

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Menurut Kusrini & Andi Koniyo (2007 : 5), kata *sistem* mempunyai beberapa pengertian, tergantung dari sudut pandang mana kata tersebut didefinisikan. Secara garis besar ada dua kelompok pendekatan, yaitu :

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, yang dalam hal ini sistem itu didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi di dalam sistem. Prosedur (*procedure*) didefinisikan oleh F. Nueshl sebagai urutan operasi kerja (tulis-menulis), yang biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

Pendekatan sistem yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan sistem sebagai sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian di dalam suatu sistem, komponen-komponen ini tidak dapat berdiri

sendiri-sendiri, tetapi sebaliknya, saling berhubungan hingga membentuk satu kesatuan sehingga tujuan sistem itu dapat tercapai.

II.1.1. Karakteristik Sistem

Menurut Kusriani & Andi Koniyo (2007 : 6), sistem mempunyai beberapa karakteristik atas sifat-sifat tertentu, antara lain :

1. Komponen Sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang saling bekerja sama membentuk suatu komponen sistem atau bagian-bagian dari sistem.

2. Batasan Sistem (*Boundary*)

Merupakan daerah yang membatasi suatu sistem dengan sistem yang lain atau dengan lingkungan kerjanya.

3. Subsistem

Bagian-bagian dari sistem yang beraktivitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasarannya masing-masing.

4. Lingkungan Luar Sistem (*Environment*)

Suatu sistem yang ada di luar dari batas sistem yang dipengaruhi oleh operasi sistem.

5. Penghubung Sistem (*Interface*)

Media penghubung antara suatu subsistem dengan subsistem lain. Adanya penghubung ini memungkinkan berbagai sumber daya mengalir dari suatu subsistem ke subsistem lainnya.

6. Masukan Sistem (*Input*)

Energi yang masuk ke dalam sistem, berupa perawatan dan sinyal.

Masukan perawatan adalah energi yang dimasukkan supaya sistem tersebut dapat berinteraksi.

7. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

8. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan mengubah masukan menjadi keluaran.

9. Sasaran Sistem (*Object*)

Tujuan yang ingin dicapai oleh sistem, akan dikatakan berhasil apabila mengenai sasaran atau tujuan.

II.1.2. Klasifikasi Sistem

Menurut Kusriani & Andi Koniyo (2007 : 7), suatu sistem dapat diklasifikasikan menjadi seperti berikut :

1. Sistem abstrak dan fisik

Sistem abstrak adalah suatu sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan sistem fisik adalah sistem yang ada secara fisik

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam sedangkan sistem buatan manusia adalah sistem yang dirancang oleh manusia.

3. Sistem tertentu dan sistem tak tentu

Sistem tertentu adalah suatu sistem yang operasinya dapat diprediksi secara tepat sedangkan sistem tak tertentu adalah sistem dengan perilaku ke depan yang tidak dapat diprediksi.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak terpengaruh oleh lingkungan luar atau otomatis, sedangkan sistem terbuka adalah sistem yang berhubungan dan berhubungan oleh lingkungan luar.

II.2. Pengertian Informasi

Menurut Kusrini & Andi Koniyo (2007 : 7), informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Data belum memiliki nilai, sedangkan informasi sudah memiliki nilai. Informasi dikatakan bernilai bila manfaatnya lebih besar dibandingkan biaya untuk mendapatkannya.

II.2.1. Kualitas Informasi

Menurut Kusrini & Andi Koniyo (2007 : 8), informasi yang berkualitas memiliki 3 karakteristi, yaitu :

1. Akurat (*Accurate*)

Informasi harus bebas dari kesalahan, tidak bias ataupun menyesatkan. Akurat juga berarti bahwa informasi itu harus dapat dengan jelas mencerminkan maksudnya.

2. Tepat pada waktunya (*timeliness*)

Informasi yang datang pada penerima tidak boleh terlambat, didalam pengambilan keputusan, informasi yang sudah usang tidak lagi bernilai. Bila informasi datang terlambat sehingga pengambilan keputusan terlambat dilakukan, hal itu dapat berakibat fatal bagi perusahaan.

3. Relevan (*relevance*)

Informasi yang disampingkan harus mempunyai keterkaitan dengan masalah yang akan dibahas dengan informasi tersebut. Informasi harus bermanfaat bagi pemakai, disamping karakteristik, nilai informasi juga ikut menentukan kualitasnya. Nilai informasi (*value of information*) ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih besar dibandingkan biaya untuk mendapatkannya.

II.3. Sistem Informasi

Menurut Kusriani & Andi Koniyo (2007 : 8), untuk menghasilkan informasi yang berkualitas maka dibuatkanlah sistem informasi. Sistem informasi didefinisikan oleh Robert A. Litch dan K. Roscoe Bavis sebagai berikut :
 ”Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.”

Defenisi umum sistem informasi adalah : ”Sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan.”

