

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

II.1.1. Sistem

Menurut O'Brien (2006:29), sistem adalah sekelompok komponen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima *input* serta menghasilkan *output* dalam proses transformasi yang teratur. Sistem memiliki tiga fungsi dasar yang saling berinteraksi menurut O'Brien (2006:29):

1. *Input* melibatkan penangkapan dan perakitan berbagai elemen yang memasuki sistem untuk diproses.
2. Pemrosesan melibatkan proses transformasi yang mengubah *input* menjadi *output*.
3. *Output* melibatkan perpindahan elemen yang telah diproduksi oleh proses transformasi ke tujuan akhirnya.

Selain memiliki fungsi dasar, sistem mempunyai karakteristik. Gondodiyoto (2007:108) menyebutkan karakteristik dari sistem sebagai berikut:

1. Sistem adalah kumpulan elemen-elemen atau sumber daya yang saling berkaitan secara terpadu, terintegrasi dalam suatu hubungan hirarkis tertentu, dan bertujuan untuk mencapai tujuan tertentu.
2. Memiliki karakteristik khusus yang menjadi "jiwa atau roh" dari sistem tersebut.

3. Mempunyai sasaran yang akan dicapai.
4. Konstruksi sistem terdiri dari masukan, proses, keluaran.
5. Sistem memerlukan pengendalian. Pengendalian merupakan proses pengaturan yang dipergunakan sistem untuk mengoreksi setiap penyimpangan dari suatu rangkaian langkah untuk menuju sasaran.
6. Sistem memiliki pengguna. Setiap sistem harus mengarah sub sistemnya agar dapat mencapai sasaran. Sasaran sistem sebagai ukuran penentu keberhasilan suatu sistem.
7. Sistem mempunyai keterbatasan.
8. Terdiri dari subsistem-subsistem yang membentuk suatu jaringan terpadu. Setiap sistem terdiri dari lebih dari satu komponen yang saling terjalin satu sama lain disebut subsistem, yang menjalankan peran tertentu dan menjadi bagian di dalam sistem yang lebih besar. Dalam sistem, subsistem-subsistem tidak berdiri sendiri melainkan saling berinteraksi serta saling berhubungan membentuk satu kesatuan terpadu sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

II.1.2. Informasi

Husein dan Amin mendeskripsikan, informasi adalah data yang telah diolah menjadi suatu bentuk yang mempunyai arti dan bermanfaat bagi manusia. Sehingga informasi lebih dikenal sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna bagi penerimanya, dimana informasi tersebut menggambarkan tentang suatu kejadian nyata yang dapat dipahami

dan dapat digunakan untuk pengambilan keputusan sekarang maupun masa depan.

Dalam perusahaan, informasi menggambarkan tentang kinerja perusahaan. Ada dua tolok ukur untuk menggambarkan kinerja karyawan dalam perusahaan, yaitu:

1. Efisiensi

Efisiensi adalah penggunaan sumber daya secara minimum guna pencapaian hasil yang optimum (Wibisono, 2010).

Efisiensi menganggap bahwa tujuan-tujuan yang benar telah ditentukan dan berusaha untuk mencari cara-cara yang paling baik untuk mencapai tujuan-tujuan tersebut. Sehingga efisien lebih mengarah melakukan sesuatu dengan benar. Efisiensi hanya dapat dievaluasi dengan penilaian-penilaian relatif, membandingkan antara masukan dan keluaran yang diterima.

2. Efektifitas

Definisi efektif adalah suatu pencapaian tujuan secara tepat atau memilih tujuan-tujuan yang tepat dari serangkaian alternatif atau pilihan cara dan menentukan pilihan dari beberapa pilihan lainnya (Wibisono, 2010). Efektifitas lebih mengarah pada hasil akhir dari kegiatan yang berjalan dalam perusahaan. Jika hasil kegiatan yang dilakukan perusahaan semakin mendekati sasaran yang ditentukan perusahaan maka semakin tinggi efektifitasnya. Sehingga efektifitas dapat juga diartikan sebagai pengukur keberhasilan dalam pencapaian tujuan yang telah ditentukan.

Ada beberapa kriteria dari informasi menurut Gondodiyoto (2007:111), antara lain:

1. Akurat, *reliable* (dapat dipercaya). Informasi tersebut kecil kesalahannya sehingga tidak menyesatkan penggunaannya.
2. Relevan (cocok atau sesuai). Informasi harus memberikan arti kepada penggunaannya atau informasi tersebut mempunyai manfaat bagi pemakainya.
3. *Timely* (tepat waktu). Informasi disajikan tepat pada saat dibutuhkan dan bisa mempengaruhi proses pengambilan keputusan. Informasi datang kepada pengguna tidak boleh terlambat.
4. *Complete* (lengkap). Informasi yang disajikan semua data-datanya sesuai dengan kenyataan.
5. *Understandable* (dimengerti). Informasi yang disajikan dalam bentuk yang mudah dimengerti oleh pembuat keputusan.
6. *Verifiable*. Informasi yang dihasilkan tidak bias, sehingga dalam pemahamannya tidak menimbulkan kesalahpahaman.
7. *Accessible*. Informasi tersedia pada saat yang diperlukan dalam format sesuai dengan kepentingannya. Informasi dikatakan bernilai apabila manfaat dari informasi lebih efektif dibandingkan dengan biaya yang dikeluarkan untuk mendapatkannya.

