BAB II
LANDASAN TEORI

2.1 Definisi Sistem informasi

Sistem kebanyakan dapat didefinisikan secara sederhana sebagai sekelompok elemen yang saling berhubungan atau berinteraksi hingga membentuk satu kesatuan. Akan tetapi, konsep umum sistem berikut ini memberikan konsep dasar yang lebih tepat untuk bidang Sistem Informasi.

Sistem adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam proses transformasi yang teratur. (O'brien 2006:29)

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, antara lain sebagai berikut :

a. Komponen Sistem (*Component*)

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

b. Batas Sistem (*Boundary*)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luar. Batas suatu sistem menunjukkan lingkup (*scope*) dari sistem tersebut.

c. Lingkungan luar (*environtments*)

   Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi dari sistem.

d. Penghubung (*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem yang lain untuk dapat berinteraksi membentuk suatu kesatuan.

e. Masukan (*input*)

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

f. Pengolahan (*process*)

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

g. Keluaran (*output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

h. Sasaran (*objective*)

Suatu sistem harus mempunyai sasaran, karena sasaran sangat menentukan sekali masukan yang dibutuhkan oleh sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil apabila mengenai sasaran atau tujuan.

Informasi adalah hasil dari pengolahan data dalam bentuk yang lebih berguna dan 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 (*event*) adalah suatu yang terjadi pada saat tertentu. Kesatuan nyata (*fact*) adalah berupa suatu obyek nyata seperti tempat, benda atau orang yang benar-benar ada dan terjadi.

Menurut Jogiyanto HM, MBA, Akt., Ph.D. (2005) :

"Informasi (information) adalah data yang diolah menjadi bentuk yang yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan."

Untuk dapat berguna, maka informasi harus didukung oleh tiga pilar, yaitu sebagai berikut :

a. Akurat (*accurate*)

   Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.

b. Tepat waktu (*Timeliness*)

   Tepat pada waktunya berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan didalam pengambilan keputusan.

c. Relevan (*Relevance*)

   Relevan berarti informasi tersebut mempunyai manfaat untuk pemakianya. Relevansi informasi untuk tiap-tiap orang berbeda. Nilai informasi bagi seorang pemakai ditentukan oleh keandalan (*reliabilitas*). Keluaran yang tidak didukung oleh ketiga pilar ini tidak dapat dikatakan sebagai informasi yang berguna, tetapi merupakan sampah (*garbage*).

Sistem Informasi dapat didefinisikan sebagai suatu susunan dari orang, data, proses, dan teknologi informasi yang saling berhubungan untuk mengumpulkan, memroses, menyimpan, dan menyediakan keluaran informasi yang diperlukan untuk mendukung suatu organisasi. Sistem informasi dapat digolongkan menurut fungsinya, antara lain adalah sebagai berikut ini:(Whitten 2004:12)

a. *Transaction Processing System* (TPS), suatu sistem informasi yang menangkap dan memproses data tentang transaksi bisnis. seperti pesanan (*order*), kartu catatan waktu, pembayaran, reservasi, dan sebagainya.(Whitten 2004:12)

b. *Management Information System* (MIS), suatu sistem informasi yang disediakan untuk menghasilkan laporan yang berorientasi pada manajemen yang berdasarkan pada proses transaksi dan operasi dari organisasi. Atau dengan kata lain menggunakan data transaksi untuk menghasilkan informasi yang dibutuhkan oleh manajer untuk menjalankan bisnis.(Whitten 2004:12)

c. *Decision Support* System (DSS), suatu sistem informasi yang membantu mengidentifikasi pengambilan keputusan yang mungkin atau menyediakan informasi untuk membantu pengambilan keputusan manajemen.(Whitten 2004:12)

d. *Executive Information System* (EIS), suatu sistem informasi yang mendukung perencanaan dan kebutuhan penilaian dari manajer eksekutif. EIS dikhususkan untuk kebutuhan informasi yang unik dari para eksekutif yang merencanakan bisnis dan menilai capaian rencana bisnis tersebut.(Whitten 2004:13)

e. *Expert System* (ES), suatu sistem informasi yang menangkap keahlian dari para pekerja dan kemudian menirukan keahlian tersebut untuk dimanfaatkan oleh orang yang tidak ahli.(Whitten 2004:14)

b. 6) *Communications and Collaboration System*, suatu sistem informasi yang memberikan peluang komunikasi yang lebih efektif antara para pekerja, mitra, pelanggan, dan para penyalur untuk meningkatkan kemampuan mereka untuk bekerja sama. (Whitten 2004:14)

a. *Office Automation System*, suatu sistem informasi yang mendukung cakupan luas dari aktivitas kantor yang disediakan untuk meningkatkan alur kerja (*work flow*) antara para pekerja dan membantu karyawan membuat dan membagi dokumen yang dapat mendukung aktivitas kantor sehari-hari. (Whitten 2004:14)

