analisa Keluaran

Analisa keluaran merupakan analisa mengenai keluaran-keluaran yang dihasilkan melalui proses-proses yang ada dalam sistem berjalan.

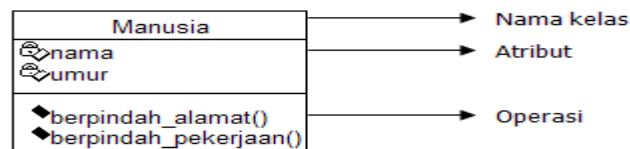
### 2.3.3 Perancangan Sistem Berorientasi Objek

#### a. *Class Diagram*

Class diagram digunakan untuk membantu visualisasi struktur kelas-kelas dari suatu sistem dan hubungan antara kelas dan penjelasan detail tiap kelas. Class diagram secara khas meliputi di bawah ini:

##### 1) Kelas (*Class*)

- 2) Relasi (*Associations, Generalization dan Aggregation*)
- 3) Atribut (*Attributes*)
- 4) Operasi (*Operations/Method*)
- 5) *Visibility*, tingkat akses objek external kepada suatu operasi atau atribut.

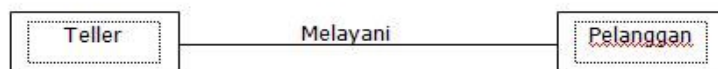


Gambar 2.2  
Contoh *Class Diagram*

Komponen-komponen *class diagram*:

1) *Association*

*Association/asosiasi* adalah class-class yang terhubungkan satu sama lain secara konseptual. Setiap *association* mempunyai dua *association end*.



Gambar 2.3  
Contoh *Association*

Sebuah *association end* juga memiliki “*multiplicity*”. *Multiplicity* menunjukkan beberapa banyak obyek yang berpartisipasi dalam suatu relasi. Secara umum, *multiplicity* menunjukkan batasan terendah dan tertinggi untuk obyek-obyek yang berpartisipasi. *Multiplicity* yang paling umum digunakan adalah 1, \*, dan 0..1.

Langkah-langkah transformasi dari conceptual data model ke tabel relasi adalah sebagai berikut :

- (a) Jika hubungan yang terjadi antar class adalah 1 to 1 (one to one) maka atribut dari *relationship* set diambil dan dimasukkan ke entitas yang lebih membutuhkan.

(b) Jika hubungan yang terjadi antar class adalah 1 to 0..1 (one to zero one) maka atribut dari *relationship* set digabung ke entitas yang memiliki *multiplicity* 0..1.

(c) Jika hubungan yang terjadi antar class adalah 1 to \* (one to many) maka atribut dari *relationship* set digabung dengan set entitas yang memiliki *multiplicity* banyak (many).

2) Atribut (*Attribute* )

*Attribute* adalah properti dari sebuah class. *Attribute* ini melukiskan batas nilai yang mungkin ada pada obyek dari class.

3) Operasi

Operasi adalah sesuatu yang bisa dilakukan oleh sebuah class atau yang ada (atau class lain) dapat dilakukan untuk sebuah class.

b. *Entity Relationship Diagram* (ERD)

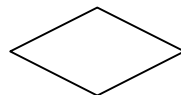
*Entity Relationship Diagram* (ERD) adalah pemodelan data utama dan akan membantu mengorganisasikan data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Di dalam sebuah ERD terdapat beberapa komponen utama, yaitu:

1) Entitas/*entity*, menggambarkan himpunan orang, tempat, obyek yang berperan di dalam sistem.



Simbol Entitas

2) Relasi, menggambarkan hubungan yang ada diantara himpunan entitas.

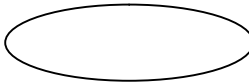


Simbol Relasi

- 3) *Cadinality*/kardinalitas, menggambarkan tingkat hubungan atau derajat relasi.

—————  
Simbol Kardinalitas

- 4) Atribut, merupakan elemen data yang memiliki sebuah entitas.