II.3.1. Komponen Sistem Informasi

Menurut Kusrini & Andi Koniyo (2007 : 9), 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 (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

II.4. Sistem Informasi Akuntansi

Menurut Anastasia Diana & Lilis Setiawati (2011:3) 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 saling tergantung dan bekerja sama untuk mencapai tujuan.

Menurut Kusriani & Andi Koniyo (2007 : 10) Sistem Informasi akuntansi merupakan sebuah sistem informasi yang mengubah data transaksi bisnis menjadi informasi keuangan yang berguna bagi pemakainya.

Tujuan dari sistem informasi akuntansi adalah :

1. Mendukung operasi sehari-hari.
2. Mendukung pengambilan keputusan manajemen.
3. Memenuhi kewajiban yang berhubungan dengan pertanggung jawaban.

Komponen-komponen yang terdapat dalam sistem informasi akuntansi adalah sebagai berikut:

1. Orang-orang yang mengoperasikan sistem tersebut.
2. Prosedur-prosedur, baik manual maupun yang terotomatisasi, yang dilibatkan dalam pengumpulan, pemrosesan dan penyimpanan data aktivitas-aktivitas organisasi.
3. Data tentang proses-proses bisnis.
4. Software yang dipakai untuk memproses data organisasi.
5. Infrastruktur teknologi informasi.

Didalam organisasi, sistem informasi akuntansi berfungsi untuk :

1. Mengumpulkan dan menyimpan aktivitas yang dilaksanakan disuatu organisasi, sumber daya yang dipengaruhi oleh aktivitas-aktivitas tersebut dan para pelaku aktivitas tersebut.
2. Mengubah data menjadi informasi yang berguna bagi manajemen.
3. Menyediakan pengendalian yang memadai.

Sistem informasi akuntansi merupakan pendukung aktivitas organisasi, yang termasuk pendukung aktivitas organisasi adalah:

1. Infrastruktur perusahaan : akuntansi, hukum, administrasi umum.
2. Sumber daya manusia : perekrutan, pengontrolan, pelatihan dan kompensasi kepada pegawai.
3. Teknologi : peningkatan produk dan jasa (penelitian).
4. Pembelian.

Sementara itu aktivitas utamanya adalah:

1. *Inbound Logistics* : Penerimaan, penyimpanan dan distribusi bahan-bahan masukan.
2. Operasi : aktivitas untuk mengubah masukan menjadi barang atau jasa.
3. *Outbound Logistics* : distribusi produk ke pelanggan.
4. Pemasaran dan Penjualan.
5. Pelayanan : Dukungan purna jual dan *maintenance*.

Dari penjelasan tersebut dapat diambil simpulan bahwa akuntansi adalah proses pengidentifikasian, pengukuran dan melaporkan informasi ekonomi untuk memungkinkan adanya pembuatan pertimbangan dan keputusan bagi yang menggunakan informasi.

Kutipan diatas menjelaskan bahwa sistem informasi akuntansi adalah kumpulan dari sumber-sumber seperti orang dan peralatan yang dirancang untuk mentransformasikan data keuangan dan data lainnya menjadi informasi, dan informasi ini akan dikomunikasikan kepada para pembuat keputusan.

Manfaat Sistem Informasi Akuntansi tersebut adalah sebagai berikut :

1. Mengamankan harta/kekayaan perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan.
3. Menghasilkan informasi untuk pihak eksternal
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan)
6. Menghasilkan informasi untuk penyusunan data evaluasi anggaran perusahaan.
7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian.

II.5. Sekilas Tentang Pengertian Laba

Menurut M. Nafarin (2007 : 788), Laba (*income*) adalah perbedaan antara pendapatan dengan keseimbangan biaya-biaya dan pengeluaran untuk periode tertentu, laba dapat dibedakan menjadi laba akuntan dengan laba ekonom, dan laba akunting dengan laba ekonomi.

Laba akunting merupakan titik perhatian utama dalam laporan keuangan. laba mempunyai arti penting sebagai dasar pengenaan pajak, penentu *dividen*, dan penentu kebijakan infestasi. Sampai saat ini masalah laba masih menjadi salah satu titik kritik kuat terhadap akunting. Apakah laba akunting mampu mencerminkan laba yang semestinya ?. Dua ekonom keuangan, seperti Merton H. Miller (1958) dan franco Modigliani (1961) berpendapat bahwa laba ekonomi merupakan sumber penciptaan nilai (*value creation*) di perusahaan dan bahwa

tingkat imbalan atau pengembalian (*return*) ditentukan berdasarkan tingkat risiko yang diasumsikan oleh pemodal (*investor*).