Adapun kegiatan dari sistem informasi antara lain adalah:

a. *Input*, Menggambarkan suatu kegiatan untuk menyediakan data untuk diproses.
b. *Proses*, Menggambarkan bagaimana suatu data di *proses untuk menghasilkan* suatu informasi yang bernilai tambah.
c. *Output*, Suatu kegiatan untuk menghasilkan laporan dari proses di atas tersebut.
d. Penyimpanan, Suatu kegiatan untuk memelihara dan menyimpan data.
e. *Control*, Suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

2.2 Siklus Sistem Informasi

Siklus merupakan putaran waktu yang di dalamnya terdapat rangkaian kejadian yang berulang-ulang secara tetap dan teratur. Siklus sistem informasi merupakan proses menghasilkan informasi harus melalui tahapan-tahapan yang dilakukan komputersebagai teknologi informasi. Tahapan –tahapan tersebut terdiri atas Input proses output yang disebut sebagaisiklus sistem informasi. Artinya, bila tahap telah sampai pada output tersebut dapat dijadikan input kembali.Dengan demikian dapat dikatakan bahwa informasi yang dihasilkan dapat pula dijadikan data kembali sebagai inputuntuk diproses selanjutnya.

2.3 Subsistem Sistem Informasi

Komponen Sistem Informasi adalah sebagai berikut:

a. Perangkat Keras (*Hardware*), Terdiri dari komputer, *peripheral*, jaringan, dsb.
b. Perangkat Lunak (*Software*), Merupakan kumpulan dari perintah/fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu. *Software* dapat digolongkan menjadi Sistem Operasi (Windows 2000, Linux, Unix, dll), Aplikasi (Akuntansi, database, dll), Utilitas (Anti Virus, Speed Disk, dll), serta Bahasa (Java, VB, Delphi, C++, dll).

c. Data, Merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.
d. Prosedur, Dokumentasi prosedur / proses sistem, buku penuntun operasional (aplikasi) dan teknis.
e. Manusia (*Human*), Yang terlibat dalam komponen manusia seperti operator, pemimpin sistem informasi dan sebagainya. Oleh sebab itu perlu suatu rincian tugas yang jelas.

2.4. Analisa dan Perancangan Sistem Berorientasi Objek dengan UML

Analisa sistem dapat dinyatakan sebagai pemisahan suatu hal dalam bagian bagian tertentu. Bagian-bagian tersebut kemudian dipelajari dan dievakuasi untuk mengetahui apakah terdapat cara-*cara yang lebih baik untuk* memenuhi kebutuhan manajemen.

"Analisa sistem adalah proses menentukan kebutuhan sistem – apa yang harus dilakukan sistem untuk memenuhi kebutuhan klien, bukanlah bagaimana sistem tersebut diimplementasikan." (Ariesto Hadi Sutopo, 2002:242):

Konsep dasar berorientasi obyek mencapai kematangannya pada saat masalah analisis dan desain menjadi lebih diperhatikan dari pada masalah coding. Secara spesifik, pengertian "berorientasi obyek" (Ariesto Hadi Sutopo, 2002:3) berarti bahwa "kita mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya".

2.4.1 Unified Modelling Language

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi danjaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class*

dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi obyek. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya : Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Cakupan UML diantaranya : Pertama, UML menggabungkan konsep BOOCH, OMT, dan OOSE, sehingga UML merupakan suatu bahasa permodelan tunggal yang umum dan digunakan secara luas oleh para user ketiga metode tersebut dan bahkan para user metode lainnya. Kedua, UML menekankan pada apa yang dapat dikerjakan dengan metode-metode tersebut. Ketiga, UML berfokus pada suatu bahasa permodelan standar, bukan pada proses standar.

