

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Sistem adalah Serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem tersusun dari sub-sub sisitem yang lebih kecil yang juga saling tergantung dan bekerja sama untuk mencapai tujuan. Tujuan dasar suatu sistem tergantung pada jenis sistem itu sendiri. Sebagai contoh, sistem peredaran darah manusia merupakan sistem biologi yang memiliki tujuan untuk mengedarkan darah yang mengandung oksigen dan sari makanan keshuruh tubuh. Sedangkan sistem buatan manusia seperti sistem yang terdapat di sekolah, organisasi bisnis, atau instansi pemerintah juga mempunyai tujuan yang berbeda-beda. (Anastasia Diana dan Lilis Setiawati : 2011 : 3)

II.1.1. Informasi

Informasi adalah data yang diolah menjadi suatu bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengam,bilan keputusan pada saat sekarang atau yang akan datang. Informasi juga merupakan fakta-fakta atau data yang telah diproses sedemikian rupa atau mengalami proses transformasi data sehingga berubah bentuk menjadi informasi.

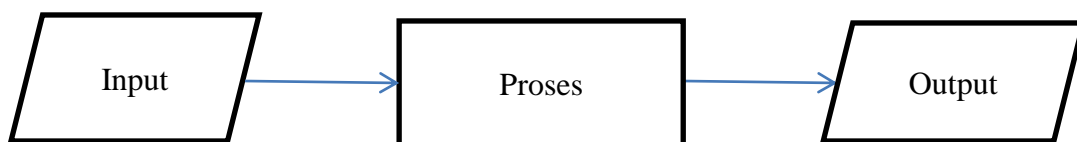
Kualitas dari suatu informasi tergantung pada tiga hal yaitu :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan.

2. Tepat pada waktunya, berarti informasi yang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda. (Sulindawati : 2010: 3)

II.1.2. Sistem Informasi

Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar. Komponen suatu sistem terdiri dari Input, Proses dan Output. Komponen sistem informasi dapat dilihat pada gambar II.1:



Gambar II.1. Komponen Sistem Informasi

(Sumber : Anastasia Diana & Lilis Setiawati ; 2011)

II.2. Produksi

Produksi adalah bidang yang terus berkembang selaras dengan perkembangan teknologi, di mana produksi memiliki suatu jalinan hubungan

timbal-balik (dua arah) yang sangat erat dengan teknologi. Produksi dan teknologi saling membutuhkan. Kebutuhan produksi untuk beroperasi dengan biaya yang lebih rendah, meningkatkan kualitas dan produktivitas, dan menciptakan produk baru telah menjadi kekuatan yang mendorong teknologi untuk melakukan berbagai terobosan dan penemuan baru. Produksi dalam sebuah organisasi pabrik merupakan inti yang paling dalam, spesifik serta berbeda dengan bidang fungsional lain seperti keuangan, personalia, dan lain-lain. (Jurnal Ekonomi Vol. 4 No. 1 : 2015: 3)

II.3. Sistem Produksi

Sistem produksi merupakan sistem integral yang mempunyai komponen struktural dan fungsional. Dalam sistem produksi modern terjadi suatu proses transformasi nilai tambah yang mengubah input menjadi output yang dapat dijual dengan harga kompetitif di pasar. Proses transformasi nilai tambah dari input menjadi output dalam system produksi modern selalu melibatkan komponen struktural dan fungsional. Sistem produksi memiliki beberapa karakteristik berikut:

1. Mempunyai komponen-komponen atau elemen-elemen yang saling berkaitan satu sama lain dan membentuk satu kesatuan yang utuh. Hal ini berkaitan dengan komponen struktural yang membangun sistem produksi itu.
2. Mempunyai tujuan yang mendasari keberadaannya, yaitu menghasilkan produk (barang atau jasa) berkualitas yang dapat dijual dengan harga kompetitif di pasar.

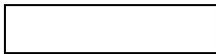

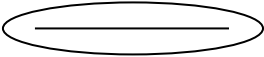
3. Mempunyai aktivitas berupa proses transformasi nilai tambah input menjadi output secara efektif dan efisien.

