

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem merupakan suatu kumpulan dari komponen-komponen yang membentuk satu kesatuan. Sebuah organisasi dan sistem informasi adalah sistem fisik dan sosial yang ditata sedemikian rupa untuk mencapai tujuan tertentu. Seorang manajer yang mengorganisasikan pekerjaan unitnya adalah orang yang menciptakan sesuatu sistem pengembangan personal *computer* (PC) menciptakan suatu sistem yang terdiri dari perangkat keras dan lunak PC ini menjadi subsistem dari suatu perangkat konferensi elektronik. Sistem fisik dan sosial adalah sistem yang abstrak (*abstract system*) dari konsep dan ide. Contoh sistem yang abstrak ialah pengembangan daur hidup atau pengembangan sistem perangkat lunak (Tyoso ; 2016 : 1).

II.2. Informasi

Informasi adalah data yang berguna yang telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Informasi sangat penting bagi organisasi. Pada dasarnya informasi adalah penting seperti sumber daya yang lain, misalnya peralatan, bahan, tenaga, dan sebagainya. Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi. Dalam sistem informasi akuntansi, kualitas dari informasi yang disediakan merupakan hal penting dalam kesuksesan sistem. Secara konseptual seluruh sistem

organisasional mencapai tujuannya melalui proses alokasi sumberdaya, yang diwujudkan melalui proses pengambilan keputusan manajerial. Informasi memiliki nilai ekonomik pada saat ia mendukung keputusan alokasi sumberdaya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan (Agustinus Mujilan : 2012 : 1).

II.3. Sistem Informasi

Sistem informasi berbasis komputer merupakan sekelompok perangkat keras dan perangkat lunak yang dirancang untuk mengubah data menjadi informasi yang bermanfaat. Jenis sistem informasi berbasis komputer :

1. Pengolahan Data. Pengolahan data elektronik – *electronic data processing (EDP)* adalah pemanfaatan teknologi komputer untuk melakukan pengolahan data transaksi-transaksi dalam suatu organisasi. EDP adalah aplikasi sistem informasi akuntansi paling dasar dalam setiap organisasi. Sehubungan dengan perkembangan teknologi komputer, istilah pengolahan data mulai dikenal dan mempunyai arti yang sama dengan istilah EDP.
2. Sistem Informasi Manajemen (SIM), menguraikan penggunaan teknologi komputer untuk menyediakan informasi bagi pengambilan keputusan para manajer.
3. Sistem Pendukung Keputusan – *Decision Support Systems (DSS)*. DSS diarahkan untuk melayani permintaan informasi tertentu, khusus, dan tidak rutin dari manajemen.

4. Sistem Pakar – *expert systems* (ES) adalah sistem informasi berbasis pengetahuan yang memanfaatkan pengetahuannya tentang bidang aplikasi tertentu untuk bertindak seperti seorang konsultan ahli bagi pemakainya.
5. Sistem Informasi Eksekutif – *executive information systems* (EIS). EIS dibuat bagi kebutuhan informasi strategik manajemen tingkat puncak.
6. Sistem Informasi Akuntansi – sistem berbasis komputer yang dirancang untuk mengubah data akuntansi menjadi informasi (Agustinus Mujilan : 2012).

II.4. Akuntansi

Akuntansi berasal dari bahasa Inggris yaitu “*To account*” yang artinya menghitung atau mempertanggungjawabkan sesuatu yang ada kaitannya dengan pengelolaan bidang keuangan dari suatu perusahaan kepada pemiliknya atas kepercayaan yang telah diberikan kepada pengelola tersebut untuk menjalankan kegiatan perusahaan. Pengertian lain akuntansi merupakan kumpulan prosedur berupa kegiatan mencatat, mengikhtisarkan, mengklasifikasikan dan melaporkan keuangan dalam bentuk laporan keuangan dalam periode waktu. Laporan keuangan yang dihasilkan harus dapat di pertanggungjawabkan kepada pihak-pihak yang berkepentingan. Akuntansi adalah proses dari transaksi yang dibuktikan dengan faktur, lalu dari transaksi dibuat jurnal, buku besar, neraca lajur kemudian akan menghasilkan Informasi dalam bentuk laporan keuangan yang digunakan pihak-pihak tertentu (Wiratna Sujarweni : 2016 : 1).