II.1.3. Sistem Informasi

Sistem informasi dapat secara manual maupun terkomputerisasi. Menurut O'Brien (2006:5), sistem informasi merupakan kombinasi teratur apa pun dari orang-orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah

organisasi. Organisasi menyesuaikan sistem informasinya dengan kebutuhan pemakainya. Tiga tujuan utama bagi sistem informasi, yaitu:

1. Untuk mendukung fungsi kepengurusan manajemen. Sistem informasi menyediakan informasi tentang kegunaan sumber daya ke pemakai eksternal melalui laporan keuangan tradisional dan laporan-laporan yang diminta lainnya.
2. Untuk mendukung pengambilan keputusan manajemen. Sistem informasi memberikan para manajer informasi yang mereka perlukan untuk melakukan tanggung jawab pengambilan keputusan.

Untuk mendukung kegiatan operasi perusahaan hari demi hari. Sistem informasi menyediakan informasi bagi personel operasi untuk membantu mereka melakukan tugas mereka setiap hari dengan efisien dan efektif. Menurut Hall (2007:62) bagian perusahaan yang terlibat dalam pengembangan sistem adalah praktisi sistem, pengguna akhir, dan pemegang kepentingan dalam perusahaan.

II.2. Sistem Informasi Akuntansi

II.2.1. Akuntansi

Heri (2009 : 1) menjelaskan, ada beberapa kutipan pendapat dari ahli mengenai definisi akuntansi, yaitu sebagai berikut : Menurut **ABP Statement** No.4 (Tahun 1970) yang berjudul *“Basic Concepts and Accounting principles Underlying Financial Statement of Business Enterprises”* : “Akuntansi adalah sebuah aktivitas jasa, dimana fungsinya adalah memberikan informasi kuantitatif, terutama informasi mengenai keuangan dan entitas ekonomi yang dimaksudkan

akan menjadi berguna dalam pengambilan keputusan ekonomi (dalam membuat pilihan di antara berbagai alternative yang ada)”. Sedangkan menurut *American Institute of Certified Public Accountants (AICPA)*, akuntansi adalah : “seni pencatatan, pengklasifikasian, dan pengikhtisaran transaksi dan peristiwa keuangan dengan cara tertentu dan dalam ukuran moneter termasuk penafsiran atas hasil-hasilnya” (Fitri Sukmawati : 2011 : 5).

II.2.2. Sistem Informasi Akutansi

Menurut Anastasia Diana dan Lilis Setiawati (2011 : 4) Sistem informasi akutansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan.

Menurut Jogiyanto, (2007:17) SIA adalah kumpulan kegiatan-kegiatan dari organisasi yang bertanggung-jawab untuk menyediakan informasi keuangan dan informasi yang didapatkan dari transaksi data untuk tujuan pelaporan internal kepada manajer untuk digunakan dalam pengendalian dan perencanaan sekarang dan operasi masa depan serta pelaporan eksternal kepada pemegang saham, pemerintah, dan pihak-pihak luar lainnya. Dari definisi tersebut dapat diambil kesimpulan bahwa SIA merupakan suatu kegiatan *input*, proses, dan *output* data yang dilakukan oleh perusahaan. Hasil data akhir yang telah di proses SIA bertujuan sebagai pelaporan bagi pihak internal dan eksternal guna melakukan pengendalian terhadap perusahaan tersebut.

II.3. Aktiva Tetap Berwujud

II.3.1. Pengertian Aktiva Tetap Berwujud

. Menurut Zaki Baridwan (2007 : 271) yang di maksud dengan aktiva tetap berwujud adalah aktiva-aktiva yang berwujud yang sifatnya relatif permanen yang digunakan dalam kegiatan perusahaan yang normal. Istilah relatif permanen menunjukkan sifat dimana aktiva yang bersangkutan dapat digunakan dalam jangka waktu yang relatif cukup lama. Untuk akuntansi, jangka waktu penggunaan ini dibatasi dengan “lebih dari satu periode akuntansi”. Jadi aktiva tetap berwujud yang umurnya lebih dari satu periode akuntansi dikelompokkan sebagai aktiva tetap berwujud.

II.3.2. Prinsip Penilaian Aktiva Tetap Berwujud

Menurut Zaki Baridwan (2007 : 273) dalam hubungannya dengan penilaian aktiva tetap berwujud , prinsip akuntansi indonesia menyatakan: ”Aktiva tetap dinyatakan sebesar nilai buku yaitu harga perolehan aktiva tetap tersebut dikurangi dengan akumulasi penyusutannya”.

Yang dimaksud dengan harga perolehan aktiva tetap adalah jumlah uang yang dikeluarkan atau utang yang timbul untuk memperoleh aktiva tetap tersebut. Jika aktiva tetap diperoleh dari pertukaran maka harga pasar aktiva yang diserahkan dipakai sebagai ukuran harga perolehan aktiva yang diterima. Apabila harga pasar aktiva yang diserahkan tidak diketahui, maka harga pasar aktiva yang diterima dicatat sebagai harga perolehan aktiva tetap tersebut berwujud.