2.4.2 Analisa Sistem Berorientasi Objek

Pendekatan dalam analisa berorientasi objek dilengkapi dengan alat-alat dan teknik yang dibutuhkan dalam pengembang hasil akhir dari sistem yang dikembangkan akan didapatkan sistem yang dapat terdefinisi dengan baik dan jelas.

Maka Landasan teori diagram – diagram UML yang menjadi alat bantu pada tahap analisa berorientasi objek yaitu :

a. Activity diagram

Diagram memodelkan alur kerja (*work floaw*) sebuah proses bisnis dan urutan aktivitas pada suatu proses. Diagram ini sangat mirip dengan *flow chart* karena kita dapat memodelkan prosedur logika, proses bisnis dan alur kerja. Perbedaan utamanya adalah *flow chart* dibuat untuk menggambarkan alur kerja dari sebuah sistem, sedangkan *activity diagram* dibuat untuk menggambarkan aktivitas dari aktor.

Activity diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai persan seperti halnya *flow chart*, akan tetapi perbedanya dengan *flow chart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flow chart* tidak bisa.

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

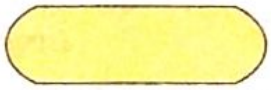
1) *Start Point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.

Simbol Start Point

2) *End Point*, diletakkan dimana saja dan menggambarkan berakhirnya aktifitas

Simbol End Point

3) *Activities*, menggambarkan suatu proses / kegiatan yang lebih dikenal dengan *activity state*.

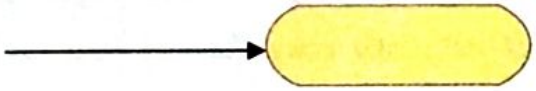


Simbol Activities

Jenis jenis *activities*, yaitu :

a) *Black hole activities*

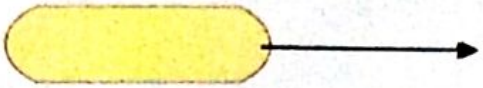
Ada masukan dan tidak ada keluaran,biasanya digunakan jika dikehendaki ada satu atau lebih transisi.



Simbol Black Hole Activities

b) Miracle activities

Tidak ada masukan dan ada keluaran,biasanya dipakai pada waktu *start point* dan dikehendaki ada 1 atau lebih transisi.



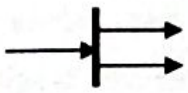
Simbol Miracle activities

c) *Parallel Activities*

Suatu *activity* yang berjalan secara bersamaan terdiri dari :

(1) Fork (Percabangan)

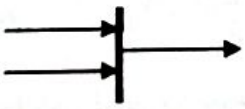
Fork digunakan untuk menunjukkan kegiatan yang dilakukan secara *parallel* atau untuk menggabungkan dua kegiatan *parallel* menjadi satu



Simbol Fork

(2) *Join* (Penggabungan)

Mempunyai 1 atau lebih transisi masuk dan hanya 1 transisi keluar dan *fork* harus berhubungan dengan join.



Simbol End Point

4) *Decision Point*

Decision digambarkan dengan lambang wajik atau belah ketupat. Mempunyai transisi (sebuah garis dari atau ke *decision point*). Setiap transisi yang ada harus mempunyai *guard* (kunci).Tidak ada keterangan (pernyataan) pada tengah belah ketupat seperti pada *flowchart*.

Simbol Decision

5) *Guard* (kunci)

Guard (kunci) adalah kondisi benar sewaktu melewati transisi. Digambarkan dengan diletakkan diantara tanda [ ]. Tanda *otherwise guard* untuk menangkap suatu kondisi yang belum terdeteksi. Setiap transisi dari atau ke *decision point* harus mempunyai *guard* yang harus konsisten dan lengkap serta tidak *overlap.*

6) Swimlane

Swimlane merupakan sebuah cara untuk mengelompokkan *activity* berdasarkan actor. Actor bisa ditulis nama *actor* ataupun sekaligus dalam lambang *actor*. *Swimlane* digambarkan secara vertical, walaupun kadang kadang digambarkan secara horizontal.

b. Analisa Dokumen Keluaran

Analisa keluaran adalah analisa mengenai dokumen – dokumen keluaran yang dihasilkan dari sebuah sistem.