II.5.1. Laba Akuntan dan Laba Ekonom

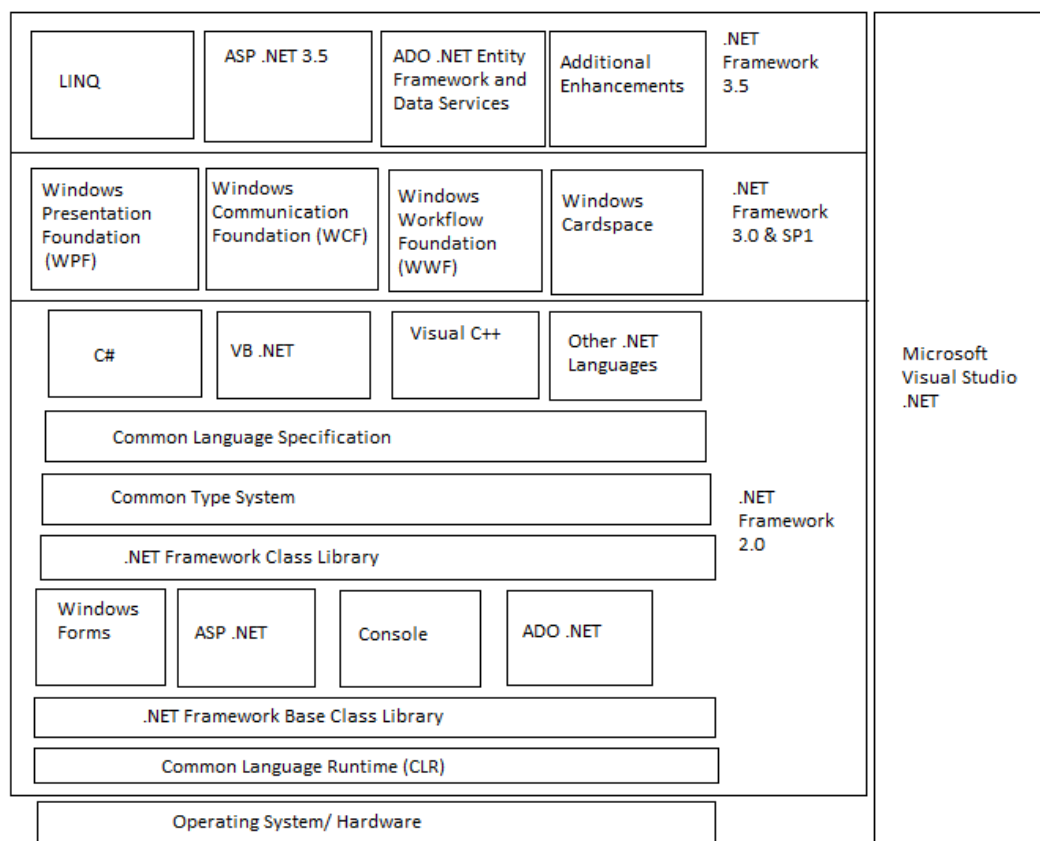
Menurut M. Nafarin (2007 : 788), Pengertian laba akuntan disini adalah definisi laba menurut para akuntan (ahli akunting) berpendapat bahwa laba adalah kelebihan dapatan (*revenues*) terhadap beban (*expenses*). Dalam bahasa akuntan, hal yang menambah modal atau jumlah yang dapat dikonsumsi seseorang dalam suatu periode tanpa memengaruhi keadaan tersebut pada akhir periode dibandingkan keadaan pada awal periode disebut "untung". Sedangkan hal yang mengurangi modal atau jumlah yang dikonsumsi mengganggu (mengurangi) modal atau jumlah yang dikonsumsi seseorang dalam suatu periode yang mengurangi keadaan orang tersebut pada akhir periode dibandingkan keadaannya pada awal periode disebut dengan "rugi", Sementara apabila jumlah yang dikonsumsi seseorang dalam suatu periode mengakibatkan keadaan orang tersebut pada awal periode sama dengan pada akhir periode disebut "impas".

Pengertian laba ekonom di sini adalah definisi laba menurut para ekonom (ahli ekonom). Adam Smith berpendapat bahwa laba adalah jumlah yang dapat dikonsumsi tanpa mengganggu modal, dan Hick berpendapat bahwa laba adalah jumlah yang dapat dikonsumsi seseorang dalam suatu periode tanpa memengaruhi keadaan orang tersebut.

II.6. Microsoft Visual Studio.Net

Microsoft Visual Studio .NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *.NET Framework*, dengan menggunakan bahasa *BASIC*. *Visual Basic* terkenal sebagai bahasa pemrograman yang mudah untuk digunakan terutama untuk membuat aplikasi yang berjalan di atas *platform windows*. (Erick Kurniawan, 2011:1)

Gambar II.1 berikut ini menerangkan arsitektur dari *Microsoft Visual Studio.Net*



Gambar II.1. Arsitektur Microsoft Visual Studio.Net

(Sumber : Prabowo Pudjo Widodo & Herlawati ; 2011:32)

Seperti yang telah ditunjukkan dalam gambar II.1, Visual Studio terdiri dari

1. CLR (*Common Language Runtime*)

Adalah pondasi utama dari *framework .NET*, yang bertanggung jawab terhadap berbagai macam hal, seperti melakukan manajemen *memory*, melakukan eksekusi kode, melakukan verifikasi terhadap keamanan kode, menentukan hak akses dari kode, melakukan kompilasi kode, dan berbagai layanan sistem lainnya.