II.4. Deprisasi

II.4.1. Pengertian Deprisasi

Menurut Zaki Baridwan (2007 : 307) deprisasi adalah sebagian dari harga perolehan aktiva tetap yang secara sistematis dialokasikan menjadi biaya setiap periode akuntansi. *Comitte on terminology* dari AICPA memberikan defenisi sebagi berikut:

Akuntansi depresiasi adalah suatu sistem akuntansi yang bertujuan untuk membagikan harga perolehan atau nilai dasar lain dari aktiva tetap berwujud, dikurangi nilai sisa (jika ada), selama umur kegunaan unit itu yang ditaksir (mungkin berupa suatu kumpulan aktiva-aktiva) dalam suatu cara yang sistematis dan rasional. Ini merupakan proses alokasi, bukan penilaian. Beban depresiasi untuk suatu tahun adalah sebagian dari jumlah total beban itu dengan sistem tersebut di alokasikan ke tahun yang bersangkutan. Meskipun di dalam alokasi itu diperhitungkan hal-hal yang terjadi selama tahun itu, tidak dimasukkan sebagai suatu alat pengukur terhadap akibat-akibat dari kejadian-kejadian itu berwujud.

Terdapat tiga faktor yang perlu dipertimbangkan dalam menentukan beban deprisasi setiap periode, yaitu :

1. Harga Perolehan

Yaitu uang yang dikeluarkan atau utang yang timbul dan biaya-biaya lain yang terjadi dalam memperoleh suatu aktiva dan menempatkannya agar dapat digunakan.

2. Nilai Sisa (residu)

Nilai sisa suatu aktiva yang didepresiasi adalah jumlah yang diterima bila aktiva itu dijual, ditukarkan atau cara-cara lain ketika aktiva tersebut sudah tidak dapat digunakan lagi, dikurangi dengan biaya-biaya yang terjadi pada saat menjual/menukarkannya.

3. Taksiran Umur Kegunaan

Taksiran umur kegunaan suatu aktiva dipengaruhi oleh cara-cara pemeliharaan dan kebijaksanaan-kebijaksanaan yang dianut dalam reparasi. Taksiran umur ini bisa dinyatakan dalam satuan periode waktu, satuan hasil produksi, atau jam kerjanya. Dalam menaksir umur aktiva, harus dipertimbangkan sebab-sebab keausan fisik dan fungsional berwujud.

II.4.2. Metode Perhitungan Depresiasi

Menurut Zaki Baridwan (2007 : 311-318) ada beberapa metode yang dapat digunakan untuk menghitung beban depresiasi periodik. Untuk dapat memilih salah satu metode hendaknya dipertimbangkan keadaan-keadaan yang mempengaruhi aktiva tetap tersebut. Metode–metode itu adalah:

1. Metode Garis Lurus (*Straight Line Method*)

Metode ini adalah metode depresiasi yang paling sederhana dan banyak digunakan. Dalam cara ini beban depresiasi tiap periode jumlahnya sama.

$\text{Depresiasi} = \frac{\text{HP} - \text{NS}}{n}$

Keterangan :

HP : Harga perolehan

NS : Nilai Residu (residu)

N : Taksiran umur kegunaan

Perhitungan depresiasi dengan garis lurus ini didasarkan pada anggapan-anggapan sebagai berikut:

1. Kegunaan ekonomis dari suatu aktiva akan menurun secara proporsional setiap periodenya.
2. Biaya reparasi dan pemeliharaan tiap-tiap periode relatif tetap.
3. Kegunaan ekonomis berkurang karena lewatnya waktu.
4. Penggunaan (kapasitas) aktiva tiap-tiap periode relatif tetap.

Dengan adanya anggapan-anggapan seperti diatas, metode garis lurus sebaiknya digunakan untuk menghitung depresiasi gedung, mebel dan alat-lat kantor. Biaya depresiasi yang dihitung dengan cara ini jumlahnya setiap periodenya tetap, tidak menghiraukan kegiatan dalam periode tersebut.

2. Metode Jam Kerja (*Service Hour Method*)

Metode ini didasarkan pada anggapan bahwa aktiva akan lebih cepat rusak apabila digunakan sepenuhnya (*full time*) dibanding dengan penggunaan yang tidak sepenuhnya (*part time*). Dalam cara ini beban depresiasi dihitung

dengan dasar satuan jam kerja. Beban depresiasi periodik besarnya akan sangat bergantung pada jam yang dipakai.

$$\begin{array}{l} \text{Depresiasi per jam} \\ \text{NS} \end{array} = \frac{\text{HP}}{n} = \text{HP} - \text{Akumulasi Penyusutan}$$

HP : Harga perolehan

NS : Nilai Sisa

N : Taksiran jam kerja

3. Metode Hasil Produksi (*Productive Output Method*)

dalam metode ini umur kegunaan aktiva ditaksir dalam satuan jumlah unit produksi. Beban depresiasi dihitung dengan dasar satuan hasil produksi, sehingga depresiasi tiap periode akan fluktuasi dalam hasil produksi. Dasar teori yang dipakai adalah bahwa suatu aktiva itu dimiliki untuk menghasilkan produk, sehingga depresiasi juga didasarkan pada jumlah produk yang dapat dihasilkan.

$$\text{Deprisiasi/unit} = \frac{\text{HP} - \text{NS}}{n}$$

Keterangan :

HP : Harga perolehan

NS : Nilai Sisa

N : Taksiran hasil produksi (unit)

4. Metode Beban Berkurang (*Reducing Charge Method*):
 - a. Metode Jumlah angka tahun (*Sun Of Years Digits Method*)
 - b. Metode Saldo Menurun (*Declining Balance Method*)
 - c. Metode Saldo Menurun Ganda (*Double Declining Balance Method*)
 - d. Metode Tarif Menurun (*Declining Rate on Cost Method*)

II.5. Visual Basic 2010

Menurut Wahana Komputer (2011 : 1-2) *Visual Basic 2010* merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *microsoft*, yaitu *Microsoft visual Basic 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*. Adapter *database* standar yang disertakan adalah *Microsoft SQL Server 2008 express*.

Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic 2008*. Beberapa pengembangan yang terdapat di dalamnya 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 di kalangan *programmer* komputer, yaitu bahasa *BASIC*, yang oleh *Microsoft* di adaptasi 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 di dalamnya termasuk *Microsoft Visual Basic 2010*.

II.6. Database

Menurut Wahana Komputer (2010:24) *Database* atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. Data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. *Database* sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi tertentu.

II.7. SQL

Menurut Wahana Komputer (2010 : 2-3) *SQL Server 2008* adalah terobosan baru dari *microsoft* dalam bidang database. *SQL Server* adalah sebuah 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.

1. Kebutuhan *Hardware*

Adapun *hardware* minimal yang dibutuhkan adalah sebagai berikut:

- *Processor* $\geq 1\text{GHz}$
- *Memory* $\geq 512\text{ MB}$
- *OS* = *Windows XP 32 bit-Windows 7*

Sedangkan untuk jaringannya yang diperlukan adalah:

- *Shared memory*
- *TCP/IP*
- *Named Pipes*
- *VIA (Virtual Interface Adapter)*

2. Versi-versi *SQL Server 2008*

Microsoft merilis *SQL Server 2008* dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sbb:

- Versi 32-bit (x86), yang biasanya digunakan untuk komputer dengan *single processor 32 bit* dan sistem operasi windows XP .data

- Versi 64-bit (x64), yang biasanya digunakan untuk komputer dengan lebih dari satu *processor* (Misalnya: *Core 2 Duo*) dan *system* operasi 64 bit seperti Windows Xp 64, Vista, dan Windows 7.

II.8. Normalisasi

Menurut Janner Simarmata dan imam paryudi (2006:79-82)

Normalisasi adalah suatu 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 redudansi sesedikit mungkin, dan
2. Memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi, dan dihapus tanpa menimbulkan kesalahan atau ketidak konsistenan.

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” (Hoffer, dkk ; 2005). Yang dimaksud dengan aturan-aturan tersebut dan juga istilah dependensi fungsional akan dibahas belakangan. Berikut tahapan normalisasi :

1. Bentuk Tidak Normal (*Unnormalized Form*)

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

2. Bentuk Normal Pertama (1NF)

Bentuk normal kesatu mempunyai ciri setiap data dibentuk dalam *file* (*file* datar/rata), data dibentuk dalam satu *record* demi *record* dan nilai dari *field* berupa "atomic value". Tidak ada set atribut yang berulang atau atribut bernilai ganda (*multivalued*). Tiap *field* hanya satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti mendua, hanya satu arti saja dan juga bukan pecahan kata sehingga artinya lain. Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi, maka ia tidak memiliki sifat induknya. Contoh yang digunakan disini adalah sebuah perusahaan yang mendapatkan barang dari pemasok yang berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status sendiri. Masing-masing pemasok bisa menyediakan banyak barang. Tabel relasionalnya dapat ditulis sebagai berikut:

PEMASOK (p#, status, kota, b#, qty) dimana

P# : kode pemasok (kunci utama)

Status : kode status kota

Kota : nama kota

b# : barang yang dipasok

qty : jumlah barang yang dipasok

Agar bisa digabungkan jumlah barang yang dipasok (qty) secara unik dengan barang (b#) dan pemasok (p#), kita menggunakan kunci utama gabungan yang tersusun dari b# dan p#.

Sebuah tabel relasional secara definisi selalu berada dalam bentuk normal pertama. Semua nilai pada kolom-kolomnya adalah atomik. Ini berarti kolom-kolom tidak mempunyai nilai berulang.

Meskipun berada pada 1NF, tabel pemasok mengandung data berulang. Sebagai contoh, informasi tentang lokasi pemasok dan status lokasi harus diulang untuk setiap barang yang dipasok. Perulangan menyebabkan apa yang disebut *update anomalies*. *Update anomalies* adalah masalah yang timbul ketika informasi ditambahkan, dihapus, atau di-update.

3. Bentuk Normal Kedua (2NF)

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak ada pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama. Proses mengubah tabel 1NF menjadi 2NF adalah:

1. Tentukan senbarang kolom penentu selain kunci gabungan dan kolom-kolom yang ditentukannya.
 2. Buat dan beri tabel baru untuk masing-masing penentu dan kolom-kolom yang ditentukannya.
 3. Pindahkan kolom-kolom yang ditentukan dari tabel asal ke tabel baru. Penentu akan menjadi kunci utama pada tabel baru.
 4. Hapus kolom yang baru dipindahkan dari tabel asal, kecuali penentu yang berfungsi sebagai kunci tamu.
 5. Tabel asal bisa diberi nama baru.
4. Bentuk Normal Ketiga (3NF)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom bukan kunci tidak tergantung secara transitif dengan kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama.

5. Bentuk Normal *Boyce Codd* (BCNF/ *Boyce Codd Normal Form*)

Boyce Codd Normal Form mempunyai paksaan yang lebih kuat dari pada bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribut superkey.