c. Analisa Dokumen Masukan

Analisa dokumen masukan adalah bagian dari pengumpulan informasi tentang sistem yang sedang berjalan. Tujuan analisa masukan adalah memahami prosedur berjalan.

d. Use Case Diagram

Use case diagram menggambarkan sebuah *fungsionalitas* yang diharapkan dari sebuah sistem dan bagaimana sistem berinteraksi dengan dunia luar yang ditekankan dalam *use case* diagram adalah "apa" yang diperbuat system, dan bukan "bagaimana" sistem itu melakukannya.Sebuah *use case* mempersentasikan

sebuah interaksi antara *actor* dengan sistem *use case* diagram juga menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor). Use case* merupakan sebuah pekerjaan tertentu, misalnya *log in* kesistem, meng-*create* sebuah daftar belanja dan sebagainya.

Secara umum *use case* diagram terdiri dari :

1) Aktor (*actor*)

   Actor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem ,untuk mengidentifikasikan *actor* harus ditentukan pembagian kerja tugas – tugas yang berkaitan dengan peran konteks target system *actor* dilukiskan dengan peran yang mereka mainkan dalam *use case*. Seperti staff penjualan ,pelanggan,dll.

Simbol *Actor*

2) Use case

   Use case menggambarkan perilaku ,termasuk didalamnya interaksi antara *actor* dengan sistem. *Use case* dibuat berdasarkan keperluan *actor*, merupakan "apa" yang dikerjakan sistem bukan "bagaimana" sistem mengerjakannya. Setiap *use case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan *actor*. Nama *use case* boleh terdiri dari beberapa kata dan tidak boleh ada dua *use case* yang memiliki nama yang sama.

Simbol *Use Case*

3) *Asosiasi (Asociation)*

   Asosiasi menggambarkan aliran data atau informasi asosiasi atau relasi juga digunakan untuk menggambarkan bagaimana *actor* terlibat dalam *use case*. Relasi (*relationship*) digambarkan sebagai bentuk garis antara dua *symbol use case diagram.*

Simbol *Asociation*

Ada empat jenis relasi / asosiasi yang dapat timbul pada *use case* diagram,yaitu :

a) Asosiasi antara *actor* dan *use case*

Ujung panah pada *association* antara *actor* dan *use case* mengindikasikan siapa/ apa yang meminta interaksi dan bukannya mengindikasikan aliran data. sebaiknya gunakan garis tanpa panah untuk *association* antara *actor* dan *use case, association* antara *actor* dan *use case* yang menggunakan panah untuk mengindikasikan bila *actor* berinteraksi secara pasif dengan sistem

Simbol *Asosiasi* antara *actor* dan *use case*

b) Asosiasi antara *use case*

Relasi antara *use case* dengan *use case* :

(1) *Include*, menggambarkan suatu *use case* termasuk didalam *use case* lain (diharuskan). Contohnya adalah pemanggilan sebuah fungsi program. Digambarkan dengan garis lurus berpanah dengan tulisan *<<include>>*.

(2) *Extend*, digunakan ketika hendak menggambarkan variasi pada kondisi perilaku normal dan menggunakan lebih banyak control form dan mendeklarasikan *extension* pada *use case* utama atau dengan kata lain adalah perluasan dari use case lain jika syarat atau kondisi terpenuhi. Digambarkan dengan garis lurus berpanah dengan tulisan *<<extend>>*

(3) *Generalization / Inheritance* , digambarkan dengan garis lurus berpanah tertutup dari *base use case* ke *parent use case.*

c) *Generalization / Inheritance* antar *use case*

Generalization dipakai ketika ada sebuah perlakuan khusus *(single condition)* dan merupakan pola hubungan *base-parent use case.* Digambarkan dengan *generalization / inheritance* antar *use case* secara vertical dengan *inheriting use case* dibawah *base /parent use case.*

d) *Generalization / Inheritance* antar *actor*

Generalization / Inheritance antar *actor* dibuat ketika ada sebuah *actor* terbentuk dan mempunyai atribut dan metode yang sama dengan *actor* yang sudah ada.

e. Deskripsi Use Case Diagram

Deskripsi *Use Case Diagram* adalah abstraksi dari interaksi antar sistem dan *actor*. Oleh karena itu sangat penting untuk memilih abstraksi yang cocok. Sebagai contoh, saat pelanggan menelpon restoran untuk melakukan *booking*, dia akan berbicara kepada karyawan restoran yang akan mencatat *booking* tersebut ke sistem.

