

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pengertian Sistem**

Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. Karakteristik sistem terdiri dari :

##### 1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

##### 2. Batasan Sistem

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

##### 3. Lingkungan Luar Sistem

Lingkungan luar 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.

#### 4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

#### 5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

#### 6. Keluaran Sistem

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

#### 7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

#### 8. Sasaran Sistem

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya (Sulindawati: 2010 ; 375).

### **II.1.1. Informasi**

Informasi adalah data yang diolah menjadi suatu bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan 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.

Kualiatas dari suatu informasi tiga hal yaitu ;

1. Akurat, berarti informasi harus bebas dari kesalahan – kesalahan dan tidak bias atau 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.
  2. Tepat pada waktunya, berarti informasi yang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi.
  3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap – tiap orang satu dengan lainnya berbeda.
- (Sulindawati : 2010 ; 3)

### **II.1.2. Sistem Informasi**

Sistem informasi dapat diartikan sebagai suatu sistem di dalam organisasi yang merupakan kombinasi dari orang – orang, fasilitas, teknologi, media, prosedur – prosedur, dan pengendalian yang ditujukan untuk mendapatkan jalur kombinasi yang penting. (Sulindawati : 2010 ; 4)

Dalam suatu sistem informasi terdapat komponen - komponen sebagai berikut :

1. Perangkat keras (*hardware*), mencakup berbagai peranti fisik seperti komputer dan printer.
2. Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
4. Orang, yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.
5. Basis data (*database*), yaitu sekumpulan *tabel*, hubungan dan lain - lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resource*) dipakai secara bersama atau diakses oleh sejumlah pemakai. (Sulindawati : 2010 : 4)

## **II.2. Sistem Informasi Akuntansi**

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Lingkup sistem informasi akuntansi dapat dijelaskan dari manfaat yang didapat dari informasi akuntansi. Manfaat atau tujuan sistem informasi akuntansi tersebut adalah sebagai berikut :

1. Mengamankan harta / kekayaan perusahaan. Harta / kekayaan di sini meliputi kas perusahaan, persediaan barang dagangan, termasuk aset tetap perusahaan.

2. Menghasilkan beragam informasi untuk pengambilan keputusan. misal, pengelola toko swalayan memerlukan informasi mengenai barang apa saja yang diminati oleh konsumen. Membeli barang yang kurang laku berarti kas akan terjebak dalam persediaan dan berarti kehilangan kesempatan untuk membeli barang dagangan yang laku.
3. Menghasilkan informasi untuk pihak eksternal. Setiap pengelola usaha memiliki kewajiban untuk membayar pajak. Besarnya pajak yang dibayar tergantung pada omset penjualan (jika pengelola memilih menggunakan norma dalam perhitungan pajaknya) atau tergantung pada laba rugi usaha (jika pengelola memilih untuk tidak menggunakan norma dalam perhitungan pajaknya).
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi. Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan). Data yang tersimpan dengan baik sangat memudahkan proses audit (pemeriksaan).
6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan. Anggaran merupakan alat yang sering digunakan perusahaan untuk mengendalikan pengeluaran kas.
7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian. Selain berguna untuk membandingkan informasi yang berkaitan dengan anggaran dan biaya standar dengan kenyataan seperti yang telah dikemukakan (Anastasia Diana ; 2011 : 6).

### II.2.1. Kualitas Data Sistem Informasi Akuntansi

Seperti telah disebutkan di atas bahwa kualitas informasi merupakan hal yang sangat penting dalam sistem informasi. Informasi yang baik akan tergantung pula dengan data yang baik. Untuk membentuk sistem yang mampu menyediakan data dan informasi yang baik dan dapat dipercaya membutuhkan suatu usaha yang tidak mudah.

Menggaris bawahi hal penting dalam kaitannya dengan kualitas data.

1. Personil yang kompeten (competent personel) agar sistem sesuai atau cocok digunakan.
2. Kontrol input (input control) merupakan hal bagian dari kontrol yang penting, apalagi dalam kasus transaksi *online*.
3. Jika perlu sediakan manajer kualitas data yang bertanggung jawab pada kualitas data dalam sistem informasi akuntansi.