II.9. *Entity Relationship Diagram (ERD)*

Menurut Janner Simarmata dan Imam Paryudi (2006 : 67) ERD adalah sebuah pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek kedalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur basis data yang baik sehingga data dapat disimpan dan diambil secara efisien. Pada dasarnya ada 3 macam simbol yang digunakan, yaitu :

1. Entitas (*Entity*)

Entitas adalah sesuatu yang nyata atau abstrak dimana kita akan menyimpan data. Ada 4 kelas entitas yaitu misalnya pegawai, pembayaran, kampus, dan buku. Contoh suatu entitas disebut instansi, misalnya pegawai Adi, pembayaran joko, dan lain sebagainya.

2. Atribut

Atribut adalah ciri umum semua atau sebagian besar instansi pada entitas tersebut. Sebutan lain atribut adalah properti, elemen data, dan field. Misalnya nama, alamat, nomor pegawai, dan gaji adalah atribut entitas pegawai. Sebuah atribut atau kombinasi atribut yang mengidentifikasi satu dan hanya satu instansi suatu entitas disebut kunci utama atau pengenal. Misalnya nomor pegawai adalah kunci utama untuk pegawai.

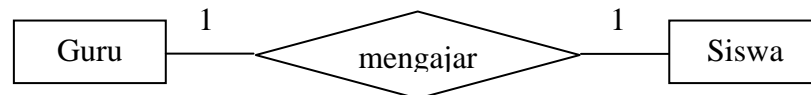
3. Relasi (*relationship*)

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai. Kardinalitas menentukan kejadian

suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya mahasiswa bisa mengambil banyak mata kuliah.

Jenis-jenis hubungan:

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.

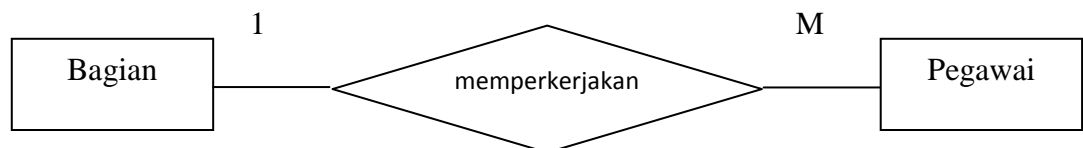


Gambar II.1. Relational 1 to 1

Sumber : (Janner Simarmata & Imam Paryudi ; 2006 : 63-64)

2. *One to Many*

Sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan paling banyak satu entitas pada A.



Gambar II.2. Relational One to Many

Sumber : (Janner Simarmata & Imam Paryudi ; 2006 : 64-65)

3. *Many to One*

Sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A.



Gambar II.3. Relational *Many to One*
Sumber : (Janner Simarmata & Imam Paryudi ; 2006 : 65)

II.10. Kamus Data

Menurut Janner Simarmata & Imam Paryudi ; 2006 Model berikutnya yang akan dibahas adalah data dictionary/DD (kamus Data/KD). KD tidak menggunakan notasi grafis sebagaimana halnya DAD, tetapi porsinya dalam memodelkan sistem tidak perlu diragukan lagi (sebuah model tidak lengkap tanpa KD). KD juga mempunyai fungsi yang sama dalam pemodelan sistem. Selain itu KD berfungsi membantu pelaku sistem untuk mengerti aplikasi secara detail, kamus data mereorganisasi semua elemen data yang digunakan dalam sistem dengan presisi yang sedemikian rupa sehingga pemakai dan penganalisa sistem memiliki dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses. Kamus data adalah katalog fakta tentang data selain digunakan untuk dokumentasi dan mengurangi redudansi, dapat digunakan untuk:

1. Memvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan.
3. Menentukan muatan data yang disimpan dalam file-file.
4. Mengembangkan logika-logika untuk proses-proses diagram aliran data.

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

Menurut Prabowo Pudjo Widodo Herlawati (2011:6) *UML (Unified Modelling Language)* adalah bahasa pemodelan standar. (Chonoles, 2003: bab 1) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep *UML* ada aturan - aturan yang harus diikuti. Bagaimana elemen pada model – model yang dibuat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. *UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

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. Mendokumentasi sistem yang ada, proses – proses dan organisasinya.

II.11.1. Tujuan Pemanfaatan UML

Berikut tujuan utama dalam desain UML adalah (Sugrue J.2009):

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktek terbaik .

II.10.2. Komponen-komponen UML

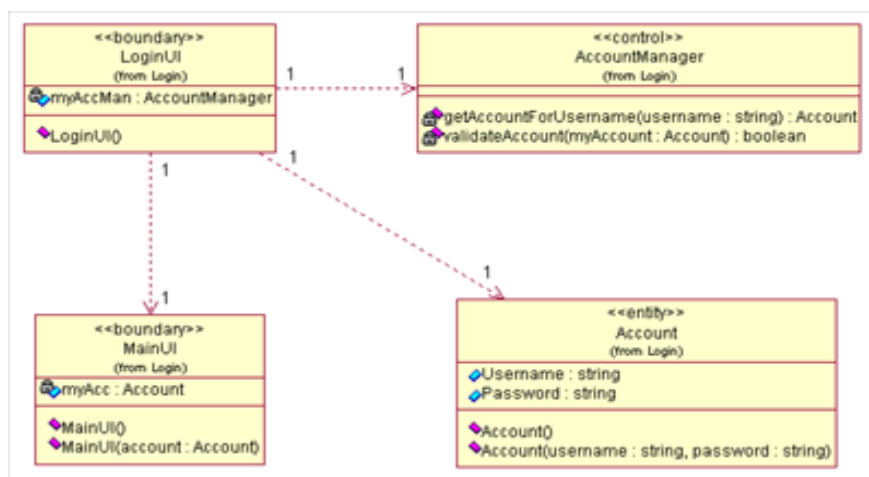
Menurut Sugrue J (2009) Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu *Grady Booch*, OOD (*Object-Oriented Design*), *Jim Rumbaugh*,

OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*). Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu Struktur Diagram yang Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Struktur diagram dalam UML terdiri atas:

1. *Class diagram*

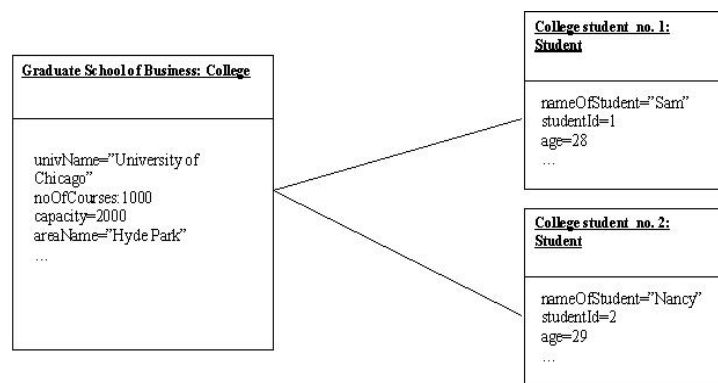
Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok : Nama (dan *stereotype*), atribut, metoda.



Gambar II.4. Notasi *class diagram*
Sumber : (Jurnal Informatika; 2011 : 3)

2. Object diagram

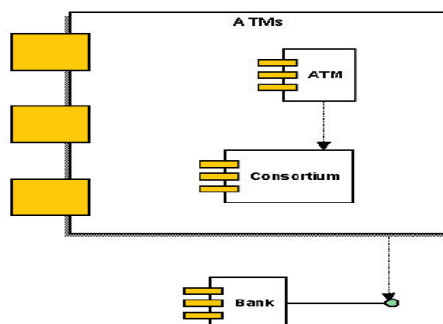
Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak. Berikut notasi *object diagram*.



Gambar II.5. Notasi *Object Diagram*
Sumber : (Haviluddin; 2011 : 3)

3. Component diagram

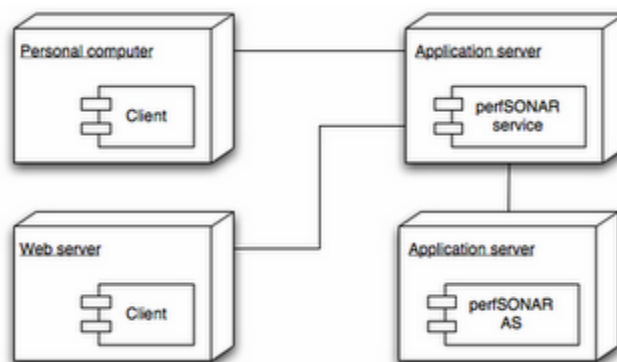
Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.



Gambar II.6. Notasi *Component Diagram*
Sumber : (Jurnal Informatika; 2011 : 3)

4. *Deployment* diagram (*Collaboration* diagram in version 1.x)

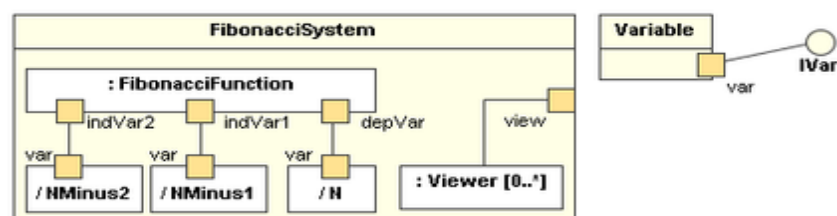
Deployment diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment* diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem.



Gambar II.7. Notasi *Deployment Diagram*
Sumber : (Jurnal Informatika; 2011 : 4)

5. *Composite structure* diagram

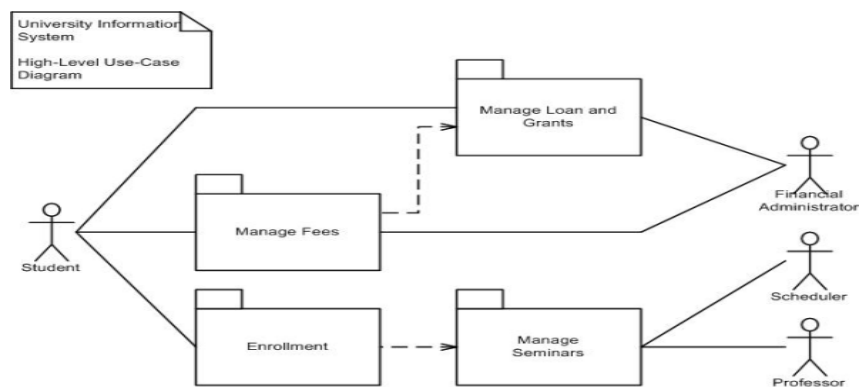
Sebuah diagram struktur komposit mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi.



