

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem merupakan serangkaian bagian yang saling tergantung dan bekerjasama untuk mencapai tujuan tertentu. Suatu sistem pasti tersusun dari sub-sub system yang lebih kecil yang juga saling tergantung dan bekerjasama untuk mencapai tujuan. Tujuan dasar suatu sistem tergantung pada jenis sistem itu sendiri. Termasuk juga sistem buatan manusia seperti sistem yang terdapat di sekolah, organisasi bisnis, atau instansi pemerintah juga mempunyai tujuan yang berbeda-beda. (Anastasia Diana & Lilis Setiawati ; 2011 : 3).

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, dsb. Secara konseptual seluruh sistem organisasional mencapaitujuannya 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. (Mujilan ; 2012 : 1).

Sistem Informasi yang kadang kala disebut sebagai sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen baik manual ataupun berbasis computer yang terintegrasi untuk mengumpulkan, menyimpan, dan mengola data serta menyediakan informasi kepada pihak-pihak yang berkepentingan sebagai pemakai informasi tersebut.

Komponen dari Sistem Informasi terdiri dari :

1. Input

Input dalam system informasi adalah data-data yang relevan untuk menghasilkan informasi yang diinginkan. Contohnya seperti penerimaan pesanan pembelian peralatan kantor.

2. Proses

Proses adalah langkah-langkah yang perlu dilakukan untuk mengolah data menjadi informasi. Contohnya seperti update catatan persediaan untuk setiap item peralatan kantor setiap kali terjadi pembelian peralatan kantor.

3. Output

Output adalah berupa informasi yang merupakan hasil dari pemrosesan data. Contohnya berupa daftar persediaan berbagai jenis peralatan kantor. (Anastasia Diana & Lilis Setiawati ; 2011 : 4).

II.3. Sistem Informasi Akuntansi

Sistem Informasi Akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Misalnya, salah satu input dari Sistem Informasi

Akuntansi pada sebuah Toko Baju yang berupa transaksi penjualan. Kita memproses transaksi dengan mencatat penjualan tersebut kedalam jurnal penjualan, mengklarifikasi transaksi dengan menggunakan kode rekening, dan memposting transaksi ke dalam jurnal. Kemudian, secara periodic Sistem Informasi Akuntansi akan menghasilkan Output berupa laporan keuangan yang terdiri dari Neraca dan Laporan Laba Rugi. (Anastasia Diana & Lilis Setiawati ; 2011 : 4-5).

II.3.1. Sistem Informasi Akuntansi Persediaan

Persediaan merupakan faktor yang paling penting dalam perusahaan dagang dan juga sering disebut sebagai sumber kekayaan bagi perusahaan. Untuk dapat mengelola persediaan dengan baik, dibutuhkan sistem informasi akuntansi persediaan. Sistem informasi persediaan adalah alat yang digunakan untuk menghasilkan informasi persediaan sehingga memudahkan perusahaan dapat mengambil keputusan yang berkaitan dengan persediaan.

Jaringan prosedur yang bersangkutan dengan sistem persediaan bahan baku adalah :

1. Prosedur pencatatan harga pokok persediaan yang dibeli Dalam prosedur ini dicatat harga pokok persediaan yang dibeli

Dokumen yang digunakan :

- a. Laporan penerimaan barang

Laporan penerimaan barang digunakan oleh bagian gudang sebagai dasar pencatatan tambahan kuantitas barang dari pembelian kedalam kartu gudang.

- b. Bukti kas keluar

Bukti kas keluar yang dilampiri dengan laporan penerimaan barang, surat order pembelian, dan faktur dari pemasok sebagai dokumen sumber dalam pencatatan harga pokok persediaan yang dibeli dalam register bukti kas keluar, bukti kas keluar juga dipakai sebagai dasar pencatatan tambahan kuantitas dan harga pokok persediaan.

2. Prosedur pencatatan harga pokok persediaan yang dikembalikan kepada pemasok Jika persediaan yang telah dibeli dikembalikan kepada pemasok, maka akan mengurangi jumlah persediaan dalam kartu gudang yang diselenggarakan oleh bagian gudang dan harga pokok persediaan yang dicatat oleh bagian kartu persediaan dalam kartu persediaan yang bersangkutan.