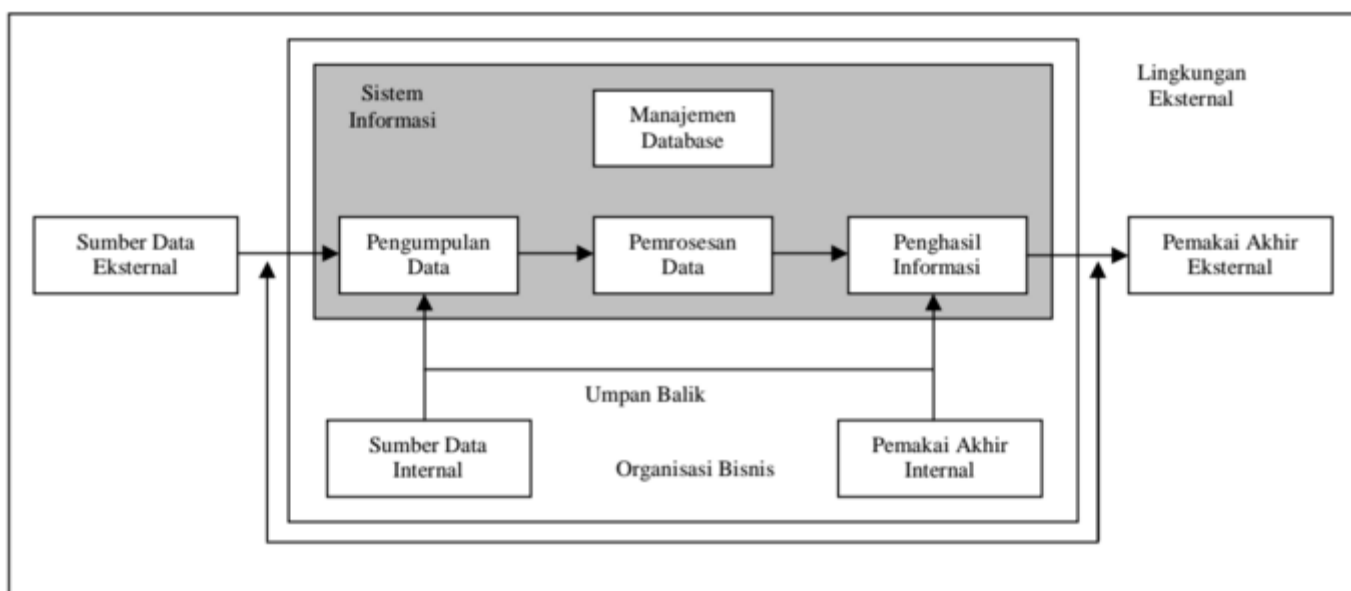
Berikut ini merupakan gambaran tentang pentingnya kualitas data dalam sistem informasi akuntansi dari hasil wawancara yang dilakukan oleh Xu (2009) pada para manajer.

*We have to monitor our cash balances fairly closely and it [data quality] is definitely one of the highest priorities. We have forecasts that need to be met, so we need to give ourselves early warning signals if a part of the business looks like it is not performing. The numbers will tell us that hopefully, so we can address the issue.*

Kami harus memonitor saldo kas secara jelas dan kualitas data merupakan salah satu prioritas penting. Kami memperkirakan kebutuhan itu untuk dipenuhi, sehingga kami perlu mendapat tanda peringatan awal jika salah satu bagian dari bagian bisnis tidak berjalan sebagaimana mestinya. Suatu nomor aka memberitahukannya, jadi kita dapat mengenali permasalahannya]. (Agustinus Mujilan : 2012 : 4).

## II.2.2. Arsitektur Sistem Informasi Akuntansi

Arsitektur sistem informasi akuntansi merupakan kerangka model umum yang menggambarkan semua sistem yang digunakan dalam pembelajaran SIA dari mulai sumber data, proses pengumpulan, pemrosesan dan penyajian hasil informasi baik di lingkungan internal maupun eksternal perusahaan, yang dapat kami paparkan sebagai berikut :



**Gambar II.1. Arsitektur Sistem Informasi Akuntansi**  
(Sumber : Ebook aris koerniawan 2012 : 8)

### **II.3. Produksi**

Organisasi industry merupakan salah satu mata rantai dari sistem perekonomian, karena ia memproduksi dan mendistribusikan produk (barang atau jasa). Produksi merupakan fungsi pokok dalam setiap organisasi, yang mencakup aktivitas yang bertanggung jawab untuk menciptakan nilai tambah produk yang merupakan output dari setiap organisasi industry itu.

