

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem pasti tersusun dari sub-sub sistem yang lebih kecil yang juga saling tergantung dan bekerja sama untuk mencapai tujuan (Anastasia Diana & Lilis Setiawati; 2011: 3).

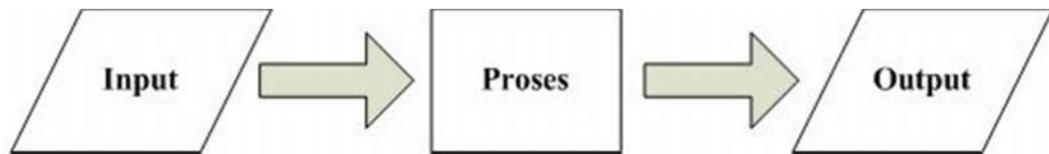
II.2. Pengertian Informasi

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, dsb. Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi (Xu, 2009). Dalam sistem informasi akuntansi, kualitas dari informasi yang disediakan merupakan hal penting dalam kesuksesan sistem. (Agustinus Mujilan; 2012: 1).

II.3. Pengertian Sistem Informasi

Input dalam sistem informasi adalah data-data yang relevan untuk menghasilkan informasi yang diinginkan. Proses adalah langkah-langkah yang perlu dilakukan untuk mengolah data menjadi informasi. Sedangkan output adalah

berupa informasi yang merupakan hasil dari pemrosesan data. (Anastasia Diana & Lilis Setiawati; 2011: 3).



Gambar II.1. Komponen Sistem Informasi
(Sumber : Anastasia Diana & Lilis Setiawati; 2011: 4)

II.4. Pengertian Sistem Informasi Akuntansi

Sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi. Misalnya, salah satu input dari sistem informasi pada sebuah toko baju adalah transaksi penjualan. (Anastasia Diana & Lilis Setiawati; 2011: 3).

Sistem informasi akuntansi adalah kumpulan sumber daya, 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.

SIA 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.

SIA pada umumnya meliputi beberapa siklus pemrosesan transaksi :

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayara-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.

Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas. (Agustinus Mujilan; 2012: 3).

II.4.1. Tujuan Sistem Informasi Akuntansi

Lingkup sistem informasi akuntansi dapat dijelaskan dari manfaat yg didapat dari informasi akuntansi. Manfaat atau tujuan sistem informasi akuntansi tersebut adalah sebagai berikut:

1. Mengamankan harta/kekayaan perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan.
3. Menghasilkan informasi untuk pihak internal.
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit(pemeriksaan).
6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan.
7. Menghasilkan informasi yg diperlukan dalam kegiatan perencanaan dan pengendalian. (Anastasia Diana & Lilis Setiawati; 2011; 5-6).

II.5. Pengertian Leasing

II.5.1. Pengertian Leasing Menurut Literatur

Zaki Baridwan (1981: 1) memberikan definisi sebagai berikut: “Leasing adalah suatu perjanjian yang memberikan hak untuk menggunakan harta, pabrik atau alat-alat (tanah atau aktiva yang didepresiasi atau kedua-duanya) yang umumnya mempunyai jangka waktu tertentu.” Pihak-pihak yang langsung terlibat dalam perjanjian ini adalah:

1. Lessor : si pemilik aktiva yang bersangkutan atau yang menyewakan aktiva.
2. Lessee : yang memanfaatkan leasing yang bersangkutan atau yang menyewa aktiva.

II.5.2. Penggolongan Leasing

Pada awalnya leasing adalah bentuk transaksi sewa menyewa yang sederhana saja, kemudian keadaannya menjadi semakin berkembang sehingga leasing harus di bebaskan antara *Financial lease* atau *Capital lease* dan *Operating lease*.

Perkembangan berikutnya dari pada bentuk-bentuk leasing adalah semakin rumit dan bermacam-macam sehingga para ahli ekonomi merasa perlu untuk mengadakan penggolongan yang lebih lengkap dan terperinci.

Financial Accounting Standart Board No.13 yang dikemukakan oleh Englowood Cliffs (1982: 9-12) bahwa :

Penggolongan yang ditinjau dari sudut *lessee* :

1. *Capital Lease (Financial Lease)*.
2. *Operating Lease*, yakni semua *non capital lease*.

Penggolongan yang ditinjau dari sudut *lessor*:

1. *Direct Financial Lease*.
2. *Sales Type Lease*.
3. *Operating Lease*.
4. *Leveraged Lease*.