2. FCL (*.NET framework Class Library*) atau BCL (*Base Class Library*)

Adalah koleksi dari *reusable types* yang sangat banyak dan terintegrasi secara melekat dengan CLR. Kumpulan *Class Library* ini sangat berguna untuk pengembangan aplikasi karena pengembangan tidak perlu membuat semuanya dari awal karena telah disediakan oleh *.NET*, misalnya *class* untuk membuat aplikasi berbasis *windows*

Microsoft menciptakan *.NET Framework* dengan beberapa tujuan, antara lain :

1. Lingkungan pengembangan program multibahasa : *.NET Framework* mendukung beberapa bahasa pemrograman seperti *VB*, *C#*, *C++*, dan lain-lain.
2. Mendukung penuh konsep *object oriented programming*, semua yang ada di *.NET Framework* adalah *object*.
3. Menyederhanakan pemrograman pada *platform windows*.
4. Menggunakan *managed code* sehingga penulisan kode menjadi lebih aman.

II.7. Microsoft SQL Server

Menurut Harip Santoso (2006 : 5) Sejarah *SQL Server* berbeda jauh dengan sejarah *Visual Basic*. Bila *Visual Basic* berasal dari pengembangan *QuickBasic* yang juga merupakan produk *Microsoft*, maka *SQL Server* adalah hasil kerja sama antara *Microsoft* dengan *Sybase* untuk memproduksi sebuah software penyimpanan data (database) yang bekerja pada sistem operasi OS/2.

Sistem Operasi OS/2 merupakan sistem operasi baru dari hasil kerja sama antara *Microsoft* dengan *IBM*. Sistem operasi OS/2 mengenal bentuk-bentuk perintah DOS, sekaligus memiliki kemampuan *multitasking*. Untuk mendapat pengakuan pasar, maka *Microsoft* bekerja sama dengan *Ashton-Tate* yang telah dikenal dengan produk *dBase*. Sayangnya, kerja sama ini tidak berlangsung lama, karena setelah peluncuran *SQL Server* versi 1.0 pada tahun 1989 kerja sama itupun berakhir.

Peluncuran *SQL Server* 1.0 dilanjutkan dengan peluncuran *SQL Server* versi 1.1 pada tahun 1990. Adapun fitur terpenting dari produk ini adalah dukungan untuk *platform* baru dari sisi client yang dikenal sebagai sistem operasi *Windows 3.1*.

Pada tahun 1991, *SQL Server* versi 1.11 diluncurkan dan berisi fasilitas perawatan database. Pada tahun yang sama, *Microsoft* mengembangkan sendiri sistem operasi *multi user* yang dikenal sebagai *Windows NT*, *Microsoft SQL Server* versi 4.2 dirilis pada tahun 1992 dan berisi tool administrator database berbasis *GUI-Windows*.

II.8. Database

Menurut Abdul Kadir (2009:10) terdapat butir-butir yang bisa diambil dari definisi *database* yaitu :

1. Sebuah *database* menghimpun data yang terkait atau data yang saling berhubungan.
2. Kumpulan data tersebut terorganisasi.
3. Bisa melibatkan lebih dari satu organisasi.

Butir ketiga mengisyaratkan bahwa sebuah database bisa digunakan oleh lebih dari satu organisasi. Sebuah *database* mencatat berbagai data yang diperlukan oleh suatu organisasi, rekaman-rekaman data tersebut pada suatu saat akan diambil dan melalui suatu pemrosesan akan diperoleh suatu informasi yang dikehendaki oleh pengguna.

Database berbeda dengan sistem pemrosesan berbasis berkas. Sistem pemrosesan berbasis berkas adalah suatu model penyimpanan data yang mendasarkan pada penyimpanan data dalam bentuk *file* (berkas), yang memiliki banyak kelemahan dibanding database.sistem seperti ini banyak dipakai dimasa lalu, salah satu perangkat lunak yang biasa dipakai untuk mengimplementasikannya adalah *COBOL*.

II.8.1. Pemodelan Data

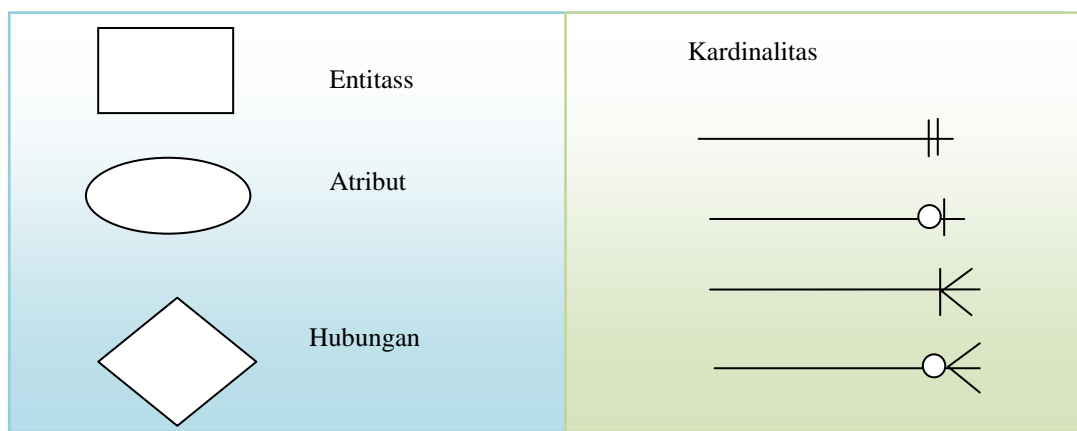
Menurut Abdul Kadir (2009:30) Pada perancangan konseptual diperlukan suatu pendekatan yang digunakan untuk menggambarkan hubungan antar data. Hubungan tersebut dapat dinyatakan dalam bentuk E-R, karena model E-R adalah dasar penting dalam merancang database maka akan dijelaskan tentang gambaran