Untuk melakukan hal tersebut, karyawan akan menjalankan peran sebagai resepsionis meskipun pekerjaan tersebut mungkin bukan pekerjaan formalnya. Pada situasi ini, karyawan merupakan *instance* dari *actor resepsionis* dan interaksi diantara karyawan dengan system adalah instance dari use case. Use case dibuat berdasrkan keperluan *actor*. *Use case* harus merupakan 'apa' yang dikerjakan *software* aplikasi,bukan 'bagaimana' *software* aplikasi mengerjakannya.

Setiap *use case* harus diberi nama yang menyatakan apa hal yang ingin dicapai dari hasil interaksinya dengan *actor* . Nama *use case* boleh terdiri dari beberapa kata dan tidak boleh ada dua *use case* yang memiliki nama yang sama.

2.4.3 Perancangan Sistem Berorientasi Objek

Perancangan berorientasi objek merupakan tahap lanjutan setelah analisa berorientasi objek, perancangan berorientasi objek adalah suatu pendekatan yang digunakan untuk menspesifikasi kebutuhan – kebutuhan system dengan menkolaborasikan objek – objek , atribut – atribut dan *method – method* yang ada. (Whitten 2008:686)

Tujuan perancangan sistem itu untuk memahami kebutuhan kepada pemakai sistem (*user*) dan memberikan gambaran yang jelas serta rancang bangun yang lengkap.

a. ERD

ERD adalah sebuah model data yang menggunakan beberapa notasi untuk menggambarkan data dalam hal entitas dan relasi yang digambarkan oleh data tersebut (Whitten 2004:295). Elemen – elemen ERD yaitu sebagai berikut :

1) Entity (Entitas)

Sesuatu (obyek) yang ada didalam sistem. Entitas merupakan kata benda yang dikelompokkan menjadi empat jenis nama, yaitu : orang, benda, lokasi dan kejadian. Entitas disimbolkan dengan persegi panjang.

Simbol Entitas

2) *Relationship* (Hubungan atau relasi)

Simbol *Relationship* pada ERD digambarkan dengan *diamond* atau *decision*. Jika satu entitas dihubungkan dengan *Relationship*, maka digambarkan dengan garis lurus.Kumpulan dari *Irelationship* yang sejenis disebut *Relationship set*.

Simbol Relasi

3) *Atribute* (atribut)

Suatu deskripsi karakteristik dari entitas (Whitten 2004:296). Atribut juga merupakan karakteristik dari relationship, maksudnya sesuatu

yang menjelaskan apa yang sebenarnya yang dimaksud dengan *entitas* maupun *relationship*. Atribut disimbolkan dengan sebuah ellips. Dari setiap atribut entitas terdapat satu atribut yang dijadikan sebuah *key* (kunci). Beberapa jenis *key*,yaitu :

a) *Primary Key*

Field yang mengidentifikasikan sebuah record dalam file dan bersifat unik.

b) *Secondary Key*

Field yang mengidentifikasikan sebuah record dalam file dan tidak bersifat unik.

c) *Candidate key*

Beberapa field yang dapat dijadikan calon *primary key*.

d) *Alternative Key*

File dari *candidate key* yang tidak terpilih jadi *primary key*.

e) *Composite Key*

Beberapa field yang digabungkan untuk membentuk *primary key*.

f) *Foreign Key*

Field yang bukan *key* , tetapi merupakan *key* pada file lain.

4) *Cardinality* (Kardinaliti)

Ada 3 kemungkinan hubungan yang ada yaitu :

a) One To One

Artinya tingkat hubungan dimana satu kejadian pada entitas yang pertama hanya mempunyai satu hubungan dengan satu kejadian pada entitas kedua, demikian juga sebaliknya.

b) One To Many

Artinya tingkat hubungan dimana satu kejadian pada entitas pertama mempunyai banyak hubungan dengan kejadian pada entitas kedua, demikian juga sebaliknya.

c) Many To Many

Artinya tingkat hubungan dimana tiap kejadian sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya.

b. Logical Record Structure (LRS)