II.5.3. Metode Capital Lease

Jay M. Smith & K. Fred Skousen (1984: 545) mengemukakan bahwa :

Suatu lease digolongkan sebagai *Capital Lease* apabila lease tersebut memenuhi satu atau lebih kriteria berikut :

1. Pada saat berakhirnya kontrak *lease*, hak milik pindah ketangan *lessee*.
2. Perjanjian *lease* harus menyebutkan bahwa *lessee* mempunyai hak untuk membeli objek *lease* dengan harga yang menguntungkan, yaitu dengan harga yang lebih rendah dari taksiran nilai harganya (*expected fair value*) pada saat hak membeli tersebut dapat direalisasikan.
3. Jangka waktu *lease* sewa lebih besar dari taksiran dari 75 % taksiran umur ekonomis dari aktiva yang bersangkutan (dalam hal *lease* tersebut dimulai

pada saat properti sudah berumur sudah dipakai, maka kriteria ini tidak dapat diterapkan).

4. Pada waktu permulaan *lease*, *present value* dari pada pembayaran sewa minimum (tidak termasuk *executory cost*) harus sama atau lebih besar dari $90\% \times \text{fair market value}$).

Financial lease atau *capital lease* ini intinya adalah untuk mentransfer sebagian besar resiko dan keuntungan kepemilikan kepada *lessee* atau penyewa (yang mengikat kepentingan *lessor* atau pihak yang menyewakan pada penarikan sewanya). Jadi *lessee* dari suatu mesin biasanya membayar biaya perbaikan dan asuransi serta memikul resiko keuangan. Biasanya penyewaan dalam jangka panjang.

Karena setelah seluruh pembayaran sewa telah dilunasi selama masa sewa, maka *lessee* mempunyai hak pilih untuk membeli nilai sisa tersebut, jadi memiliki peralatan itu secara fisik. Dengan demikian *lessee* harus membukukan sebagai perolehan aktiva (*lease assets under capital lease*). (Drs. Siti Mirhani, MM. Ak; Jurnal Akuntansi Aktiva Leasing; 2003: 3)

Apabila transaksi leasing merupakan *Capital Lease* maka *lessee* harus mencantumkan aktiva *lease* pada sisi aktiva dan hutang *lease* pada sisi pasiva. Besarnya aktiva dan hutang yang mencantumkan adalah mana yang lebih rendah antara harga pasar aktiva atau nilai tunai pembayaran sewa minimum selama jangka waktu sewa yang dihitung pada awal jangka waktu sewa (tidak termasuk *Executory Cost* seperti pajak, asuransi dan biaya pemeliharaan). Harga pasar aktiva adalah harga pasar pada awal masa *lease*, sedangkan tingkat bunga yang

digunakan untuk mendiskontokan pembayaran *lease* adalah angka yang lebih rendah antara *incremental borrowing rate* dengan *implicit interest rate*. *Incremental borrowing rate* adalah tingkat bunga yang harus ditanggung *lessee* seandainya ia mau menambah hutangnya untuk membeli aktiva yang bersangkutan, sedangkan *implicit interest rate* adalah tingkat keuntungan yang diperoleh *lessor* dengan memberi jasa leasing tersebut.

Metode Depresiasi terhadap aktiva *lease* harus dilakukan sesuai dengan metode depresiasi yang dipilih perusahaan untuk mendepresiasi aktiva sejenis, sebagai periode depresiasi dapat digunakan taksiran umur ekonomis atau masa *lease*, tergantung kriteria mana yang terpenuhi sehingga transaksi *lease* harus dikapitalisasi. Jika yang terpenuhi adalah kriteria a atau b, maka periode depresiasi adalah umur ekonomis aktiva. Jika yang memenuhi adalah kriteria c atau d, maka periode depresiasi adalah jangka waktu *lease* (masa *lease*). Selama masa *lease*, hutang *lease* dikurangi dengan metode bunga dan sebagai pengembalian pokok pengalaman.

II.6. Pajak

Menurut Soemitro (dikutip dalam Mardiasmo 2006, h.1) pajak adalah :

“Turan rakyat kepada kas negara berdasarkan undang-undang (yang dapat dipaksakan) dengan tiada mendapat jasa timbal (kontraprestasi) yang langsung dapat ditunjukkan dan yang digunakan untuk membayar pengeluaran umum”.