tentang model E-R, penjelasan mengenai komponen-komponen yang menyusun model E-R, hingga cara penyusunan model E-R.

II.8.2. Model E-R

Menurut Abdul Kadir (2009:10) Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antar entitas. Huruf E sendiri menyatakan entitas dan R menyatakan hubungan (dari kata *Relationship*). Model ini dinyatakan dalam bentuk diagram, itulah sebabnya model E-R sering disebut sebagai diagram E-R.

Model E-R melibatkan sejumlah notasi, beberapa notasi dasar dalam model E-R ditunjukkan pada gambar II.2., notasi-notasi tersebut diberikan hanya untuk memberikan suatu pengetahuan dasar.



Gambar II.2. : Sejumlah notasi pada model E-R

Sumber : Abdul Kadir (2009:31)

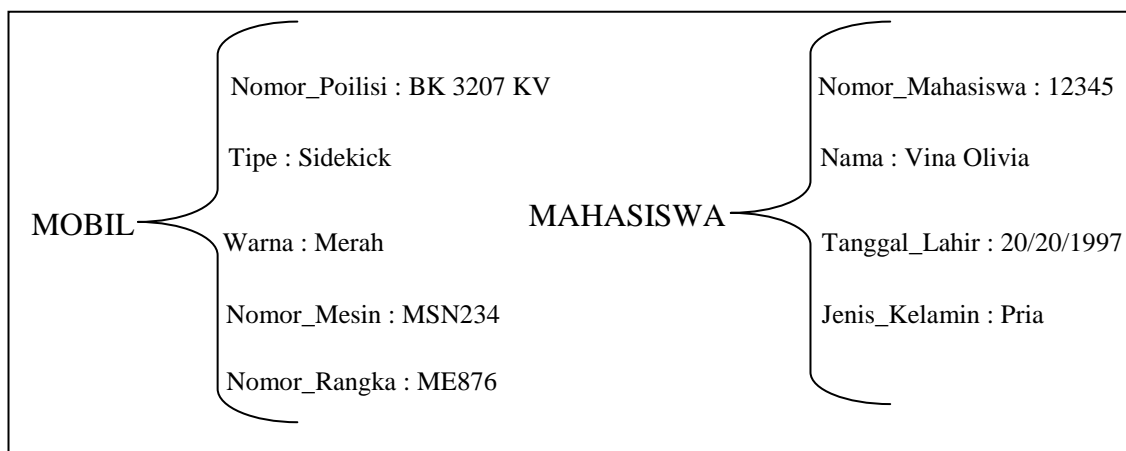
1. Entitas

Yang dimaksud dengan entitas adalah sesuatu dalam dunia nyata yang keberadaannya tidak bergantung pada yang lain. Sebagai contoh, setiap pegawai dalam sebuah organisasi adalah sebuah entitas. Entitas dapat berupa

sesuat yang nyata ataupun abstrak (berupa suatu konsep). Secara lebih rinci dijelaskan bahwa entitas dapat berupa seseorang, sebuah tempat, sebuah objek, sebuah kejadian atau suatu konsep.

2. Atribut

Setiap entitas dinyatakan dalam sejumlah atribut. Atribut adalah properti atau karakteristik yang terdapat pada setiap entitas. Sebagai contoh, pada gambar II.3., terdapat entitas MOBIL yang mengandung atribut Nomor_Polisi, Tipe, Warna, Nomor_Rangka dan Nomor_Mesin. Selain itu terdapat entitas MAHASISWA yang mengandung atribut Nomor_Mahasiswa, Nama, Tanggal_Lahir, dan Jenis_Kelamin.