Sebuah model sistem yang digambarkan dengan sebuah Diagram-ER akan mengikuti pola aturan pemodelan tertentu. Dalam kaitannya dengan konversi ke LRS, maka perubahan yang terjadi adalah mengikuti aturan – aturan berikut ini :

1) Setiap entitas akan diubah ke bentuk kotak.
2) Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi bersatu dengan *cardinality* M) atau tingkast hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi), sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungannya M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumya saling berhubungan.

c. Tabel

Tabel adalah koleksi objek yang terdiri dari sekumpulan elemen yang diorganisasi secara kontinyu,artinya memori yang dialokasi antara satu elemen dengan elemen yang lainnya mempunyai *address* yang berurutan.Pada table , pengertian perlu dipahami adalah :

1) Keseluruhan table (sebagai koleksi) adalah *container* yang menampung seluruh elemen.
2) Indek table, yang menunjukkan *address* dari sebuah elemen.
3) Element table, yang dapat dipacu melalui indeknya, bertipe tertentu yang sudah terdefinisi.
4) Seluruh elemen table bertipe "sama". Dengan catatan: beberapa bahasa pemrograman memungkinkan pendefinisian table dengan elemen genetik, tapi pada saat diinstansiasi harus dengan tipe yang sama.

d. Spesifikasi Basis Data

Basis data merupakan kumpulan dari data yang *saling berhubungan satu* dengan yang lain dan tersimpan diluar computer serta digunakan perangkat lunak (*software*) tertentu untuk memanipulasinya. Sedangkan system berbasis data adalah suatu sistem penyusunan dan pengelolaan *record-record* dengan meggunakan 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.

e. Rancangan Dokumen Keluaran

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

f. Rancangan Dokumen Masukan

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

g. Rancangan Layar Program

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

h. Sequence Diagram

Sequence diagram adalah suatu diagram UML yang memodelkan logika dari suatu *use case* dengan menggambarkan interaksi berupa pemrograman pesan (*message*) antar obyek dalam urutan waktu (Whitten 2004:702).

Beberapa simbol yang umum digunakan pada sequence diagram, yaitu:

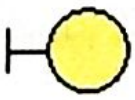
1) *Actor*, menggambarkan orang yang sedang berintraksi dengan sistem

Simbol *Actor*

2) *Entity Object*, suatu obyek yang berisi informasi kegiatan yang terkait yang tetap dan disimpan ke dalam suatu *database*.(Whitten 2004:686)

Simbol *Entity Object*

3) *Interface/Boundary Object*, sebuah obyek yang menjadi penghubung antara user dengan sistem. Contohnya *window*, *dialogue box* atau *screen* (tampilan layar).(Whitten 2004:686)



Simbol *Boundary Object*

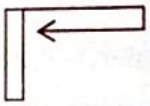
4) *Control Object*, suatu obyek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas. contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai obyek. *Control object* mengkoordinir pesan (*message*) antara *boundary* dengan entitas.(Whitten 2004:686)

Simbol *Control*

5) *Simple Message*, simbol pengiriman pesan dari sebuah obyek ke obyek lain.(Whitten 2004:704)

Simbol *Message*

6) *Recursive*, sebuah obyek yang mempunyai sebuah *operation* kepada dirinya sendiri.(Munawar 2005:89)



Simbol *Recursive*

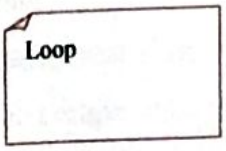
7) *Activation*, mewakili sebuah eksekusi operasi dari obyek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.(Munawar 2005:87;89)

Simbol *Activation*

8) *Lifeline*, garis titik-titik yang terhubung dengan obyek, sepanjang *lifeline* terdapat *activation*. (Munawar 2005:87;89)

Simbol *Lifeline*

9) *Loop*, menggambarkan suatu kegiatan yang dilakukan secara berulang-ulang.



Simbol *Loop*

i. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/property) suatu system, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur objek system,dimana diperlihatkan hubungan antara mereka.

Class diagram memiliki tiga area pokok yaitu nama, atribut, method, ditambah lagi asosiasi sebagai penghubung.