  
Simbol Atribut

c. *Logical Record Structure (LRS)*

LRS adalah sebuah bagan relasi. Sebuah relasi adalah sebuah tabel yang berisi informasi mengenai sebuah entitas. Setiap tabel harus memiliki paling tidak satu (1) key, dimana sebuah key merupakan bagian dari kelompok atribut yang memberikan nilai yang unik di dalam sebuah tabel.

d. Tabel

Sebuah relasi adalah sebuah tabel yang berisi informasi mengenai sebuah entitas. Setiap tabel harus memiliki paling tidak satu (1) key, dimana sebuah key merupakan bagian dari kelompok atribut yang memberikan nilai yang unik di dalam sebuah tabel.

e. Spesifikasi Basis Data

Menurut Jogiyanto H.M (469), bahwa:

Basis data (data base) adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer yang digunakan perangkat lunak untuk memanipulasinya.

Sedangkan sistem berbasis data adalah suatu sistem penyusunan dan pengelolaan record-record dengan menggunakan komputer dengan tujuan untuk menyimpan atau merekam serta melihat data operasional lengkap pada sebuah organisasi, sehingga mampu menyediakan informasi yang diperlukan untuk kepentingan proses pengambilan keputusan.

f. Rancangan Dokumen Masukan

Rancangan masukan merupakan data yang dibutuhkan untuk menjadi masukan sistem yang dirancang.

g. Rancangan Dokumen Keluaran

Rancangan keluaran merupakan informasi yang akan dihasilkan dari keluaran sistem yang dirancang.

h. Rancangan Layar Program

Rancangan layar program merupakan bentuk tampilan sistem layar computer sebagai antar muka dengan pemakai yang akan dihasilkan dari sistem yang dirancang.

i. *Sequence Diagram*

Setelah kita menentukan tanggung jawab dan perilaku dari objek, kita dapat menciptakan suatu model yang terperinci dari bagaimana objek tersebut akan saling berhubungan satu sama lain untuk menyediakan kemampuan atau fungsi yang ditetapkan pada setiap use case yang telah didesain sebelumnya. UML menyediakan dua jenis diagram untuk melukiskan interaksi tersebut dengan nyata. Sequence Diagram dan collaboration diagram.

Sequence Diagram menunjukkan secara detil bagaimana objek saling berhubungan satu sama lain dari waktu ke waktu, dan collaboration diagram menunjukkan bagaimana objek bekerja sama dalam bentuk urutan pesan untuk memenuhi fungsionalitas dari suatu use case. Beberapa simbol yang umum digunakan pada sequence diagram, yaitu:

1) *Entity Object*

Menurut Jeffery L. Whitten *dkk* (686), bahwa suatu objek yang berisi informasi kegiatan yang terkait yang tetap dan disimpan ke dalam suatu database.

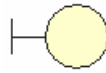


Simbol Entity Object



2) *Interface/Boundary Object*

Menurut Jeffery L. Whitten *dkk* (686), bahwa sebuah objek yang menjadi penghubung antara user dengan sistem. Contohnya window, dialogue box atau screen(tampilan layar).



Simbol Boundary Object

3) *Control Object*

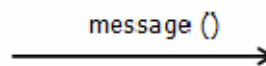
Menurut Jeffery L. Whitten *dkk* (686), bahwa:

*Control object* adalah suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas. Contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. Control object mengkoordinir pesan (message) antara boundary dengan entitas.



Simbol Control Object

4) *Simple Message*, simbol pengiriman pesan dari sebuah objek ke objek lain.



Simbol Message

5) *Recursive*, sebuah objek yang mempunyai sebuah operation kepada dirinya sendiri.



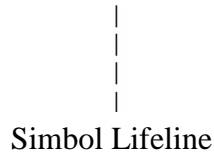
Simbol Recursive

6) *Activation*, Activation mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.