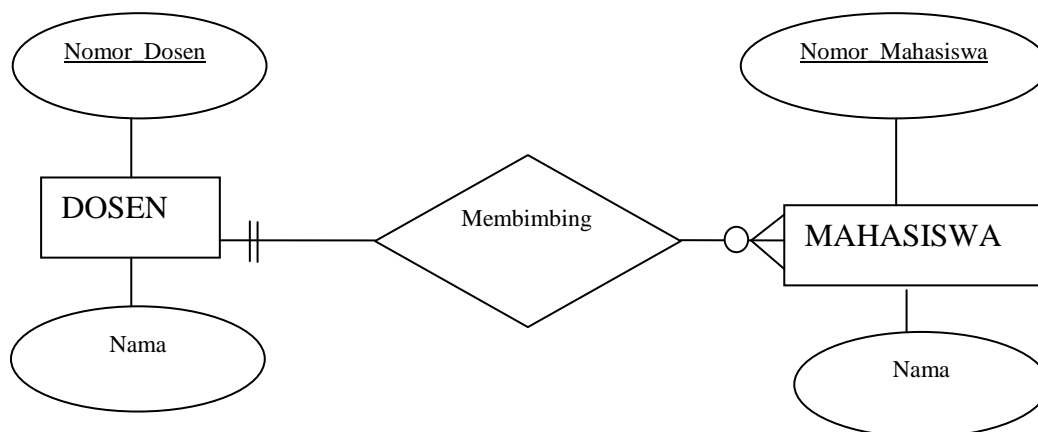
Gambar II.3 : Contoh Entitas dan Atribut

Sumber : Abdul Kadir (2009:32)

3. Hubungan (*Relationship*)

Hubungan (*Relationship*) menyatakan ketertarikan antara beberapa tipe entitas. Sebagai contoh , tipe entitas MAHASISWA dan DOSEN mempunyai

hubungan yang mencerminkan bahwa seorang mahasiswa memiliki dosen pembimbing akademis. Gambar II.4. menunjukkan hubungan tersebut.



Gambar II.4. : Contoh Hubungan Antara Tipe Entitas

Sumber : Abdul Kadir (2009:45)

II.8.3. Normalisasi

Normalisasi adalah proses yang digunakan untuk menentukan pengelompokan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang berstruktur baik. Dalam hal ini yang dimaksud dengan relasi yang berstruktur baik adalah relasi yang memenuhi dua kondisi berikut

1. Mengandung redundansi sedikit mungkin, dan
2. Memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi dan dihapus tanpa menimbulkan kesalahan atau ketidakkonsistenan.

Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. Dalam hal ini yang disebut bentuk normal adalah suatu keadaan relasi yang dihasilkan oleh penerapan aturan-aturan sederhana yang berhubungan dengan dependensi fungsional

terhadap relasi tersebut. Gambar II.5. berikut ini akan memperlihatkan hubungan keenam bentuk normal tersebut.



Gambar II.5. : Langkah-Langkah Dalam Normalisasi

Sumber : Abdul Kadir (2009:118)

II.9. Unified Modeling Language (UML)

UML Adalah perkakas untuk analisis dan perancangan yang sesungguhnya digunakan untuk pederhanaan permasalahan. UML merupakan metodologi kolaborasi antara metoda-metoda *Booch*, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Engineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering di gunakan pada saat ini untuk mengadaptasi maraknya penggunaan bahasa pemrograman berorientasi objek.

Menurut Prabowo Pudjo Widodo & Herlawati (2011 : 6) UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

1. Merancang perangkat Lunak.
2. Sarana Komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

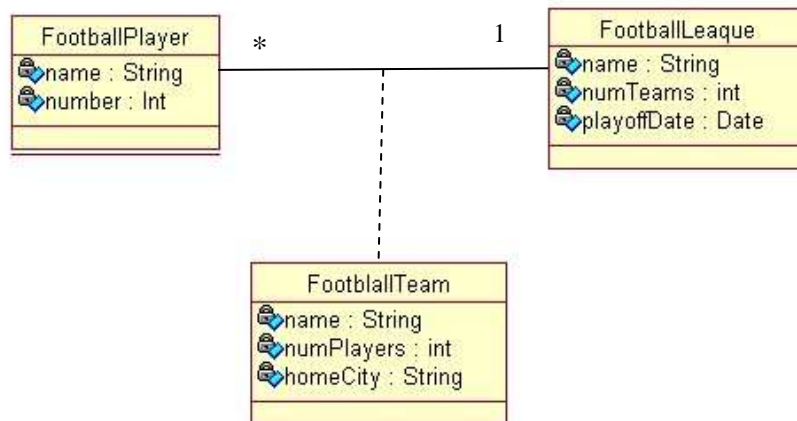
Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dengan mengaplikasikan beragam sistem. Intinya UML merupakan alat komunikasi yang konsisten dalam mendukung para pengembang sistem saat ini.

II.9.1 Diagram-Diagram UML

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, model-model ini dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. ***Class Diagram*** : Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif. Terkadang hubungan antara dua elemen tidak sederhana misalnya suatu tim pemain bola (*football player*) berasosiasi dengan liga (*league*) lewat suatu

regu. Jika hubungannya terlalu rumit, bisa dibuatkan hubungan asosiasi antar kelas. Suatu asosiasi kelas memiliki nama dan atribut seperti kelas biasa, notasi untuk kelas asosiasi adalah dengan garis putus-putus mengenai garis asosiasi utama.