Dokumen yang digunakan :

a. Laporan pengiriman barang

Digunakan oleh bagian gudang untuk mencatat jumlah persediaan yang dikirimkan kembali kepada pemasok kedalam kartu gudang

b. Memo debit yang diterima dari Bagian Perencanaan dan Keuangan

Digunakan oleh bagian kartu persediaan untuk mencatat jumlah dan harga pokok persediaan yang dikembalikan kepada pemasok dalam kartu gudang

3. Prosedur permintaan dan pengeluaran barang gudang

Dalam prosedur ini dicatat harga pokok persediaan bahan baku, bahan penolong, bahan habis pakai pabrik, yang dipakai dalam kegiatan produksi dan kegiatan non produksi.

Dokumen yang digunakan :

a. Bukti permintaan dan pengeluaran barang gudang

Bukti ini digunakan oleh bagian gudang untuk mencatat pengurangan persediaan karena pemakaian intern, digunakan juga oleh bagian kartu persediaan untuk mencatat berkurangnya jumlah dan harga pokok persediaan, digunakan juga sebagai dokumen sumber dalam pencatatan pemakaian persediaan kedalam jurnal bahan baku atau jurnal umum.

4. Prosedur pengembalian barang gudang

Pengembalian barang gudang akan mengurangi biaya dan menambah persediaan barang di gudang

Dokumen yang digunakan :

a. Bukti pengembalian barang gudang

Bukti ini digunakan oleh bagian gudang untuk mencatat tambahan jumlah

persediaan kedalam kartu gudang, digunakan juga oleh bagian kartu persediaan untuk mencatat tambahan jumlah dan harga pokok persediaan kedalam kartu persediaan, untuk mencatat berkurangnya biaya kedalam kartu biaya dan untuk mencatat pengembalian barang gudang tersebut kedalam jurnal umum.

5. Perhitungan fisik persediaan

Sistem penghitungan fisik persediaan umumnya digunakan oleh perusahaan untuk menghitung secara fisik persediaan yang disimpan di gudang, yang hasilnya digunakan untuk meminta pertanggung jawaban bagian gudang mengenai pelaksanaan fungsi penyimpanan, dan pertanggung jawaban bagian kartu persediaan mengenai keandalan catatan persediaan yang diselenggarakan, serta untuk melakukan penyesuaian terhadap catatan persediaan di bagian kartu persediaan.

Dokumen yang digunakan dalam sistem perhitungan fisik persediaan :

a. Kartu penghitungan fisik

Digunakan untuk merekam hasil perhitungan fisik persediaan. Dalam penghitungan fisik persediaan, setiap jenis persediaan dihitung dua kali secara independent oleh penghitung dan pengecek.

b. Daftar hasil penghitungan fisik

Digunakan untuk meringkas data yang telah direkam dalam bagian kartu persediaan fisik yang diisi oleh pengecek. Data yang disalin ke daftar ini adalah : nomer kartu persediaan fisik, nomer kode persediaan, nama persediaan, kuantitas dan satuan.

c. Bukti memorial

Digunakan untuk membukukan penyesuaian rekening persediaan sebagai akibat dari hasil penghitungan fisik kedalam jurnal umum. Data yang digunakan sebagai dasar pembuatan bukti memorial adalah selisih jumlah harga pokok total dalam daftar hasil penghitungan fisik dengan saldo harga pokok persediaan yang bersangkutan menurut kartu persediaan.

Catatan akuntansi yang digunakan dalam sistem penghitungan fisik adalah sebagai berikut :

- Kartu persediaan

Digunakan untuk mencatat penyesuaian terhadap data (jumlah dan harga pokok total) yang tercantum dalam kartu persediaan oleh bagian kartu persediaan berdasar hasil penghitungan fisik persediaan.

- Kartu gudang

Digunakan untuk mencatat penyesuaian terhadap data persediaan (jumlah) yang tercantum dalam kartu gudang yang diselenggarakan oleh bagian gudang berdasarkan hasil penghitungan fisik persediaan.