Simbol Activation

- 7) *Lifeline*, garis titik - titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.



## 2.4 Manajemen Proyek

Menurut Wysocki *dkk* (134) menyatakan, bahwa proyek adalah urutan sementara aktivitas-aktivitas unik, kompleks dan terkoneksi yang memiliki satu tujuan atau sasaran dan harus dilengkapi dengan waktu yang spesifik, sesuai anggaran, dan sesuai dengan spesifikasi.

Menurut Jeffery L. Whitten *dkk* (134) menyatakan, bahwa manajemen proyek adalah proses melingkupi, merencanakan, menyediakan staff, mengorganisasikan, dan mengontrol pengembangan sebuah sistem yang dapat diterima dengan biaya minimal dan selama jangka waktu tertentu.

Sebelum proyek dilaksanakan perlu dibuat suatu daftar yang memuat cakupan proyek (*Project Scope*). Cakupan proyek tersebut memuat beberapa hal, yaitu:

a. Sasaran proyek (*Project Objective*)

*Project Objective* yaitu hal-hal yang berisi tentang tujuan pelaksanaan proyek, hal-hal yang melandasi terciptanya sebuah proyek serta manfaat dan tujuan proyek.

b. Identifikasi *Stakeholder*

*Stakeholder* dapat berupa bagian-bagian dalam perusahaan itu sendiri maupun pihak luar perusahaan tergantung ruang lingkup proyek.

Menurut Freeman (1984), bahwa stakeholder sebagai kelompok atau individu yang dapat mempengaruhi dan atau dipengaruhi oleh suatu pencapaian tujuan tertentu.

Pada dasarnya *stakeholder* adalah pengguna atau lingkungan luar (*external entity*) dari proyek. *Stakeholder* merupakan orang ataupun organisasi yang terlibat dalam pengelolaan proyek, atau yang kepentingannya dipengaruhi oleh

keberhasilan atau kegagalan pelaksanaan atau penyelesaian proyek. *Stakeholder* ini juga bisa mempengaruhi jalannya proyek dan *deliverables* proyek. Beberapa *stakeholder* pada lingkup manajemen proyek antara lain:

- 1) *Customer / End User*
- 2) Sponsor
- 3) Tim Proyek
- 4) Calon pengguna hasil proyek
- 5) Manajer Portfolio, Manajer Program, Manajer Proyek
- 6) Manajer Operasional
- 7) Staff pendukung
- 8) Pelanggan
- 9) Pemasok

c. Identifikasi *Deliverables*

*Deliverables* diartikan sebagai hasil kerja atau output yang diharapkan dari umur hidup proyek. Dalam hal ini dapat berarti Identifikasi atau perkiraan dari hasil pekerjaan atau hasil proyek bersangkutan seperti *product* atau barang dan jasa yang dihasilkan dari proyek. Hasil pekerjaan proyek dapat berupa materi yang dapat dihitung (*tangible*) ataupun berupa hasil yang tidak dapat dihitung (*Intangible*) seperti kebanggaan dan lain sebagainya.

d. Penjadwalan proyek

Menetapkan jadwal realistis proyek dengan melakukan analisa terhadap urutan kegiatan, durasi kegiatan, dan kendala penjadwalan yang ada. Penjadwalan proyek digunakan untuk memonitor kemajuan proyek dari aspek waktu. Dalam penjadwalan proyek meliputi beberapa cakupan, yaitu:

1) *Work Breakdown Structure* (WBS)

WBS adalah teknik pemecahan kegiatan menjadi bagian-bagian yang lebih kecil dan terstruktur. WBS dapat juga diartikan sebagai penggambaran menyeluruh dari semua tugas yang diperlukan untuk mencapai penyelesaian proyek secara sukses. Dari gambaran ini kemudian disusun penjadwalan (*scheduling*), pendelegasian dan penganggaran kerja.