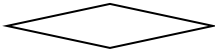
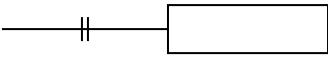
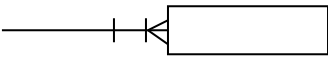
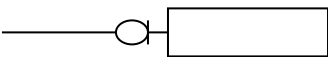
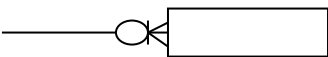
Sistem produksi memiliki komponen atau elemen struktural dan fungsional yang berperan penting dalam menunjang kontinuitas operasional sistem produksi itu. Komponen atau elemen struktural yang membentuk sistem produksi terdiri dari: bahan (material), mesin dan peralatan, tenaga kerja, modal, energi, informasi, tanah, dan lainlain. Sedangkan komponen atau elemen fungsional terdiri dari: supervisi, perencanaan, pengendalian, koordinasi, dan kepemimpinan, yang kesemuanya berkaitan dengan manajemen dan organisasi. Suatu sistem produksi selalu berada dalam lingkungan, sehingga aspek-aspek lingkungan, seperti perkembangan teknologi, sosial dan ekonomi, serta kebijakan pemerintah akan sangat mempengaruhi keberadaan system produksi itu. (Jurnal Ekonomi Vol. 4 No. 1 : 2015: 3)

II.4. *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) adalah bagian yang menunjukkan hubungan antara entity yang ada dalam sistem. Simbol-simbol yang digunakan dapat dilihat dari tabel II.1. (Jurnal Momentum Vol 17 No.2 : 2015: Hal : 70).

Tabel II.1. Simbol Yang Digunakan Pada *Entity Relationship Diagram (ERD)*

SIMBOL	KETERANGAN
	<i>Entity</i>
	Atribut Dan <i>Entity</i>
	Atribut Dan <i>Entity</i> Dengan <i>Key</i> (Kunci)

	Relasi Atau Aktifitas Antar <i>Entity</i>
	Hubungan Satu Dan Pasti
	Hubungan Banyak Dan Pasti
	Hubungan Satu Tapi Tidak Pasti
	Hubungan Banyak Tapi Tidak Pasti

(Sumber : Jurnal Momentum Vol 17 No.2 : 2015: Hal : 70)

II.5. Kamus Data

Kamus Data (KD) adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data selain digunakan untuk dokumentasi dan mengurangi redundansi, juga dapat digunakan untuk:

1. Memvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan
3. Menentukan muatan data yang disimpan dalam file-file
4. Mengembangkan logika untuk proses-proses diagram aliran data

KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis sistem, KD dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, KD digunakan untuk merancang input, merancang laporan-laporan dan

database. Kamus data dibuat berdasarkan arus data yang ada di DFD. KD mendefinisikan elemen data dengan fungsi sebagai berikut:

1. Menjelaskan arti aliran data dan penyimpanan data dalam DFD
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran (misalnya alamat diuraikan menjadi kota, negara dan kode pos)
3. Mendeskripsikan komposisi penyimpanan data
4. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran.
5. Mendeskripsikan hubungan detail antar penyimpanan (yang akan menjadi titik perhatian dalam *entity-relationship diagram*). (Jurnal : Jurusan teknik elektro politeknik negeri lhokseumawe : 2010)

II.6. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional

1. Bentuk Normal Pertama (1 NF)

Contoh yang kita gunakan di sini adalah sebuah perancangan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri. Masing-masing pemasok

bisa menyediakan banyak barang. Tabel relasionalnya dapat dituliskan sebagai berikut :

PEMASOK (P#, Status, Kota, b#, qty) di mana

p# : kode pemasok (kunci utama)

status : kode status kota

Kota : nama kota

b# : barang yang dipasok

qty : jumlah barang yang dipasok.

Sebuah tabel relasional secara defenisi selalu berada dalam bentuk normal pertama. Semua nilai pada kolom-kolomnya adalah atomi. Ini berarti kolom-kolom tidak mempunyai nilai berulang. Tabel II.2. menunjukkan tabel pemasok dalam 1 NF.