Dari definisi tersebut, dapat disimpulkan bahwa pajak memiliki unsur :

1. Iuran dari rakyat kepada Negara

Yang berhak memungut pajak hanyalah negara. Iuran tersebut berupa uang (bukan barang).

2. Berdasarkan Undang-undang

Pajak dipungut berdasarkan atau dengan kekuatan undang-undang serta pelaksanaannya.

3. Tanpa jasa timbal atau kontraprestasi dari negara yang secara langsung dapat ditunjuk. Dalam pembayaran pajak tidak dapat ditunjukkan adanya kontraprestasi individual oleh pemerintah.

4. Digunakan untuk membiayai rumah tangga negara, yakni pengeluaran-pengeluaran yang bermanfaat bagi masyarakat luas. (Farrisa Tantry, Siti Khairani; Jurnal Pengaruh Pemahaman Wajib Pajak Orang Pribadi Terhadap Penerapan Self Assessment System Pada KPP Pratama Palembang Ilir Barat: 2009: 2)

II.7. Aktiva Tetap

Aktiva tetap adalah semua jenis aktiva yang akan dibeli atau diperoleh perusahaan untuk digunakan dalam kegiatan operasi/bisnisnya dalam waktu yang panjang, seperti membayar biaya di muka (*prepaid expense*) untuk waktu yang panjang, sejalan dengan penggunaannya, masing-masing aktiva akan disusutkan sesuai waktu/periode penggunaannya, yang disebut biaya/beban penyusutan (*depreciation expense*). (Elvy Maria Manurung; 2011: 91-92)

II.7.1. Pengkategorian Aktiva Tetap

Ada tiga jenis aktiva tetap yaitu aktiva tetap berwujud (*tangible assets*), aktiva tetap tak berwujud (*intangible assets*), dan sumber daya alam. Pengkategorian aktiva tetap dapat dilihat pada penjelasan dibawah ini :

1. Aktiva tetap berwujud adalah aktiva tetap yang manfaatnya cukup panjang, dan memiliki wujud secara fisik. Contohnya: tanah (*land*), bangunan (*buildings*), peralatan (*equipment*), mesin (*machine*), kendaraan (*vehicle*).
2. Aktiva tetap tidak berwujud adalah aktiva tetap yang manfaatnya cukup panjang, namun tidak berwujud secara fisik. Contohnya: hak paten/ hak cipta (*copy right*), merek dagang (*trademarks*), franchises/nama baik (*goodwill*).
3. Sumber daya alam adalah area-area tertentu yang diperoleh untuk menggali dan mengeksplorasi sumber daya alam. Contohnya: minyak bumi, mineral, emas, timah, dan sebagainya.

II.8. Microsoft Visual Basic 2010

Merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio 2010*. *Visual Studio* merupakan produk pemrograman andalan dari *Microsoft Corporation*, dimana di dalamnya berisi beberapa jenis IDE pemrograman seperti *Visual Basic*, *Visual C++*, *Visual Web Developer*, *Visual C#*, dan *Visual F#*. Semua IDE pemrograman tersebut sudah mendukung penuh implementasi *.Net Framework*

terbaru, yaitu *.Net Framework 4.0* yang merupakan pengembangan dari *.Net Framework 3.5*. Adapun *database* standar yang disertakan adalah *Microsoft SQL Server 2008 Express*. (Wahana Komputer; 2011: 2)

II.9. *SQL Server 2008*

SQL Server 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang *database*. *SQL Server* adalah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti *IBM* dan *Oracle*. *SQL Server 2008* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data.

Microsoft merilis *SQL Server 2008* dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut. Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan 2 jenis yaitu:

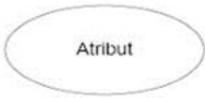
- Versi 32-bit(x86), yang biasanya digunakan untuk komputer dengan single prosesor (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi *Windows XP*.
- Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya *Core 2 Duo*) dan sistem operasi 64 bit seperti *Windows XP 64*, *Vista*, dan *Windows 7*. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut ini :

- Versi *Compact*, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi *desktop* pada *SQL Server 2000*. Versi ini juga digunakan pada handled drvice seperti *Pocket PC*, *PDA*, *SmartPhone*, *Tablet PC*.
- Versi *Express*, ini adalah versi “Ringan” dari semua versi yang ada (tetapi versi ini berbeda dengan versi *compact*) dan paling cocok untuk latihan para pengembang aplikasi. (Wahana Komputer; 2010: 2).