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. (Leo Willyanto Santoso : 2011:1)

### **II.4. Full Costing**

Metode *full costing* merupakan metode penentuan harga pokok produksi yang memperhitungkan semua unsure biaya produksi ke dalam harga pokok produksi, yang terdiri atas biaya bahan baku, biaya tenaga kerja langsung dan biaya overhead pabrik, baik yang berperilaku variable maupun tetap. Dengan demikian, harga pokok produksi dengan full costing terdiri atas unsure harga

pokok produksi (biaya bahan baku, biaya tenaga kerja langsung, biaya overhead pabrik variable, dan biaya overhead pabrik tetap) ditambah dengan biaya nonproduksi (biaya pemasaran, biaya administrasi dan umum). (Sumber : Iman Firmansyah:2013:101)

Biaya penuh (*full cost*) dengan pendekatan *full costing* merupakan jumlah *full production costs* untuk memproduksi suatu produk dengan pendekatan *full costing*, biaya administrasi dan umum, dan total biaya pemasaran. Sedangkan *full production costs* dengan pendekatan *full costing* merupakan total biaya bahan baku, biaya tenaga kerja langsung, biaya overhead pabrik variabel, dan biaya overhead pabrik. (Islahuzzaman : 2011 : 28)

Harga pokok produksi menurut metode *full costing* ;

Biaya bahan baku		xx
Biaya tenaga kerja langsung	xx	
Biaya overhead pabrik tetap	xx	
Biaya overheaad pabrik variable	xx +	
Harga pokok produksi	<hr/>	xx

(Sumber Islahuzzaman : 2011 : 28)

## II.5. Biaya Produksi

Biaya produksi adalah biaya yang terjadi untuk mengolah bahan baku mejadi bahan jadi. Komponen biaya produksi digolongkan menjadi biaya produksi langsung dan biaya produksi tidak langsung.

- a. Biaya produksi langsung adalah biaya yang terjadi karena adanya sesuatu yang dibiayai. Biaya ini langsung diperhitungkan kedalam harga pokok produksi yang terdiri atas biaya-biaya berikut :
  1. Biaya bahan langsung, artinya semua bahan untuk membentuk sesuatu kesatuan yang tidak dapat dipisahkan dari barang jadi dan dapat langsung diperhitungkan dalam harga pokok produksi.
  2. Biaya Tenaga kerja langsung, artinya upah para ekerja yang secara langsung membuat produk dan jasanya dapat langsung diperhitungkan ke dalam harga produk.
- b. Biaya produksi tak langsung adalah biaya selain biaya bahan langsung dan biaya tenaga kerja langsung yang terjadi dipabrik. Biaya ini lazim disebut dengan biaya overhead pabrik (BOP), dikelompokkan menjadi biaya-biaya berikut :
  1. Biaya bahan penolong, bahan yang diperlukan untuk pembuatan produk dan penggunaannya relative kecil.
  2. Biaya tenaga kerja tak langsung, upah untuk tenaga kerja yang secara tidak langsung berhubungan dengan pembuatan produksi.
  3. Biaya produksi tak langsung lainnya, seperti biaya penyusutan mesin, asuransi, dan perlengkapan mesin. (Sumber : Iman Firmansyah:2013:31)

## **II.6. Model Jaringan**

### **1. Peer to Peer**

Model jaringan peer to peer (P2P) memungkinkan seorang user untuk membagi sumber daya yang ada pada komputernya, baik berupa data/ informasi, hardware, dan lain-lain serta mengakses sumber daya yang terdapat pada computer lain. Namun, model ini tidak dapat memiliki sebuah file server atau sumber daya yang terpusat, sehingga seluruh computer mempunyai kemampuan yang sama untuk memakai sumber daya yang tersedia di jaringan. Model ini disesain untuk jaringan berskala kecil dan menengah. (Macdons Madiun : 2015 : 4)

### **2. Client Server**

Model jaringan computer Client server memusatkan/ mesentralisasikan fungsi dan aplikasi kepada satu dua computer server. Sebuah server menjadi jantung dari keseluruhan sistem, memungkinkan untuk mengakses sumber daya dan menyediakan keamanan bagi client/ workstation selama masih terhubung dalam suatu jaringan computer.

Modal jaringan client server menghubungkan computer server dengan beberapa computer client/ workstation. Komputer server adalah komputer yang menyediakan fasilitas bagi computer lain yang terhubung dalam jaringan.