Gambar II.8. Notasi *Composite Diagram*
Sumber : (Jurnal Informatika; 2011 : 4)

6. Package diagram

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model.



. Gambar II.9. Notasi *Package Diagram*

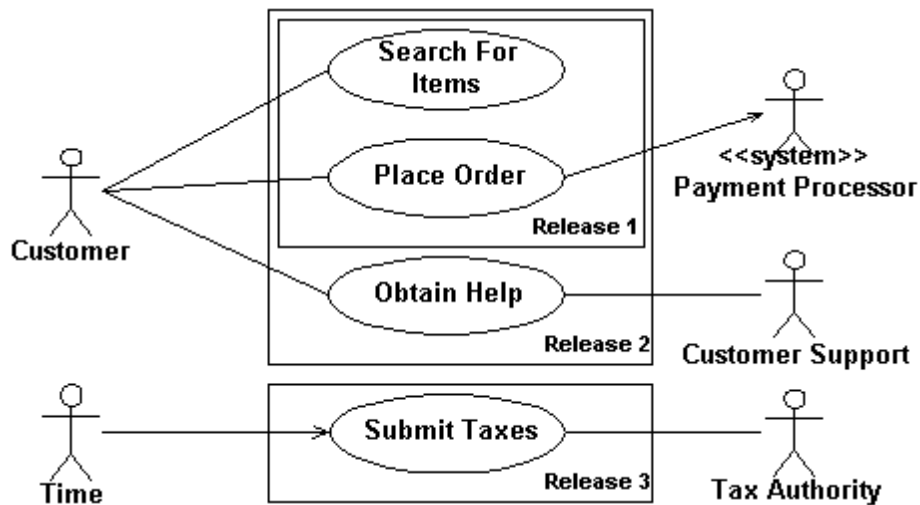
Sumber : (Jurnal Informatika; 2011 : 4)

7. Use case diagram

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*.

Use Case memiliki dua istilah :

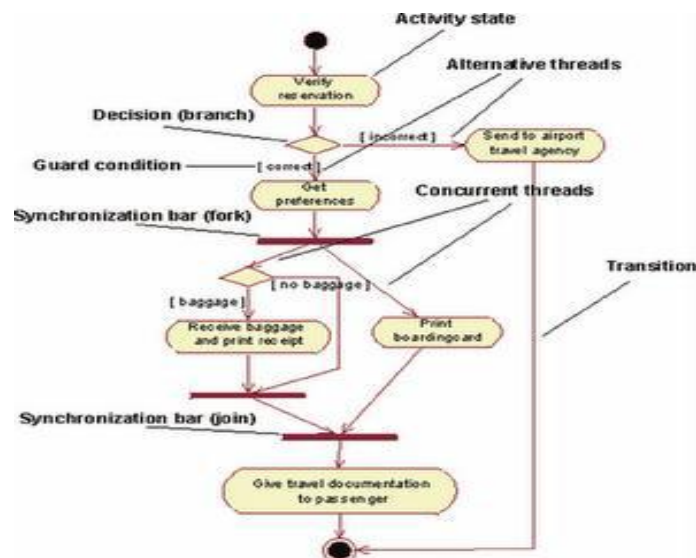
1. *System use case*; interaksi dengan sistem.
2. *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata



Gambar II.10. Notasi Use Case Diagram
Sumber : (Jurnal Informatika ; 2011 : 4)

8. Activity diagram

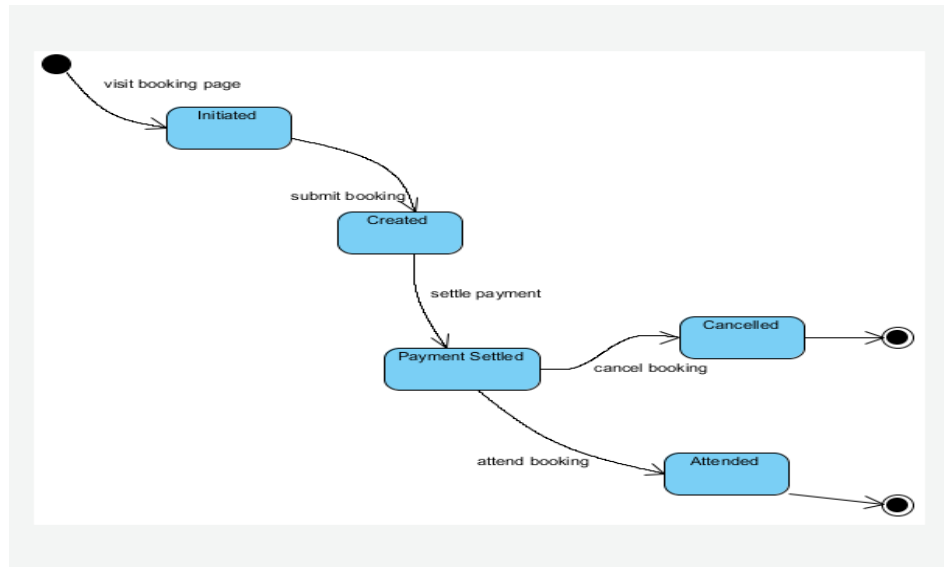
Menggambarakan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas



Gambar II.11. Notasi Activity Diagram
Sumber : (Jurnal Informatika; 2011 : 4)

9. *State Machine* diagram (*State chart* diagram in diagram in version 1.x)

Menggambarakan *state*, transisi *state* dan *event*.