Tabel II.2. Normalisasi Pertama Pemasok

P#	Status	Kota	B#	Qty
P1	20	Yogyakarta	B1	300
P1	20	Yogyakarta	B2	200
P1	20	Yogyakarta	B3	400
P1	20	Yogyakarta	B4	200
P1	20	Yogyakarta	B5	100
P1	20	Yogyakarta	B6	100
P2	10	Medan	B1	300
P2	10	Medan	B2	400
P3	10	Medan	B2	200
P4	20	Yogyakarta	B2	200
P4	20	Yogyakarta	B4	300
P4	20	Yogyakarta	B5	400

Sumber : (Janner Simarmata, dkk, 2010).

2. Bentuk Normal Kedua (2 NF).

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1 NF, tetapi tidak pada 2 NF, sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1 NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama. Tabel pemasok berada pada 1 NF, tetapi tidak pada 2 NF karena status dan kota tergantung secara fungsional hanya pada kolom p# dari kunci gabungan (p#, b#). Ini dapat digambarkan dengan membuat daftar ketergantungan fungsional.

P# \longrightarrow Kota, Status

Kota \longrightarrow Status

(P#, B#) \longrightarrow qty

Proses mengubah tabel 1 NF ke 2 NF adalah :

- a. Tentukan sembarang kolom penentu selain kunci gabungan dan kolom-kolom yang ditentukannya.
- b. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom-kolom yang ditentukan.
- c. Pindahkan kolom-kolom yang ditentukan dari tabel asal ke tabel baru penentu akan menjadi kunci utama pada tabel baru
- d. Hapus kolom yang baru dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.

e. Tabel asal bisa diberi nama baru.

Pada contoh, kita memindahkan kolom p#, status, dan kota ke tabel baru yang disebut pemasok2. Kolom p# menjadi kunci utama tabel ini. Tabel II.3. menunjukkan hasilnya.

Tabel II.3. Tabel Bentuk Normal Kedua (2NF).

Pemasok2			Barang		
P#	Status	Kota	P#	B#	Qty
P1	20	Yogyakarta	P1	B1	300
P2	10	Medan	P1	B2	200
P3	10	Medan	P1	B3	400
P4	20	Yogyakarta	P1	B4	200
P5	30	Bandung	P1	B5	100
			P1	B6	100
			P2	B1	300
			P2	B2	400
			P3	B2	200
			P4	B2	200
			P4	B4	300
			P4	B5	400

Sumber : (Janner Simarmata, dkk, 2010).

3. Bentuk Normal Ketiga (3 NF).

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional hanya pada kunci utama. Secara defenisi, sebuah tabel berada pada bentuk normal ketiga (3 NF) jika tabel sudah berada pada 2 NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama. Tabel barang sudah dalam bentuk normal ketiga. Kolom bukan kunci, qty, tergantung sepenuhnya pada kunci utama (p#, b#). Pemasok masih berada pada 2 NF, tetapi belum berada pada 3 NF karena dia mengandung ketergantungan transitif. Ketergantungan transitif terjadi ketika sebuah kolom bukan kunci, yang

ditentukan oleh kunci utama, menentukan kolom lainnya. Konsep ketergantungan transistif dapat digambarkan dengan menunjukkan ketergantungan fungsional pada pemasok2, yaitu :

Pemasok2. p# \longrightarrow Pemasok2, status

Pemasok2. p# \longrightarrow Pemasok2, kota

Pemasok2. kota \longrightarrow Pemasok2, status

Perlu dicatat bahwa pemasok2, status ditentukan, baik oleh kunci utama p#, maupun kolom bukan kunci, kota

Proses mengubah tabel menjadi 3 NF adalah :

- a. Tentukan semua penentu selain kunci utama dan kolom yang ditentukannya.
- b. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom yang ditentukannya.
- c. Pindahkan kolom yang ditentukan dari tabel asal ke tabel baru. Penentu menjadi kunci utama tabel baru.
- d. Hapus kolom yang baru saja dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.
- e. Tabel asal bisa diberi nama baru.

Untuk mengubah PEMASOK2 menjadi 3 NF, kita membuat tabel baru yang disebut KOTA_STATUS dan memindahkan kolom kota dan status ke tabel baru. Status dihapus dari tabel diberi nama baru PEMASOK_KOTA. Tabel II.4 menunjukkan hasilnya.