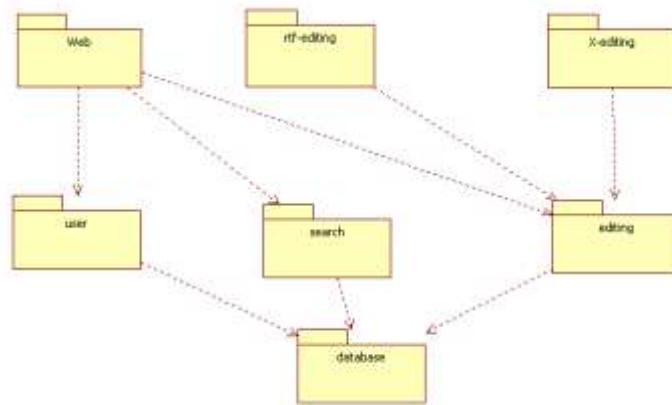


Gambar II.6. : Diagram Kelas Asosiasi

Sumber : Prabowo Pudjo Widodo & Herlawati (2011 : 69)

Ketika ditranslasikan menjadi kode, biasanya kelas asosiasi dianggap sebagai kelas bias, sehingga ada tiga kelas yang terbentuk, yang perlu diperhatikan dari gambar II.5 di atas adalah bahwa *FootballPlayer* tidak memiliki referensi langsung kepada *FootballLeague* tapi memiliki referensi terhadap *FootballTeam*. *FootballTeam* akan memiliki referensi terhadap *FootballLeague*.

2. **Package Diagram** : Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.



Gambar II.7. : Diagram Paket

Sumber : Prabowo Pudjo Widodo & Herlawati (2011 : 88)

Gambar di atas memiliki sedikit informasi, oleh karena itu perlu ditambahkan suatu dokumen tertulis yang menjelaskan dasar-dasar paket itu. Dokumen itu mungkin berisi daftar sebagai berikut :

a. Web

Dibutuhkan keterampilan khusus : *HTML*, *css* dan *Struts*, teknologi presentasi. Seluruhnya memiliki ketergantungan.

b. Database

Dibutuhkan suatu manajemen *database* dan keterampilan membuat model. Memiliki sifat bebas atau kurang tergantung.

c. User

Dibangun diluar lokais (*remote team*).

d. Search

Dibutuhkan suatu kemampuan dengan dengan teknologi *search engine*.

Bersifat *subsistem* lokal.

e. Editing

Terdiri dari dasar *fitur editing* untuk penerbitan pertama.

f. *Rtf-editing*

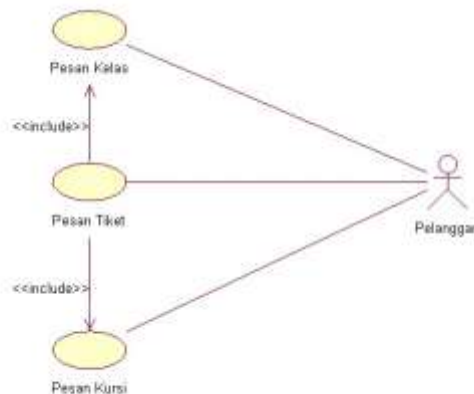
Terdiri dari dasar *fitur editing* untuk release berikutnya.

g. *X-editing*

Terdiri dari *fitur editing* atas permintaan klien khusus

Diagram paket pada gambar II.7. diatas tidak memperlihatkan paket yang komplit. Diagram tersebut hanya bermanfaat dari sisi manajemen, sementara itu *programer* membuat diagram paket level bawah agar dihasilkan kode program yang benar.

3. ***Use-Case Diagram*** : Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.



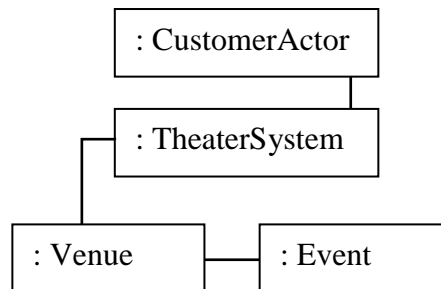
Gambar II.8. : Diagram Use Case

Sumber : Prabowo Pudjo Widodo & Herlawati (2011 : 30)

Gambar II.8. menggambarkan hubungan inklusi antara *use case* pesan tiket dengan *use case* pesan kelas dan pesan kursi. Pesan tiket disebut *use case* pemanggil (*calling use case*) sedangkan pesan kelas dan pesan kursi disebut

use case terpanggil (*called use case*). *Use case* pesan tiket belum lengkap karena harus pesan kelas dan pesan kursi terlebih dahulu.

4. **Sequence Diagram dan Interaksi** : Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. **Diagram Komunikasi (*Communication Diagram*)** : Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.