- Jurnal umum

Digunakan untuk mencatat jurnal penyesuaian rekening persediaan karena adanya perbedaan antara saldo yang dicatat dalam rekening persediaan dengan saldo menurut penghitungan fisik. (Berlian Astrarini ; 2010 : 4-7).

II.3.2. Tujuan Sistem Informasi Akuntansi

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 atau Kekayaan Perusahaan

Harta atau Kekeyaan disini Meliputi kas perusahaan, persediaan barang dagangan, termasuk asset tetap perusahaan.

2. Menghasilkan Beragam Informasi Untuk Pengambilan Keputusan

Misal, pengelola Toko Swalayan memerlukan informasi mengenai barang apa yang diminati oleh konsumen.

3. Menghasilkan Informasi untuk Pihak Eksternal

Setiap pengelola usaha memiliki kewajiban untuk membayar pajak. Besarnya pajak yang dibayar tergantung pada omset penjualan atau tergantung pada laba rugi usaha.

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 sangat 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, data historis yang diproses oleh system informasi dapat digunakan untuk meramal pertumbuhan penjualan dan aliran kas atau untuk mengetahui tren jangka panjang beserta korelasinya. (Anastasia Diana & Lilis Setiawati ; 2011 : 5-7).

II.3.3. Keterkaitan Sistem Informasi Akuntansi dengan Proses Bisnis dan Organisasi

Sistem Informasi Akuntansi memiliki peranan yang penting dalam proses bisnis karena Sistem Informasi Akuntansi mengidentifikasi, mengukur, dan mencatat proses bisnis tersebut dalam suatu model yang sedemikian rupa sehingga informasi yang dihasilkan dapat dimanfaatkan oleh pihak-pihak yang berkepentingan. Dari sudut pandang akuntansi model, proses bisnis tersebut diwujudkan dalam bentuk siklus transaksi. Pengelompokan siklus transaksi biasanya berkaitan dengan beberapa kejadian yang berurutan.

Organisasi merupakan suatu sistem yang tersusun dari sub-sub sistem. Sub-sub sistem dalam organisasi meliputi manajemen, operasi, dan informasi. Sub-sub sistem tersebut berkaitan baik dengan pihak internal perusahaan, seperti

karyawan, maupun dengan pihak eksternal perusahaan, seperti pelanggan dan instansi pemerintah. (Anastasia Diana & Lilis Setiawati ; 2011 : 9).

II.4. Perlengkapan Kantor

Moenir (2004 : 87), menyatakan bahwa yang dimaksud dengan perlengkapan kantor adalah segenap proses penyelenggaraan yang meliputi fungsi perencanaan, mengatur pelaksanaan dan mengontrol terhadap barang-barang milik kantor sehingga tercapainya efisiensi di bidang perlengkapan kantor.

The Liang Gie (2002 : 174), menyatakan bahwa asas-asas efisiensi dalam pengelolaan barang meliputi :

1. Asas Perencanaan berarti menggambarkan di muka mengenai tindakan yang akan dilaksanakan untuk mencapai tujuan. Perwujudan dari asas tersebut meliputi : pedoman tentang warkat, pedoman tentang penetapan prosedur, pedoman tentang pengadaan perlengkapan kantor, pedoman tentang perencanaan formulir.
2. Asas Penyederhanaan, berarti membuat suatu sistem yang ruwet atau pekerjaan yang sukar menjadi lebih mudah dan ringan.
3. Asas Penghematan, berarti mencegah pemakaian benda-benda secara berlebihan, sehingga biaya pekerjaan menjadi mahal.
4. Asas Penghapusan, berarti meniadakan langkah-langkah atau kegiatan dalam pelaksanaan sesuatu pekerjaan yang dianggap kurang perlu atau tidak berhubungan dengan hasil kerja.

5. Asas Penggabungan, berarti mempersatukan pekerjaan-pekerjaan yang mempunyai persamaan atau benda-benda yang mungkin dikerjakan. (Rian Fahlevi ; 2012 : 3).