Sedangkan komputer client adalah komputer yang menggunakan fasilitas yang disediakan oleh computer server. Komputer server pada sebuah jaringan model client server disebut dedicated server karena computer yang digunakan hanya sebagai penyedia fasilitas unuk komputer client/ workstation. Computer

server tidak dapat berperan sebagai komputer client/ workstation begitu juga sebaliknya. (Macdons Madiun : 2015 : 4)

### **3. Topologi jaringan**

Topologi jaringan merupakan gambaran pola hubungan antara komponen-komponen jaringan, yang meliputi computer server, computer Client/ Workstation, Hub/ Switch, pengkabelan, dan komponen jaringan yang lain. Terdapat beberapa topologi jaringan yang dapat digunakan dengan kondisi di lapangan. Topologi jaringan dapat juga diartikan sebagai skema fisik jaringan yang saling terhubung satu sama lain. (Macdons Madiun : 2015 : 6)

#### **1. Topologi Bus**

Pada topologi Bus ini terdapat satu jalur umum yang berbentuk garis lurus. Ciri utama dari topologi ini adalah bahwa setiap sambungan saling bergantung, artinya apabila salah satu sambungan mengalami gangguan atau terputus, maka seluruh sambungan akan terputus dan susah untuk terhubung kembali. Meskipun topologi ini membentuk jaringan yang saling terhubung namun tidak membentuk jalur tertutup. Pada topologi bus menyediakan 1 jalur yang digunakan untuk komunikasi antar perangkat sehingga setiap perangkat harus bergantian dalam menggunakan jalur yang ada. Dalam berkomunikasi antar perangkat hanya ada 2 perangkat yang dapat saling berkomunikasi. (Macdons Madiun : 2015 : 6)

#### **2. Topologi Token Ring**

Topologi token ring memiliki ciri khusus yaitu berbentuk lingkaran seperti cincin. Semua workstation dan server dihubungkan menjadi satu

membentuk pola lingkaran, setiap workstation maupun server akan menerima data atau informasi yang disharing, jika alamat yang dimaksud sesuai dengan maka informasi atau data akan diterima, dan jika alamat yang tidak sesuai maka informasi tersebut akan dilewatkan ke computer yang lain. Kelebihan dari topologi ini adalah desain cincin dalam jaringannya yang sangat mudah dibuat dan sederhana, serta tidak membutuhkan banyak kabel konektor jaringan untuk menghubungkan. (Macdons Madiun : 2015 : 7)

### 3. Topologi Ring

Pada topologi ini, semua PC yang terhubung ke jaringan akan saling dikaitkan sehingga membentuk satu koneksi yang tidak terputus. Namun, sistem topologi ring ini memiliki kelemahan, yaitu apabila terdapat kabel konektor yang terputus atau mengalami gangguan pada dua arah sekaligus, maka seluruh koneksi yang mengarah ke server juga ikut berpengaruh. (Macdons Madiun : 2015 : 7)

### 4. Topologi Star

Topologi ini mempunyai bentuk fisik layaknya bintang, dimana setiap node dihubungkan ke pusat. Jadi, setiap transfer data selalui melalui pusat. Media transmisinya bersifat tertutup dan setiap client, sehingga apabila salah satu client mengalami kegagalan, maka client yang lain tetap bisa berkomunikasi/ terhubung dengan computer server. (Macdons Madiun : 2015 : 8)

## 5. Topologi Tree

Topologi tree (Pohon) didesain berdasarkan kombinasi antara topologi bus dan topologi ring. Hal ini dapat dicapai dengan mengintegrasikan HUB Multi jaringan star bersama-sama pada satu bus. Bayangkan sebuah diagram topologi tree dengan banyak jaringan star dengan HUB pusatnya. Kemudian gambarlah garis yang hanya dihubungkan HUB bersama ke titik pada satu line. HUB ini umumnya terhubung dengan satu HUB utama yang merupakan akar untuk seluruh jaringan data diantara semua computer. (Macdons Madiun : 2015 : 8)

## 6. Topologi Mesh

Topologi mesh adalah topologi yang menyerupai bentuk jaringan atau jala yang digunakan pada desain jaringan LAN. Topologi mesh menggunakan salah satu dari dua pengaturan koneksi adalah menggunakan mesh penuh maupun mesh parsial. Meskipun topologi mesh ini dapat diandalkan karena kemampuan interkoneksinya yang namun juga memiliki masalah redundansi (perulangan). (Macdons Madiun : 2015 : 9)