## 2) Milestone

Milestone digunakan untuk membagi pekerjaan menjadi lebih kecil sehingga mudah untuk dimonitoring dan dievaluasi, milestone hampir sama dengan WBS, namun dilengkapi dengan table-table pekerjaan proyek yang diambil dari WBS. Umumnya milestone menunjukkan awal kegiatan saat penyerahan produk/sebagian produk kepada *stakeholder*.

## 3) Jadwal proyek

Dalam membuat jadwal proyek dapat digunakan table tertentu dengan field sebagai berikut :

Task Name : Uraian kegiatan (*task*) dan sub-kegiatan (*sub-task*).

Duration : Durasi/lama kegiatan berlangsung.

Predecessor : Ketergantungan antar *task*.

Baseline : Rencana jadwal, yang menjadi dasar pengawasan kemajuan proyek.

Rebaselining : Membuat *baseline* baru (biasanya terjadi karena ada perubahan kebijakan dari *stakeholder* proyek).

Plan : Rencana awal jadwal, yang disimpan sebagai *baseline* proyek.

Forecast : Perkiraan tanggal penyelesaian kegiatan.

Actual : Tanggal realisasi penyelesaian.

## e. Rencana Anggaran Biaya (RAB)

RAB adalah sebuah bentuk perencanaan biaya yang dianggarkan untuk pengeluaran keseluruhan proyek. Rencana anggaran biaya dibuat dalam bentuk table berkaitan dengan seluruh pengeluaran Proyek yang telah direncanakan. Dalam beberapa model Penyusunan RAB dapat langsung disertakan Pengeluaran TIM proyek dan menghitung satuan standard minimum harga Product tertentu. Namun dalam pengembangan perangkat lunak hal tersebut agak sulit dilakukan.

## f. TIM Proyek

Kelompok yang terdiri atas beberapa orang yang memiliki berbagai macam keahlian dan dibentuk untuk meneliti rencana baru. Tujuan dibentuk tim

proyek adalah untuk memajukan kompetensi dan interaksi anggota tim untuk memajukan kinerja proyek.

g. *Analisa Resiko*

Resiko adalah hal yang tidak dapat dihindari pada suatu kegiatan atau aktivitas yang dilakukan manusia, termasuk aktivitas dalam suatu proyek. Sehingga dibutuhkan manajemen resiko yang baik dalam suatu proyek. Sasaran dari manajemen resiko proyek adalah untuk meningkatkan dampak dan kemungkinan dari kejadian positif, dan mengurangi dampak dan kemungkinan dari kejadian negatif di dalam proyek.

h. *Meeting Plan*

Rapat proyek dengan semua pihak yang terkait dengan pelaksanaan proyek. Menyangkut prosedur dan teknis pelaksanaan proyek serta semua hal yang akan berhubungan dalam pembangunan proyek.

## **2.5 Pengertian Lembaga Bimbingan Belajar**

Menurut UU No 20/2003 tentang Sistem Pendidikan Nasional ditegaskan bahwa sistem pendidikan nasional terdiri atas jalur pendidikan Formal, Nonformal dan Informal. Lembaga Bimbingan Belajar masuk dalam jalur Pendidikan Non Formal (PNF) sedangkan pembinaannya masuk pada tanggung jawab Ditjen Diklusepora dalam Direktorat Pembinaan Kursus dan Kelembagaan.

Sedangkan aturan lebih rinci untuk pembinaan terhadap kursus ini diatur dalam Keputusan Menteri Pendidikan dan Kebudayaan ( Kepmendikbud ) Nomor 0151/U/1977 tentang Pokok-pokok Pelaksanaan Pembinaan Program Pendidikan Luar Sekolah yang diselenggarakan masyarakat.

Berdasarkan uraian tersebut dapat didefinisikan lembaga bimbingan belajar adalah instansi yang memberikan layanan jasa bimbingan yang memungkinkan siswa mengembangkan diri dengan sikap dan kebiasaan belajar yang baik, materi belajar yang cocok dengan kecepatan dan kesulitan belajar atau dapat mengatasi kesulitan belajar (P3G, 1996 : 6).