Gambar II.9. : Diagram Komunikasi

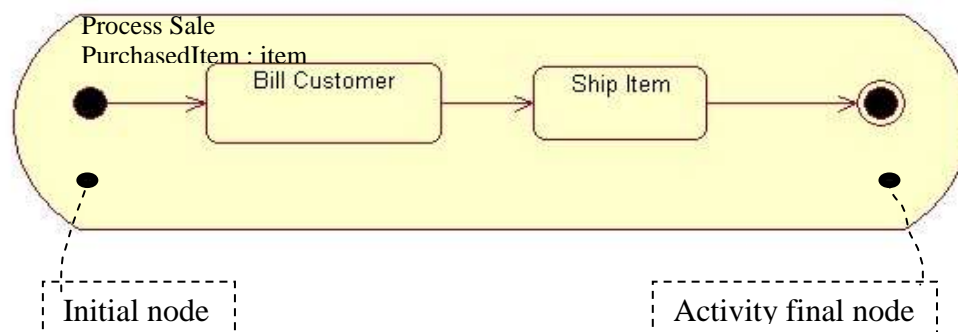
Sumber : Prabowo Pudjo Widodo & Herlawati (2011 : 196)

Diagram komunikasi dimaksudkan untuk melengkapi diagram urutan dengan memberikan tampilan visual pada pesan yang disampaikan antar objek. Dasar dari diagram komunikasi adalah diagram objek, seperti ditunjukkan pada gambar II.9. Tiap objek dalam diagram komunikasi disebut garis hidup objek (*object lifetime*).

6. **Diagram Statechart (*Statechart Diagram*)** : Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan

sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.

7. **Diagram Aktivitas (*Activity Diagram*)** : Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.



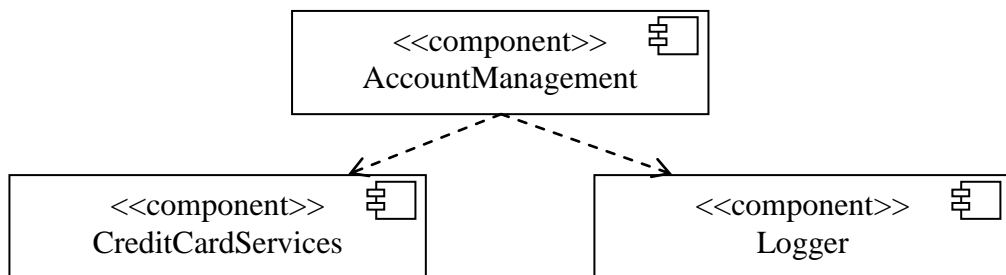
Gambar II.10. : Diagram Aktivitas

Sumber : Prabowo Pudjo Widodo & Herlawati (2011 : 146)

Seperti yang terlihat pada gambar II.10. di atas, tiap aktivitas dimulai dengan titik awal (*initial node*) dan diakhiri dengan aktivitas titik akhir (*final node*). Saat ini aktivitas mencapai aktivitas titik akhir, aktivitas itu terhenti. Titik awal digambarkan dengan titik hitam dan titik akhir dengan lingkaran tebal yang didalamnya terdapat titik hitam.

8. **Diagram Komponen (*Component Diagram*)** : Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta ketergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana

komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.

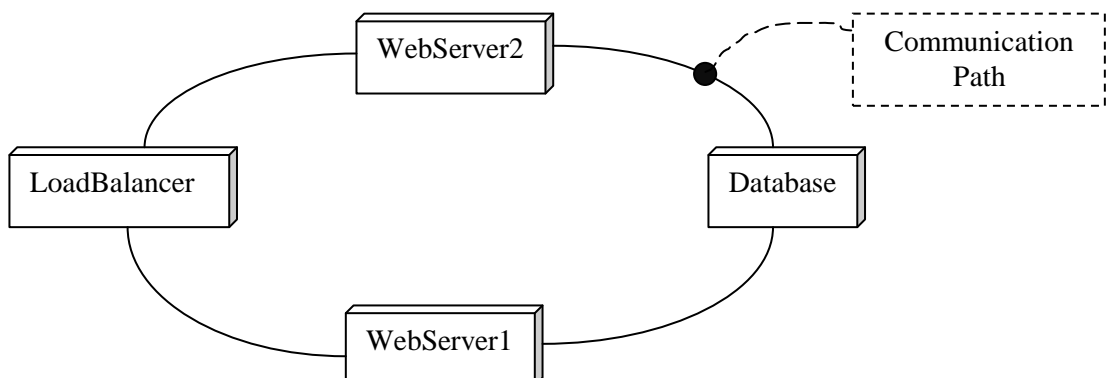


Gambar II.11 : Diagram Komponen

Sumber : Prabowo Pudjo Widodo & Herlawati (2011 : 96)

Gambar II.11 menggambarkan komponen *AccountManagement* memiliki ketergantungan dengan kedua komponen lainnya. Penggambaran ketergantungan pada gambar di atas merupakan level tertinggi dan tidak merinci ketergantungan secara lebih besar

9. **Diagram *Deployment (Deployment Diagram)*** : Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen.



Gambar II.12. : Diagram *Deployment*

Sumber : Prabowo Pudjo Widodo & Herlawati (2011 : 117)

Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*), (prabowo Pudjo Widodo & Herlawati ; 2011: 6).