II.5. Sistem Informasi Akuntansi

Sistem informasi akuntansi adalah kumpulan sumberdaya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. Sistem Informasi Akuntansi mewujudkan perubahan ini secara manual atau terkomputerisasi. Sistem Informasi Akuntansi juga merupakan sistem yang paling penting di organisasi dan merubah cara menangkap, memproses, menyimpan, dan mendistribusikan informasi. Saat ini, digital dan informasi *online* semakin digunakan dalam sistem informasi akuntansi. Organisasi perlu menempatkan sistem di lini depan, dan mempertimbangkan baik segi sistem ataupun manusia sebagai faktor yang terkait ketika mengatur sistem informasi akuntansi. Sistem Informasi Akuntansi pada umumnya meliputi beberapa siklus pemrosesan transaksi :

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaran-pembayaran yang berkaitan.
2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.
3. Siklus produksi. Berkaitan dengan pengubahan sumber daya menjadi barang dan jasa.
4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas (Agustinus Mujilan : 2012 : 3).

II.6. Persediaan

Persediaan barang merupakan banyaknya barang yang digunakan untuk memfasilitasi produksi atau untuk memuaskan permintaan konsumen. Persediaan barang meliputi barang yang dibeli dan disimpan untuk dijual kembali. Dalam hal ini, untuk memenuhi persediaan barang sebuah perusahaan dagang melakukan pembelian dan selanjutnya barang yang dibeli akan dijual kembali kepada konsumen. Persediaan berpengaruh terhadap neraca dan laporan Laba/ Rugi dalam neraca perusahaan dagang atau perusahaan industri, persediaan seringkali merupakan bagian yang sangat besar dari keseluruhan Aktiva lancar yang dimiliki perusahaan ini merupakan bukti betapa pentingnya persediaan barang dagangan untuk kegiatan pembelian dan penjualan dalam operasi perusahaan semacam itu. Pencatatan persediaan yang efektif sering kali merupakan kunci keberhasilan operasi perusahaan di samping pimpinan perusahaan berusaha untuk mempertahankan kuantitas dan jenis persediaan yang cukup untuk memenuhi permintaan konsumen, pimpinan juga harus menjaga keseimbangan persediaan agar tidak terlalu tinggi dan juga tidak terlalu rendah. Persediaan yang terlalu kecil akan menimbulkan kekecewaan konsumen sebaliknya persediaan yang terlalu tinggi akan menyebabkan biaya penyimpanan dan pemeliharaan persediaan akan membengkak (Ruslan ; 2013 : 38).

II.7. Metode *Simple Average*

Metode biaya rata-rata menghitung harga pos-pos yang terdapat dalam persediaan atas dasar biaya rata-rata yang sama tersedia selama satu periode.

Pemakaian metode rata-rata biasanya dapat dibenarkan dari sisi praktis, bukan karena alasan konseptual. Metode ini mudah diterapkan, objektif dan tidak dapat dimanfaatkan untuk manipulasi laba seperti halnya beberapa metode penentuan harga persediaan bahwa secara umum perusahaan tidak mungkin mengukur arus fisik persediaan secara khusus, dan karenanya lebih baik menghitung biaya persediaan atas dasar harga rata-rata (Maulinda Octaviani ; 2012 : 5).

Harga Pokok Penjualan.

$$\text{HPP per Unit} = \frac{[\text{Rp Saldo awal} + \text{Rp Pembelian}]}{[\text{Qty saldo awal} + \text{Qty pembelian}]} \dots\dots\dots (\text{II.1})$$

$$\text{Total HPP yang terjual} = \text{HPP per Unit} \times \text{Qty Terjual} \dots\dots\dots (\text{II.2})$$

$$\text{Saldo Akhir} = \text{Saldo Awal} + \text{Pembelian} - \text{Penjualan} \dots\dots\dots (\text{II.3})$$

II.7.1. Kartu Persediaan

Kartu persediaan digunakan untuk mencatat transaksi setiap jenis persediaan, memuat nama barang, tempat penyimpanan barang, kode barang dan kolom-kolom yang dipakai untuk mencatat transaksi adalah tanggal, pembelian (pemasukan), penjualan (pengeluaran) dan sisa atau saldo persediaan. Berikut contoh kartu persediaan yang terlihat pada tabel II.1.

Tabel II.1. Kartu Persediaan

Nama Perusahaan :										
Jenis Barang :										
Kode Barang:										
Gudang :										
Tgl	Pembelian			Penjualan			Saldo			
	Unit	Harga	Jumlah	Unit	Harga	Jumlah	Unit	Harga	Jumlah	
	Persediaan Akhir Harga Pokok Penjualan (HPP)									

(Sumber : Riswan ; 2005 : 108)

II.8. Normalisasi

Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal.