## 2.6 Perangkat Lunak Pendukung

Perangkat lunak pendukung merupakan perangkat lunak atau software yang digunakan dalam membuat sistem analisa dan perancangan sistem informasi pengolahan data bimbingan belajar pada lembaga bimbingan belajar Smart Learning Center Pangkalpinang.

### 2.6.1 Microsoft Visual Basic 2008

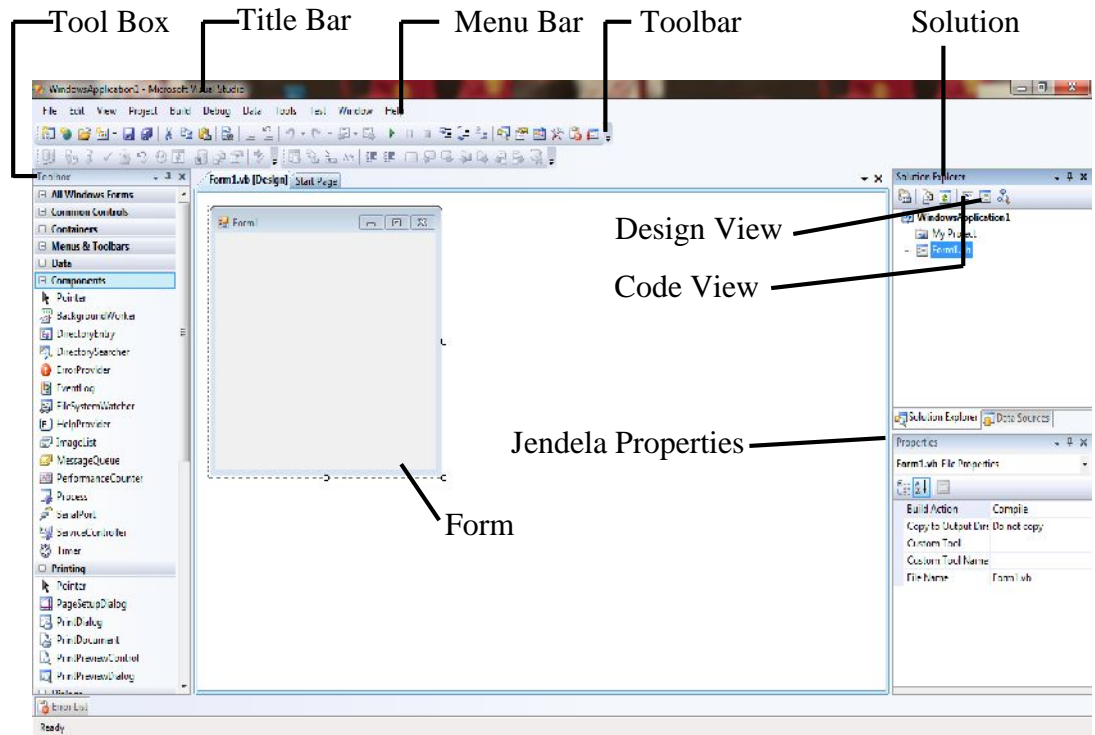
*Visual Basic* merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat.

*Visual Basic* adalah pengembangan dari bahasa komputer BASIC (*Beginner's All-Purpose Symbolic Instruction Code*). Bahasa *Basic* diciptakan oleh Professor John Kemeny dan Thomas Kurtz dari Perguruan Tinggi Dartmouth pada pertengahan tahun 1960. Bahasa program tersebut tersusun mirip dengan bahasa Inggris yang biasa digunakan oleh para programmer untuk menulis program-program komputer sederhana yang berfungsi sebagai pembelajaran bagi konsep dasar pemrograman komputer.