II.4.1. Metode FIFO

First-in, First-Out method (FIFO) adalah metode penilaian yang menganggap barang yang pertama kali masuk diasumsikan keluar pertama kali pula. Jika barang dikeluarkan dari gudang maka akan dihargai sebesar harga perolehan yang pertama. Metode ini penghasilan hanya akan dibebankan dengan biaya yang sesuai dengan arus barang yang dijual sehingga nilai persediaan menggambarkan nilai yang berlaku saat ini.

Rumus :

$$\text{barang yang pertama kali masuk} = \text{keluar pertama kali}$$

(Sri Wahyuning ; 2015 : 27).

II.4.2. Pengendalian Internal Pembelian

Pengendalian internal atas pembelian memiliki beberapa unsur yang seharusnya ada dalam sistem akuntansi pembelian, dimana unsur-unsur tersebut dirancang untuk mencapai tujuan pokok dari sebuah pengendalian internal akuntansi, yakni menjaga kekayaan (persediaan) dan kewajiban perusahaan (utang dagang atau bukti kas keluar yang akan dibayar), menjamin ketelitian dan keandalan data akuntansi (utang dan persediaan). Mulyadi (2001:312-318) menyatakan bahwa untuk merancang unsur-unsur pengendalian internal akuntansi

yang diterapkan dalam sistem akuntansi pembelian, dibutuhkan unsur pokok sistem pengendalian internal yang terdiri dari organisasi, sistem otorisasi dan prosedur pencatatan, dan praktik yang sehat. (Bernadien Kristia Devi ; 2012 : 31).

II.5. Teknik Normalisasi

Sebuah rancangan database dapat dikategorikan kurang baik, apabila :

1. Data disimpan di beberapa tempat atau *record*
2. Adanya redundansi atau pengulangan data sehingga memboroskan ruang penyimpanan serta menyulitkan ketika melakukan update atau perubahan data.
3. Timbul nilai null (*null value*)
4. Menimbulkan informasi yang tidak bermanfaat.

Oleh karena itu, normalisasi perlu dilakukan. Normalisasi database membuat struktur database yang dihasilkan semakin optimal. Intinya, proses ini menghindarkan pengguna dari redundansi data. Normalisasi database biasanya dilakukan pada database yang berskala besar. (Eko Priyo Utomo ; 2014 : 72-73).

Adapun bentuk-bentuk Normalisasi adalah sebagai berikut :

1. Bentuk normal tahap pertama (1NF)

Untuk mencegah berbagai macam anomali, dapat menormalkan tabel menjadi dua tabel bentuk normalisasi 1 (1NF). Aturan yang ada dalam 1NF adalah sebuah baris data tidak dapat terdiri dari kumpulan data yang sama, masing-masing kolom harus mempunyai nilai yang unik, dan masing-masing baris mempunyai *identifier* unik yaitu *primary key*.

Tabel II.1. Normalisasi Bentuk 1 (1NF)

Mhs_ID	Mhs_Nama
201	Amir
202	Agus
203	Bobby
204	Cahya
205	Joko

Matkul_ID	Matkul_Mhs_ID	Matkul_Nama
50	201	Rekayasa Perangkat Lunak
51	201	Komunikasi Data
52	202	Manajemen Proyek
53	203	Rekayasa Perangkat Lunak
54	204	Penambangan Data
55	205	Manajemen Strategis

(Sumber : Eko Priyo Utomo ; 2014)

Penggabungan antara kolom Matkul_ID dan Matkul_Mhs_ID merupakan *primary key*.

2. Bentuk normal tahap kedua (2NF)

Normalisasi bentuk kedua (2NF) harus memenuhi syarat 1NF dan tidak ada ketergantungan *primary key*.