Tabel II.4. Tabel Bentuk Normal Ketiga (3 NF)

PEMASOK_KOTA

P#	Kota
P1	Yogyakarta
P2	Medan
P3	Medan
P4	Yogyakarta
P5	Bandung

KOTA_STATUS

Kota	Status
Yogyakarta	20
Medan	10
Bandung	30
Semarang	40

Sumber : (Janner Simarmata, dkk, 2010).

4. Bentuk Normal Boyce Code (BCNF)

Setelah 3 NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3 NF sudah cukup karena sangat jarang entitas yang berada pada 3 NF bukan merupakan 4 NF dan 5 NF. Lebih lanjut, mereka berpendapat bahwa keuntungan yang didapat mengubah entitas ke 4 NF dan 5 NF sangat kecil sehingga tidak perlu dikerjakan. Bentuk Normal Boyce- Code (BCNF) adalah versi 3 NF lebih teliti dan berhubungan dengan tabel relasional yang mempunyai (a) banyak kunci kandidat (b) kunci kandidat gabungan, dan (c) kunci kandidat yang saling tumpang tindih.

BCNF didasarkan pada konsep penentu. Sebuah kolom penentu adalah kolom di mana kolom-kolom lain sepenuhnya tergantung secara fungsional. Sebuah tabel relasional berada pada BCNF jika dan hanya setiap penentu adalah kunci kandidat

5. Bentuk Normal Keempat (4 NF)

Sebuah tabel relasional berada pada bentuk normal keempat (4 NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional.

Bentuk normal keempat (4 NF) didasarkan pada konsep ketergantungan multivalued (MVD). Sebuah ketergantungan multivalued terjadi ketika dalam sebuah tabel relasional yang mengandung setidaknya tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda.

Defenisi secara formal diberikan oleh CJ. Date, yaitu :

Misalnya, ada sebuah tabel relasional R dengan kolom A, B dan C, Maka R.A R.B \longrightarrow (kolom A menentukan kolom B).

Adalah benar jika dan hanya jika himpunan nilai B yang cocok dengan pasangan nilai A dan nilai C pada R hanya tergantung pada nilai A dan tidak tergantung pada nilai C.

MVD selalu terjadi dalam pasangan, yaitu R.A \longrightarrow R.B dipenuhi jika dan hanya jika R.A \longrightarrow R.C dipenuhi pula.

6. Bentuk Normal Kelima (5 NF).

Sebuah tabel berada pada bentuk normal kelima jika dia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil.

Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*). Ketergantungan gabungan berarti sebuah tabel, setelah deskomposisi menjadi tiga atau lebih tabel yang lebih kecil,

harus dapat digabungkan kembali untuk membentuk tabel asal. Dengan kata lain 5 NF menunjukkan ketika sebuah tabel tidak dapat dideskomposisi lagi (Janner Simarmata, 2012).

II.7. Basis Data (Database)

Menurut Edhy Sutanta (2011), "Database atau basis data sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer". Secara harfiah pengertian basis merupakan dasar, ataupun gudang sedangkan data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. Database sering digunakan untuk melakukan proses terhadap data-data tersebut dan menghasilkan informasi tertentu.