Sejak saat itu, banyak versi BASIC yang dikembangkan untuk digunakan pada berbagai platform komputer, seperti Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASICA, Apple BASIC dan lain-lain. Popularitas dan pemakaian BASIC yang luas dengan berbagai jenis komputer turut berperan dalam mengembangkan dan memperbaiki bahasa itu sendiri, dan akhirnya berujung pada lahirnya *Visual Basic* yang berbasis *Graphic User Interface* (GUI) bersamaan dengan *Microsoft Windows*. Pemrograman *Visual Basic* begitu mudah bagi pemula dan programmer musiman karena ia menghemat waktu pemrograman dengan tersedianya komponen-komponen siap pakai. Hingga akhirnya *Visual Basic* juga telah berkembang menjadi beberapa versi.

*Visual Basic 2008* merupakan salah satu paket bahasa pemrograman dari *Visual Studio 2008*. *Visual Studio 2008* merupakan sebuah software untuk membuat aplikasi *Windows*, seperti aplikasi database, aplikasi desktop, aplikasi inventory, dan sebagainya.

## 2.6.2 Elemen - elemen Microsoft Visual Basic 2008



Gambar 2.4

Tampilan Microsoft Visual Basic 2008

- Tool Box**  
Merupakan kotak perangkat yang terdiri atas beberapa *class object* yang digunakan dalam proses pembuatan aplikasi.
- Title Bar**  
Merupakan batang judul yang terletak pada bagian atas jendela program Visual Basic yang berfungsi untuk menunjukkan nama proyek yang sedang aktif.
- Menu Bar**  
Merupakan batang menu yang berisi menu-menu utama, seperti File, Edit, View, Project, dan lain-lain yang berfungsi untuk mengoperasikan program *Visual Basic 2008*.

d. Toolbar

Merupakan sebuah batang tool yang berisi tombol-tombol dengan gambar ikon yang dapat diklik untuk melakukan suatu perintah khusus secara cepat.

e. Form

Merupakan tempat yang digunakan untuk merancang aplikasi yang sedang dibuat. Didalam form kita dapat merancang sebuah program aplikasi dengan menempatkan kontrol-kontrol yang ada di bagian toolbox.

f. Solution Explore

Solution Explorer adalah jendela yang menyimpan Informasi mengenai Solution, Project – project, beserta file-file, form – form ataupun resource yang digunakan pada program aplikasi. Pada bagian atas jendela Solution Explorer terdapat toolbox yang digunakan untuk menampilkan jendela Properties, menampilkan semua file, melihat Design form, Refresh dan View code, untuk melihat kode program.

g. Design View

Merupakan tempat untuk melihat desain form yang telah dibuat.

h. Code View

Merupakan tempat untuk menulis *coding*. Untuk menampilkan cepat jendela ini dengan menggunakan kombinasi Shift-F7.

i. Jendela Properties

Merupakan jendela yang digunakan untuk menampilkan dan mengubah properti-properti yang dimiliki sebuah objek. Pada jendela properties terdapat dua pilihan tabulasi, yaitu Alphabetic (urut berdasarkan abjad) dan Categorized (urut berdasarkan kelompok).

### 2.6.3 Database Microsoft Office Access 2007

Basis data atau *database* adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. *Database* juga merupakan kumpulan tabel. Satu tabel merepresentasikan suatu entitas tertentu. Suatu entitas terdiri atas beberapa atribut.



*Microsoft Office Access 2007 (MS Access 2007)* adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. *Microsoft Office Access 2007* merupakan versi dari *Microsoft office acces* yang mendukung untuk sistem operasi *Microsoft Windows XP (Service Pack 2)/Server 2003/Vista/Server 2008 (Beta 2 ke atas)*.

*Database Microsoft Office Access 2007* terdiri dari sekumpulan objek yang terdiri dari Tabel, *Query, Form, Report, Pages, Macro dan Module*. Objek-objek ini ditampung dalam satu wadah atau *database*. Selain itu, dalam *database MS acces 2007* juga terdapat group.