Tabel II.2. Tabel Pelanggan

Pelanggan_ID	Pelanggan_Nama	Order_ID	Order_Nama	Detail
201	Amir	60	Jeruk	Jeruk bali
202	Amir	61	Apel	Apel Malang
203	Bobby	62	Bakpia	Bakpia Yogya
204	Cahya	63	Tahu	Tahu sumedang

(Sumber : Eko Priyo Utomo ; 2014)

Pada tabel pelanggan diatas, penggabungan antara Pelanggan_Id dan Ordeer_Id merupakan *primary key*. Tabel pelanggan diatas memenuhi persyaratan normalisasi pertama (1NF) namun belum 2NF karena ada ketergantungan parsial kolom pada *primary key*. Kolom Pelanggan_Nama hanya bergantung pada Pelanggan_Id, Order_Nama bergantung pada Order_Id dan tidak ada link/hubungan antara Detail dan Pelanggan_Nama. (Eko Priyo Utomo ; 2014 : 74-75).

Tabel II.3. Normalisasi Bentuk 2 (2NF)

Pelanggan_Id	Pelanggan_Nama	Order_Id	Order_Nama
201	Amir	60	Jeruk
203	Bobby	61	Apel
204	Cahya	62	Bakpia
		63	Tahu

Pelanggan_Id	Order_Id	Barang_Detail
201	60	Jeruk bali
202	61	Apel Malang
203	62	Bakpia Yogya
204	63	Tahu sumedang

(Sumber : Eko Priyo Utomo ; 2014)

3. Bentuk normal tahap ketiga (3NF)

Normalisasi bentuk ketiga (3NF) dapat dibuat dengan kolom-kolom yang bukan primer tergantung pada *primary key*. Ketergantungan fungsional transitif seharusnya dikeluarkan dari tabel, tabel memenuhi persyaratan normalisasi kedua (2NF). (Eko Priyo Utomo ; 2014 : 76).

Tabel II.4. Normalisasi Bentuk 3 (3NF)

Mahasiswa_ID	Mahasiswa_Nama	Mahasiswa_KodePos
10	Melani	44123
11	Nania	64111
12	Rosnia	22780

Mahasiswa_KodePos	Mahasiswa_Jalan	Mahasiswa_Kota	Mahasiswa_Kabupaten
44123	Jl. Kucing 12	Condong Catur	Sleman
64111	Jl. Harimau 21	Keras	Kediri
22780	Jl. Pelanduk 33	Nagrek	Sumedan

(Sumber : Eko Priyo Utomo ; 2014)

II.6. Visual Basic 2010

Visual Basic 2010 merupakan salah satu bagian dari produk Microsoft *Visual Studio 2010*. *Visual Studio* merupakan produk pemrograman andalan dari *Microsoft Corporation*, di mana didalamnya 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 4.0 yang merupakan pengembangan dari .Net Framework 3.5. Adapun database standar yang disertakan adalah *Microsoft SQL Server 2008 R2*.

Visual Basic 2010 merupakan versi perbaikan pendahulunya, yaitu *Visual Basic 2008*. Beberapa pengembangan yang terdapat didalamnya antara lain dukungan terhadap *library* terbaru dari *Microsoft*, yaitu .Net Framework 4.0, dukungan terhadap pengembangan aplikasi menggunakan *Microsoft SilverLight*, dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap database-database, baik standalone maupun database server

Bahasa *Visual Basic 2010* sendiri awalnya berasal dari bahasa pemrograman yang sangat populer dikalangan programmer computer, yaitu bahasa BASIC, yang oleh *Microsoft* diadaptasi dalam program *Microsoft Quick BASIC*. Seiring dengan berkembangnya teknologi komputasi dan desain, *Microsoft* mengeluarkan produk yang dinamakan *Microsoft Visual Studio* dengan *Visual Basic* didalamnya. Saat ini versi *Microsoft Visual Studio* yang beredar adalah versi 10 yang populer dengan nama *Microsoft Visual Studio 2010*, yang didalamnya termasuk *Microsoft Visual Basic 2010*. (Wahana Komputer ; 2011 : 2-3).

II.7. Microsoft 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 :

- a. 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.
- b. Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya Core 2 Duo) dan system operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7.

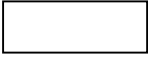
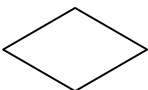
Sedangkan secara keseluruhan terdapat versi-versi seperti berikut ini :

- a. 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.
- b. 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. Versi ini memuatExpress Manager standar, integrasi dengan CLR dan XML. (Wenny Widya;2010:3-4).