## II.7. *MYSQL*

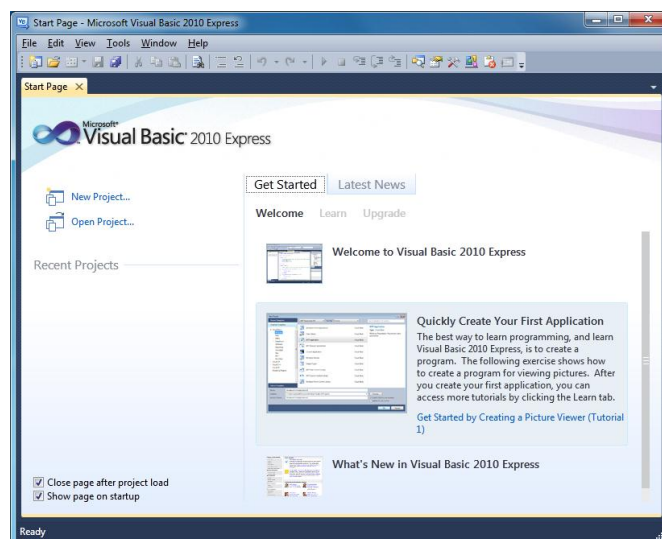
*MySql* adalah perangkat lunak basis data server yang terkenal dan bersifat open source dengan dukungan driver yang luas dari berbagai vendor. *MySql* adalah seakuntansi implementasi dari system manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*,

namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. (Yuniar Supardi, 2007).

## II.8. 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)

Untuk melihat tampilan visual basic 2010 dapat dilihat pada gambar II.2. sebagai berikut :



**Gambar II.2. Tampilan Utama Visual Basic 2010**

**(Sumber : Christopher Lee:2014:3)**

## II.9. Teknik 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.1. menunjukkan tabel pemasok dalam 1 NF.

**Tabel II.1. Normalisasi Pertama Pemasok**

<b>P#</b>	<b>Status</b>	<b>Kota</b>	<b>B#</b>	<b>Qty</b>
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, 2010:80)

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# → Kota, Status

Kota → Status

(P#, B#) → 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.2. menunjukkan hasilnya.

**Tabel II.2. Tabel Bentuk Normal Kedua**

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, 2010:82)

### 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 transistif 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 daopat 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.3 menunjukkan hasilnya.

**Tabel II.3. Tabel Bentuk Normal Ketiga**

PEMASOK_KOTA		KOTA_STATUS	
P#	Kota	Kota	Status
P1	Yogyakarta	Yogyakarta	20
P2	Medan	Medan	10
P3	Medan	Bandung	30
P4	Yogyakarta	Semarang	40
P5	Bandung		

**(Sumber : Janner Simarmata, dkk:2010:83)**

#### 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  $R.A \twoheadrightarrow R.B$  (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  $R.A \twoheadrightarrow R.B$  dipenuhi jika dan hanya jika  $R.A \twoheadrightarrow R.C$  dipenuhi pula. (Janner Simarmata:2012:85)

#### 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 dekomposisi 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:2012:86).

### II.10. UML (*Unified Modeling 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.

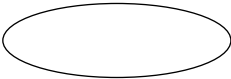
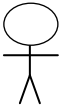
*UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Gellysa Urva dan Helmi Fauzi Siregar, 2015, : 93).


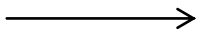
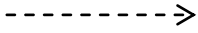
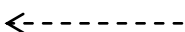
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis *UML* adalah sebagai berikut:

- *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.4 dibawah ini:

**Tabel II.4. 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>



	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.
	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.


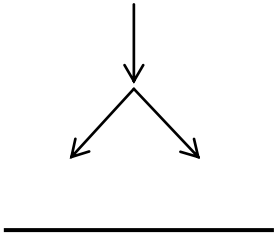
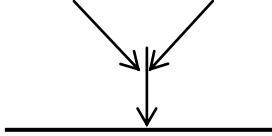
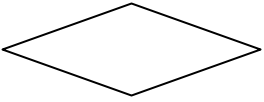

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

- 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.5 dibawah ini:

**Tabel II.5. Simbol *Activity Diagram***

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.

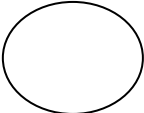
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

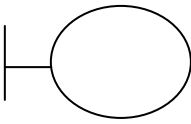
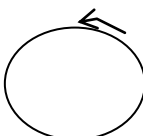
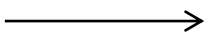
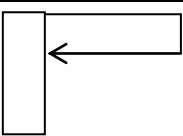


(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

- 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.6 dibawah ini :

**Tabel II.6. 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 : Gellysa Urva dan Helmi Fauzi Siregar:2015: 95)

- *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.7 dibawah ini :

**Tabel II.7. *Multiplicity Class Diagram***

<b>Multiplicity</b>	<b>Penjelasan</b>
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 : 95)