II.10. Entity Relationship Diagram (ERD)

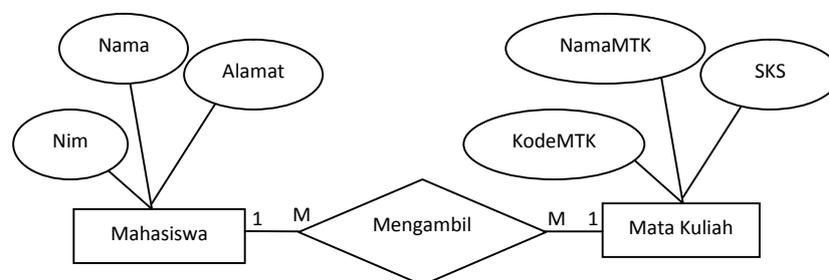
Entity Relationship Diagram atau ERD merupakan salah satu alat (tool) berbentuk grafis yang populer untuk *desain database*. Tool ini relatif lebih mudah dibandingkan dengan Normalisasi. Kebanyakan sistem analis memakai alat ini, tetapi yang jadi masalah, kalau kita cermati secara seksama, tool ini mencapai 2NF. (Yuniar Supardi; 2010 : 448).

Tabel II.1. Simbol ERD

Notasi	Keterangan
	Entitas , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi , menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut , berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah)
	Garis , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

(Sumber: Janner Simarmata, dkk; 2010)

Masing-masing komponen diberi nama entitas atau relasi yang diwakilinya. Sebagai ilustrasinya bayangkan anda mengambil bagian sistem basis data universitas yang terdiri dari mahasiswa dan mata kuliah. gambar II.3. menunjukkan diagram ERD dari contoh. Diagram menunjukkan bahwa ada dua kumpulan entitas yaitu mahasiswa dan mata kuliah dan bahwa relasi mengambil contoh mahasiswa dan mata kuliah. Contoh dapat dilihat pada gambar II.2. berikut ini :



Gambar II.2. Diagram ERD
(Sumber: Janner Simarmata, dkk; 2010)

II.11. 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 rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

II.11.1. Bentuk-bentuk Normalisasi

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

Contoh yang kita gunakan di sini adalah sebuah perusahaan 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.

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)

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.

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

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama. Tabel pemasok berada pada 1NF, tetapi tidak pada 2NF 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 1NF ke 2NF 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

P#	Status	Kota
P1	20	Yogyakarta
P2	10	Medan
P3	10	Medan
P4	20	Yogyakarta
P5	30	Bandung

Barang

P#	B#	Qty
P1	B1	300
P1	B2	200
P1	B3	400
P1	B4	200
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 tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF 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 2NF, tetapi belum berada pada 3NF 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 3NF 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 3NF, 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 (3NF)

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. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF. Lebih lanjut, mereka berpendapat bahwa keuntungan yang didapat mengubah entitas ke 4NF dan 5NF sangat kecil sehingga tidak perlu dikerjakan. Bentuk Normal Boyce-Code (BCNF) adalah versi 3NF 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 Tahap Keempat dan Kelima

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

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia 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*). (Janner Simarmata; 2010: 76).

II.12. Unified Modeling Language (UML)

Menurut Windu Gata (2013 : 4) 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.

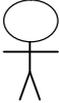
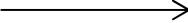
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

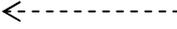
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

- *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.5. berikut 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>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>

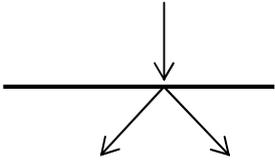
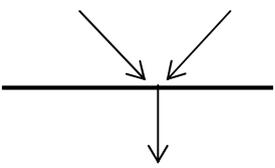
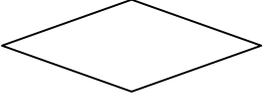
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber: Windu Gata; 2013: 4)

- 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. berikut ini :

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 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> .

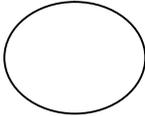
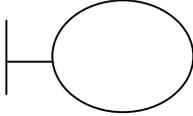
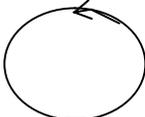
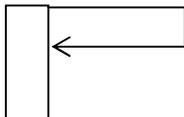
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.
---	--

(Sumber : Windu Gata; 2013: 6)

- 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. berikut ini :

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 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.

	<p><i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Windu Gata; 2013: 7)

- *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. berikut 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 : Windu Gata; 2013: 8)