II.8. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD adalah data model yang didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitias dan hubungan antar objek (Simarmata & Paryudi ; 2006:59). Berikut Penjelasan tentang komponen-komponen ERD ditunjukkan pada Tabel II.5 :

Tabel II.5. Simbol ERD

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

(Sumber : D. Tri Octafian ; 2011)

II.8.1. Pemetaan Kardinalitas

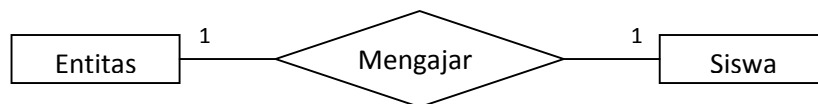
Pemetaan kardinalitas menyatakan jumlah entitas di mana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi.

1. *One to One*

Sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A.

Contoh :

Pada pengajaran privat, satu guru satu siswa. Seorang guru mengajar seorang siswa, seorang siswa diajar oleh seorang guru.



Gambar II.1. Contoh Pemetaan kardinalitas One to One
(Sumber : D. Tri Octafian ; 2011)

2. *One to Many/ Many to One*

Sebuah entitas pada A berhubungan dengan lebih dari satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A, atau sebaliknya (*Many to One*).

Contoh :

Dalam satu perusahaan, satu bagian mempekerjakan banyak pegawai. Satu bagian mempekerjakan banyak pegawai, satu pegawai kerja dalam satu bagian.



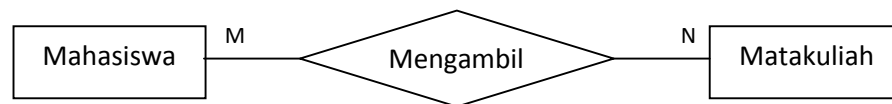
Gambar II.2. Contoh Pemetaan kardinalitas One to Many / Many to One
(Sumber : D. Tri Octafian ; 2011)

3. *Many to Many*

Sebuah entitas pada A berhubungan dengan lebih dari satu entitas pada B dan sebuah entitas pada B berhubungan dengan lebih dari satu entitas pada A.

Contoh :

Dalam universitas, seorang Mahasiswa dapat mengambil banyak mata kuliah. Satu mahasiswa mengambil banyak mata kuliah dan satu mata kuliah diambil banyak mahasiswa.



Gambar II.3. Contoh Pemetaan kardinalitas Many to Many

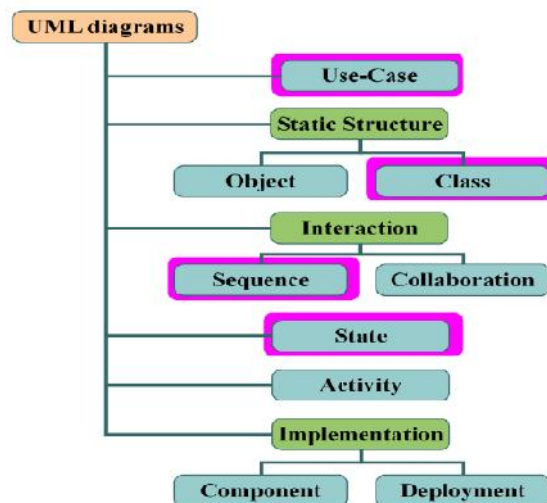
(Sumber : D. Tri Octafian ; 2011)

II.9. Unified Modeling Language (UML)

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal dalam mengembangkan sistem berorientasi objek. Hal ini dikarenakan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*Sharing*) dan mengkomunikasikan rancangan dengan baik.

UML Merupakan satu kesatuan bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama Metode *Design Object Oriented* dimana metode ini menjadikan proses analisis dan design ke dalam empat tahapan iteratif, yaitu : Identifikasi kelas-kelas dan objek-objek, Identifikasi semantik dari hubungan Objek dan Kelas tersebut, perincian Interface dan Implementasi. (Haviluddin ; 2011 : 1)

Berikut gambar dari diagram UML :



Gambar II.4. Diagram UML

(Sumber : Haviluddin ; 2011)

II.9.1. Use Case Diagram

Use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi.