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional*. Dengan normalisasi kita ingin mendesain *database relasional* yang terdiri dari tabel-tabel berikut :

1. Berisi data yang diperlukan.
2. Memiliki sedikit mungkin redundansi.
3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.
4. Mengefisienkan *update*.

5. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal (Kusrini : 2012).

II.8.1. Bentuk-bentuk Normalisasi

1. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

Tabel II.2. Tabel Tidak Normal

Kode	Kode Suplier	Nama Suplier	Kode Brg	Nama brg	Tgl
004	G01	Gobel Nustra	A01 A02	AC Model 1 AC Model 2	07/10/04
006	A03	Angkasa	B01 A03	Meja, Kursi	09/10/05

2. Bentuk normal tahap pertama (1st Normal Form)

Definisi :

Sebuah tabel disebut 1NF jika :

- a. Tidak ada baris yang duplikat dalam tabel tersebut.
- b. Masing-masing *cell* bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

Tabel II.3. Tabel Normal 1

Kode	Kode Suplier	Nama Suplier	Kode Brg	Nama brg	Tgl
004	G01	Gobel Nustra	A01	AC Model 1	07/10/04
004	G01	Gobel Nustra	A02	AC Model 2	07/10/04
006	A03	Angkasa	B01	Meja	09/10/05
006	A03	Angkasa	A03	Kursi	09/10/05

3. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh.

KdFaktur → Tgl. Jtempo.

KodeSup. NamaSup

KdFaktur, Kodebrg → namaBrg, Qty, Harga

4. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

a. X haruslah *superkey* pada tabel tersebut.

b. Atau A merupakan bagian dari *primary key* pada tabel tersebut.

5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

6. Boyce Code Normal Form (BCNF)

- a. Memenuhi 1st NF
- b. Relasi harus bergantung fungsi pada atribut *superkey* [14]

KdFaktur → Tgl, Jtempo, KodeSup

KdSup → NamaSup

KdFaktur, Kodebrg → Qty, Harga

KodeBrg → NamaBrg (Kusrini : 2012)

II.9. Visual Studio 2010

Visual Basic 2010 adalah inkarnasi dari bahasa *Visual Basic* yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa tingkat lainnya seperti C++. Anda dapat menggunakan *visual basic* 2010 untuk membuat aplikasi *Windows*, *mobile*, *web*, dan *office* atau kode yang telah ditulis oleh orang lain dan kemudian dimasukkan ke program lainnya. *Visual Basic* menyediakan berbagai *tools* dan fitur canggih yang memungkinkan dapat menulis

kode, menguji dan menjalankan program tunggal atau terkadang serangkaian program yang terkait dengan satu aplikasi (Christopher Lee:2014:1)

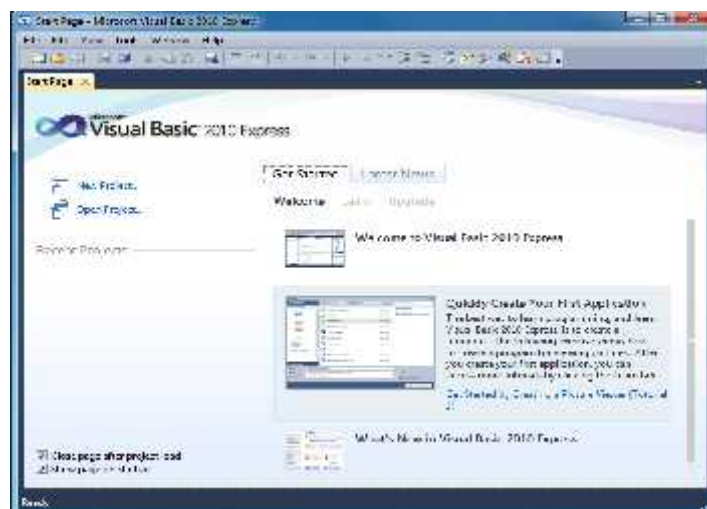
Tabel II.4. Jendela Aplikasi *Visual Studio* 2010

No	Bagian	Keterangan
1	<i>Title Bar</i>	Menampilkan nama aplikasi yang sedang terbuka.
2	<i>Menu Bar</i>	Menampilkan daftar perintah yang memungkinkan anda dapat menulis, mengedit, menyimpan, mencetak, menguji dan menjalankan program <i>Visual Basic</i> .
3	<i>Standard Toolbar</i>	Berisi Tombol yang menjalankan perintah yang sering digunakan seperti <i>open project, new project, save, cut, copy, paste, dan undo</i>
4	<i>Toolbox</i>	Berisi <i>komponen. NET</i> yang dapat Anda gunakan untuk mengembangkan antarmuka pengguna grafis untuk program <i>Visual Studio</i> 2010.
5	Area Kerja Utama	Menampilkan item yang sedang dikerjakan
6	<i>Solution Explorer</i>	Menampilkan elemen dari <i>Visual Basic solution</i> , yaitu nama yang diberikan kepada program <i>Visual Basic</i> dan item lainnya yang dihasilkan oleh <i>Visual Basic</i> 2010 sehingga program akan mengeksekusi dengan benar.
7	<i>Properties Window</i>	Setiap Objek dalam program <i>Visual Basic</i> memiliki seperangkat karakteristik yang disebut sifat-sifat objek.