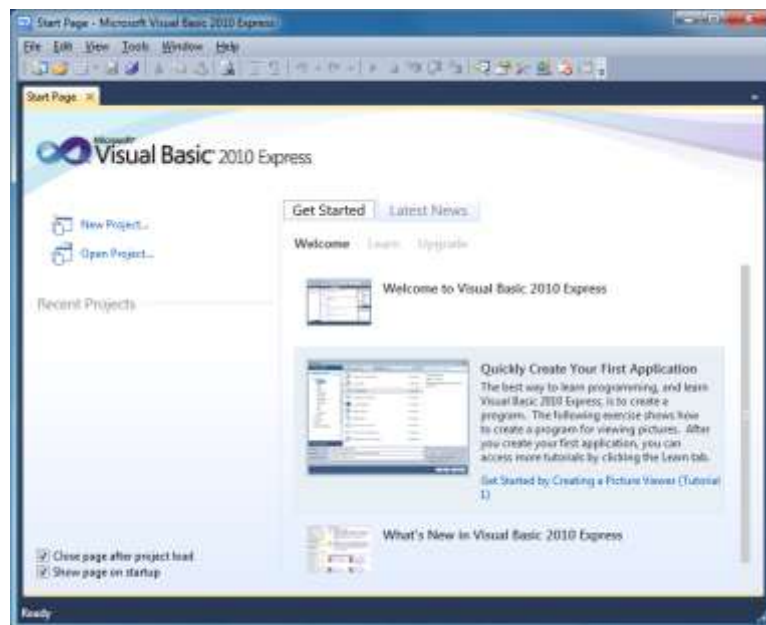
Berdasarkan beberapa pengertian diatas dapat disimpulkan arti dari basis data merupakan kumpulan atau himpunan dari data-data dan informasi yang saling terintegrasi, terorganisir dan terhubung satu sama lain tersimpan pada tempat yang sama tanpa adanya kerangkapan data atau informasi, dan pada saat dibutuhkan dapat dipanggil serta difungsikan kembali untuk kepentingan para penggunanya (*user*).

Menurut Edhy Sutanta (2011) , "Sebuah sistem database adalah sekumpulan dari komponen-komponen database-database yang meliputi:

1. Database
2. Database server
3. Komponen Client software

II.8. Visual Studio 2010

Menurut A.H. Hirin (2011) *visual Studio* merupakan salah satu bahasa pemrograman berbasis dekstop yang dikeluarkan (diproduksi) oleh perusahaan perangkat lunak komputer yaitu *microsoft*. Visual Basic memiliki berbagai hal yang patut dijadikan alasan untk diminati para programmer, selain bahasa pemrograman yang sangat (paling) mudah dipelajari oleh berbagai kalangan baik awam maupun ahli, Visual Basic yang didukung penuh oleh produsennya (*Microsoft*) selalu dikembangkan dan disesuaikan dengan kebutuhan zaman seperti penyesuaian model pemrograman modern yang berbasis OOP (*Object Oriented Programming*)



Gambar II.1. Tampilan Utama Visual Basic 2010

Sumber : (A. M. Hirin, 2011)

II.9. Mysql

MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan multi user. *MySQL* memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. Penulis sendiri dalam menjelaskan buku ini menggunakan *MySQL* yang *free software* karena bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/ GPL (*General Public License*), yang dapat anda download pada alamat resminya (Yuniar Supardi, 2007)

II.10. UML (*Unified Modeling Language*)

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industry perangkat lunak dan pengembangan sistem. (Windu Gata : 2013)

II.10.1. Diagram-Diagram UML

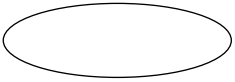
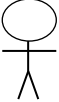



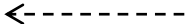
1. *Diagram Use Case (Use Case Diagram)*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat

dikatakan use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Simbol-simbol yang digunakan dalam use case diagram, yaitu :

Tabel 5. Simbol Use Case Diagram




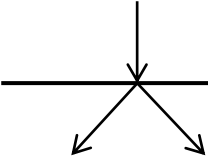
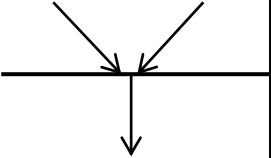
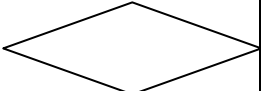

Gambar	Keterangan
	Use case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama use case.
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan use case, tetapi tidak memiliki control terhadap use case.
	Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam use case lain (<i>required</i>) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.

Sumber : (Windu Gata : 2013)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel 6. Simbol Diagram Aktivitas

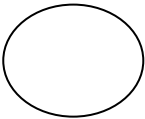
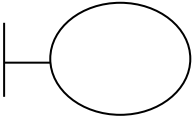
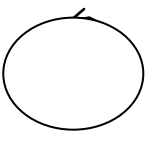

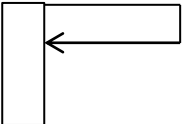

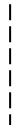
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

Sumber : (Windu Gata : 2014)

3. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel 7. Simbol Sequence Diagram

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

Sumber : (Windu Gata : 2013)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class diagram secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel 8. Simbol Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

Sumber : (Windu Gata : 2013)