Hal yang ditekankan pada diagram ini adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* menyatakan sebuah aktivitas atas pekerjaan tertentu. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain. Oleh karena itu, duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case*

lain dengan *behavior*-nya sendiri. Hubungan generalisasi antar *use case* menunjukkan bahwa *use case* merupakan spesialisasi dari yang lain (Hamim Tohari ; 2014: 47-48).

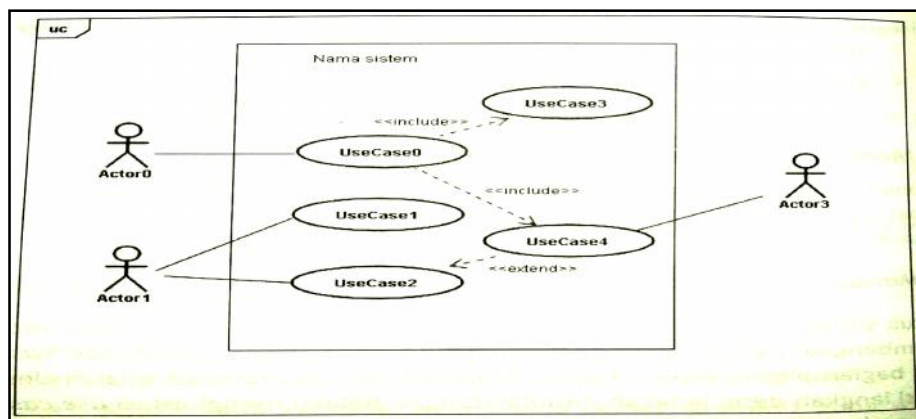
1. Menyusun Diagram Use Case

Langkah-langkah yang dibutuhkan untuk menyusun diagram *use case* :

- Mengidentifikasi pelaku bisnis.
- Mengidentifikasi *use case* persyaratan bisnis.
- Membuat diagram model *use case*.
- Mendokumentasikan naratif *use case* persyaratan bisnis (Hamim Tohari; 2014 : 49).

2. Elemen-Elemen Diagram Use Case

Beberapa elemen yang digunakan pada diagram *use case* dapat dilihat pada gambar II.5 berikut.



Gambar II.5. Elemen-Elemen Diagram Use Case
(Sumber : Hamim Tohari ; 2014)

- a. Sistem, menyatakan batasan sistem dalam relasi dengan *actor-actor* yang menggunakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem). Sistem digambarkan dengan segi empat yang membatasi semua *use case* dalam sistem terhadap pihak mana sistem akan berinteraksi.
- b. *Actor* atau actor, dapat berupa manusia, sistem, atau *device* yang memiliki peranan dalam keberhasilan operasi dari sistem. Digambarkan dengan *icon* yang mungkin bervariasi namun konsepnya sama.
- c. *Use case*, mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, sistem tidak akan memenuhi permintaan *user/actor*. Setiap *use case* mengekspresikan *goal* dari sistem yang harus dicapai. Diberi nama sesuai dengan *goal*-nya dan digambarkan dengan elips (dengan nama didalamnya).
- d. *Association*, mengidentifikasikan interaksi antara setiap *actor* tertentu dengan setiap *use case* tertentu. Digambarkan dengan garis antara *actor* terhadap *use case* yang bersangkutan. Asosiasi bias berarah (garis dengan anak panah) jika komunikasi satu arah, namun umumnya terjadi kedua arah (tanpa anak panah) karena selalu diperlukan demikian.
- e. *Stereotype*, memungkinkan perluasan UML tanpa memodifikasinya. Berperan sebagai kualifier pada suatu elemen modal. Menyediakan informasi lebih banyak mengenai peranan dari elemen tanpa menyebutkan implementasinya.