1) Class name, merupakan nama dari sebuah class
2) Atribut

   Atribut adalah sebuah nilai data yang dimiliki oleh *class*. Nama, umur,berat badan adalah atribut dari obyek manusia atau orang. Setiap atribut memiliki nilai untuk obyek *instance*.
3) *Method*

Method adalah sesuatu yang bisa dilakukan oleh kelas atau implementasi dari sebuah operasi kedalam sebuah kelas.

4) Asosiasi

Asosiasi adalah *class – class* yang berhubungan secara konseptual. Setiap asosiasi mempunyai dua *association end.* Sebuah *association end* juga memiliki "multiplicity" yang menunjukkan beberapa banyak objek yang berpartisipasi dalam satu relasi. Multiplicity menunjukkan batasan terendah dan tertinggi untuk objek – objek yang berpartisipasi. Multiplicity yang paling umum digunakan adalah 1, *, dan 0.. 1.

Hubungan antar class antara lain :

a) Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah *query* antar *class*.
b) Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..")
c) Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
d) Hubungan dinamis, yaitu rangkaian pesan (message) yang dipassing dari satu class kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence* diagram.

2.5 Teori Pengelolaan Proyek

Definisi Manajemen Proyek menurut PMBOK (*Project Management Body of Knowledge*) adalah aplikasi dari pengetahuan, keahlian, alat – alat, dan tehnik untuk melaksanakan aktivitas sesuai dengan kebutuhan proyek.

a. *Project Execution Plan* (PEP), Sebuah rencana eksekusi suatu proyek sangat erat kaitannya dengan estimasi biaya, dimana keduannya saling bergantung

dan tidak aka terpenuhi keduanya secara total jika satu diantara keduanya tidak terselesaikan.

b. *Objective* proyek (tujuan proyek)

c. *Stakeholders* dapat diartikan sebagai segenap *pihak yang terkait dengan isu* dan permasalahn yang sedang diangkat.

d. *Delivarable* adalah produk yang diproduksi sebagai bagian dari proyek seperti perangkat keras, dokumen perencanaan, atau rapat.

e. Jadwal proyek, Penjadwalan proyek adalah kegiatan menetapkan jangka waktu kegiatan proyek yang harus diselesaikan, bahan baku, tenaga kerja serta waktu yang dibutuhkan oleh setiap aktivitas.

f. *Work Breakdown Structure* (WBS), adalah cara *pengorganisasian proyek* menjadi bagan/struktur pelaporan yang bersifat hirarkis. WBS berfungsi untuk melakukan *breakdown*atau memecahkan masalah setiap proses pekerjaan menjadi lebih baik dan sempurna. Prinsip dasar dari *Work Breakdown Structure* (WBS) adalah pemecahan atau pembagian pekerjaan ke dalam bagian yang lebih kecil.

g. *Milestone,* adalah suatu bagian item pekerjaan yang dibuat seolah – olah menjadi *temporary finish*atau selesai sementara atas sekelompok atau serangkaian pekerjaan – pekerjaan yang menjadi bagian dari *schedule* besar. Item pekerjaan yang dijadikan *milestone*haruslah item pekerjaan yang dianggap menjadi bagian penting sebelum melanjutkan pekerjaan berikutnya atau berpengaruh atas kelangsungan pekerjaan berikutnya.

h. Rencana Anggaran Biaya (RAB), adalah perhitungan banyaknya biaya yang diperlukan untuk bahan dan upah, serta biaya – biaya lain yang berhubungan dengan pelaksanaan bangunan atau proyek. Anggaran biaya merupakan harga dari bahan bangunan yang dihitung dengan teliti, cermat, dan memenuhi syarat. Anggaran biaya pada bangunan yang sama akan

berbeda – beda di masing – masing daerah disebabkan karena perbedaan harga bahan dan upah tenaga kerja.

i. Organisasi matriks adalah suatu usaha untuk menggabungkan keuntungan dari struktur fungsional murni dan struktur produk organisasi. Dalam organisasi matriks, setiap manajer proyek melaporkan secara langsung kepada wakil presiden dan manajer umum karena setiap proyek merupakan *profit center*yang potensial, kekuasaan, dan otoritas yang digunakan oleh manajer proyek datang langsung dari manajer umum.

j. Analisa resiko (*Project risk*), menjelaskan proses – proses yang berhubungan dengan pengidentifikasian resiko, kuantifikasi resiko, penyusunan penanggulangan resiko dan pengendalian penanggulangan resiko.