(Sumber : Christopher Lee ; 2014 : 2)

Untuk melihat tampilan *Visual Basic* 2010 dapat dilihat pada gambar II.1.

sebagai berikut :



**Gambar II.1. Tampilan Utama Visual Basic 2010
(Sumber : Christopher Lee ; 2014 ; 2)**

II.10. SQL Server 2008

SQL (*Structured Query Language*) adalah bahasa non *procedural* untuk mengakses data pada *database* relasional. SQL adalah bahasa *database* yang dipergunakan dalam menyelesaikan permasalahan dalam *database* serta mempunyai kelebihan dalam mengolah data. Standar SQL mula-mula didefinisikan oleh ISO (*International Standards Organization*) dan ANSI (*the American National Standards Institute*) yang dikenal dengan sebutan SQL86. Dengan menggunakan SQL, kita dapat melakukan hal-hal berikut:

1. Memodifikasi struktur *database*.
2. Mengubah, mengisi, menghapus isi *database*.
3. Mentransfer data antara *database* yang berbeda.

SQL ada yang dikembangkan untuk PC dan ada juga yang dikembangkan untuk dapat mengakomodasi *database* yang sangat besar. Beberapa contohnya antara lain:

1. *Microsoft Access*

Digunakan untuk PC, sangat mudah dipakai dimana perintah SQL dapat langsung dimasukkan atau melalui fasilitas yang telah digunakan.

2. *Microsoft Query*

SQL yang dipaket dengan produk lain dari *Microsoft Windows*, yaitu *Microsoft Visual Studio* seperti *Visual Basic* dan *Visual C++*. Untuk terhubung dengan *database* lain menggunakan *ODBC*.

3. *Oracle*

Digunakan untuk perusahaan yang menggunakan database besar (Eka Iswandy; 2015 : 73)

II.11. *UML (Unified Modelling Language)*

Menurut Windu Gata (2013) 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*

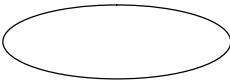
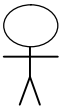

saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem (Gellysa Urva dan Helmi Fauzi Siregar, 2015, Hal : 93).

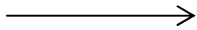
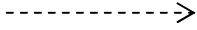
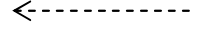
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis *UML* adalah sebagai berikut:

1. *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 dapat dilihat pada tabel II.5 di bawah ini:

Tabel II.5. Simbol *Use Case*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>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 <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>



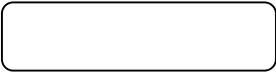
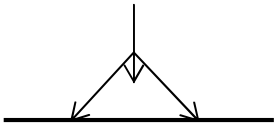
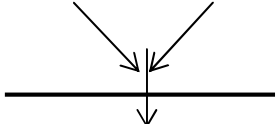
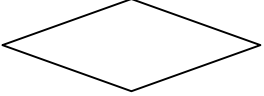
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015, Hal : 94)

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* dapat dilihat pada tabel II.6 :

Tabel II.6. Simbol *Activity Diagram*

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 paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .

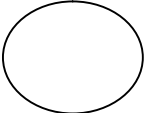
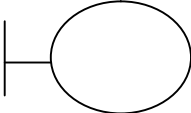
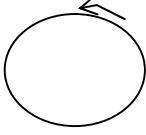

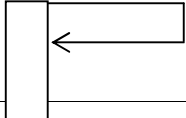
New Swimline	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.
--------------	--



(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015, Hal : 94)

3. Diagram Urutan (*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* dapat dilihat pada tabel II.7:

Tabel II.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 <i>formentry</i> 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, 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 : Gellysa Urva dan Helmi Fauzi Siregar; 2015, Hal : 95)

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 yang dapat dilihat pada tabel II.8 di bawah ini :

Tabel II.8. *Multiplicity 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 : Gellysa Urva dan Helmi Fauzi Siregar; 2015, Hal : 95)