- f. *Dependency*, dependensi <<include>> mengidentifikasi hubungan antar dua *use case* dimana yang satu memanggil yang lain, digambarkan dengan garis putus-putus bermata panah dengan notasi <<include>> pada garis, dan arah mata panah sesuai dengan arah pemanggilan. Dependensi <<extend>> jika pemanggilan memerlukan adanya kondisi tertentu maka berlaku dependensi <<extend>> dan digambarkan serupa dengan dependensi <<include>> kecuali arah panah berlawanan (Hamim Tohari, 2014;51-54).

II.9.2. Class Diagram

Kelas (*Class*) adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan perancangan berorientasi objek. Kelas menggambarkan keadaan (atribut/property) suatu system, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Dalam pemodelan statis dari sebuah sistem, diagram kelas biasanya digunakan untuk memodelkan salah satu dari tiga hal berikut :

1. Perbendaharaan dari sistem.
2. Kolaborasi.
3. Skema basis data *logical*.

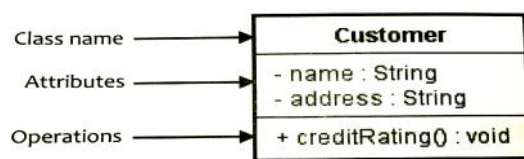
Kelas memiliki tiga area pokok :

1. Nama (dan *stereotype*).
2. Atribut.
3. Metode atau operasi.

Atribut dan metode dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.

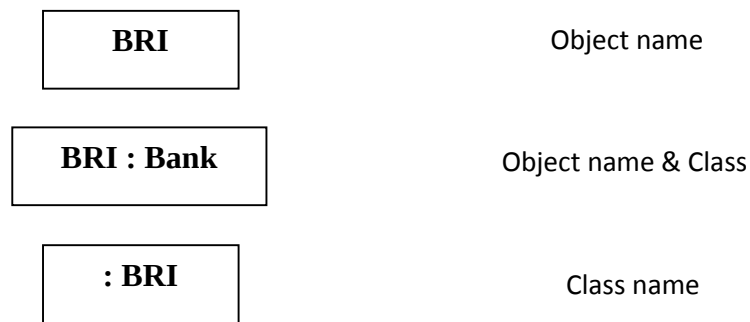
Kelas dapat berupa implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metode pada saat *run-time*. (Hamim Tohari ; 2014: 83-84).



Gambar II.6. Contoh Kelas
(Sumber : Hamim Tohari ; 2014)

II.9.3. Sequence Diagram

Sequence Diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi system. Dalam UML, objek pada diagram *sequence* digambarkan dengan segi empat, yang berisi nama dari objek yang digarisbawahi. Terdapat 3 cara untuk menamai objek yaitu, nama objek, nama objek dan *class* serta nama *class*. Misalnya :



Gambar II.7. Penamaan Objek pada *Sequence Diagram*
 (Sumber : Hamim Tohari ; 2014)

Pada *diagram sequence*, setiap objek hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antar objek digambarkan dengan anak panah dari objek yang mengirimkan pesan ke objek yang menerima pesan (Hamim Tohari ; 2014 : 101).

II.9.4. Activity Diagram

Activity diagram memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Membuat *activity diagram* pada awal pemodelan proses cukup menguntungkan untuk membantu memahami keseluruhan proses. *Activity diagram* juga bermanfaat untuk menggambarkan *parallel behaviour* atau menggambarkan interaksi antara beberapa *use case*.

Elemen-Elemen Activity Diagram :

1. Status *Start* (mulai) dan *end* (akhir).
2. Aktivitas yang mempresentasikan sebuah langkah dalam *workflow*.
3. *Transition* menunjukkan terjadinya perubahan status aktivitas (*transition show what State follows another*).

4. Keputusan yang menunjukkan *alternative* dalam *workflow*.
5. *Synchronization bars* yang menunjukkan *subflow parallel*.
Synchronization bars dapat digunakan untuk menunjukkan *concurrent threads* pada *workflow* proses bisnis.
6. *Swimlanes* yang mempresentasikan *role* bisnis yang bertanggung jawab pada aktivitas yang berjalan (Hamim Tohari ; 2014 : 114-115).