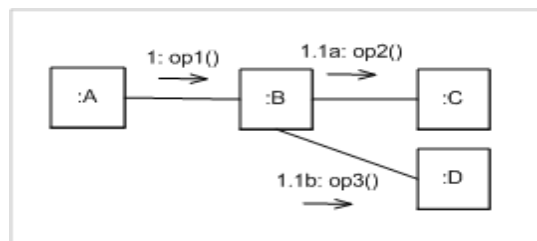


. **Gambar II.12. Notasi *State Machine Diagram***

Sumber : (Jurnal Informatika; 2011 : 5)

10. *Communication* diagram

Serupa dengan *sequence* diagram, tetapi diagram komunikasi juga digunakan untuk memodelkan perilaku dinamis dari *use case*. Bila dibandingkan dengan *Sequence* diagram, diagram komunikasi lebih terfokus pada menampilkan kolaborasi benda daripada urutan waktu.

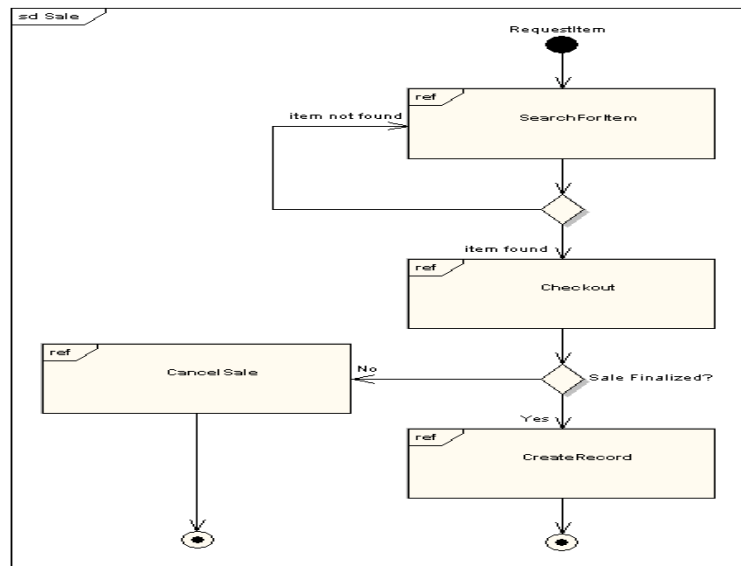


Gambar II.13. Notasi *Communication Diagram*

Sumber : (Jurnal Informatika ; 2011 : 3)

11. *Interaction Overview* diagram

Interaksi overview diagram berfokus pada gambaran aliran kendali interaksi dimana node adalah interaksi atau kejadian interaksi.

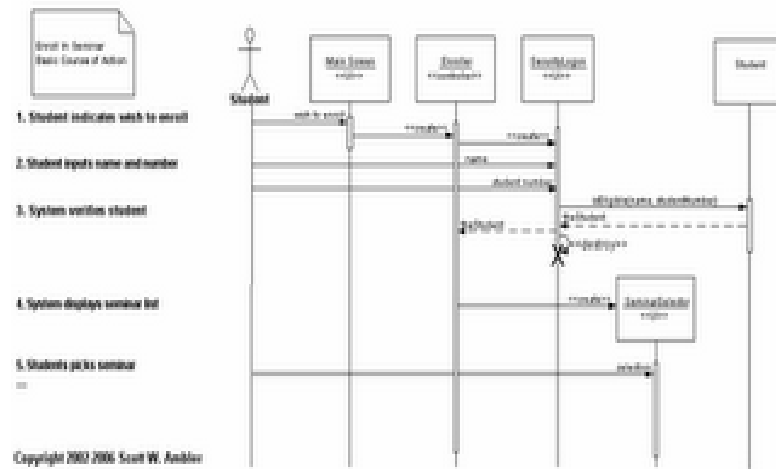


Gambar II.14. Notasi *Overview Diagram*

Sumber : (Jurnal Informatika; 2011 : 3)

12. *Sequence* diagram

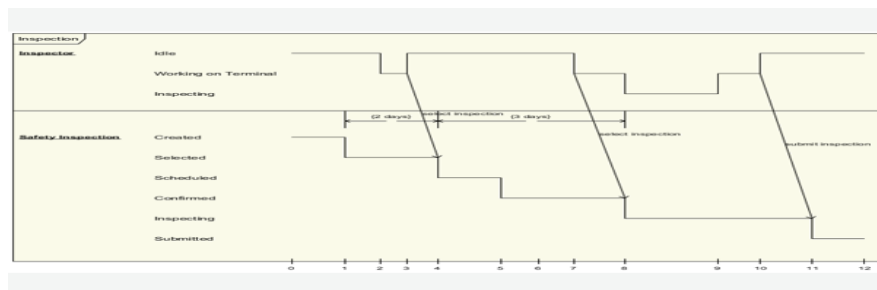
Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.



Gambar II.15. Notasi Sequence Diagram
Sumber : (Jurnal Informatika ; 2011 : 5)

13. Timing diagram

Timing diagram di UML didasarkan pada diagram waktu *hardware* awalnya dikembangkan oleh para insinyur listrik.



Gambar II.16. Notasi Timing Diagram
Sumber : (Jurnal Informatika ; 2011 : 6)