

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Untuk mengawali pembahasan tentang analisis dan perancangan sistem informasi, pemahaman akan sistem terlebih dahulu harus ditekankan. Definisi sistem berkembang sesuai dengan konteks di mana pengertian sistem itu digunakan. Berikut akan diberikan beberapa definisi sistem secara umum :

Kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama.

Sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan.

Dengan demikian, secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variable-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain. sistem sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk satu tujuan bersama. Sementara definisi sistem dalam kamus *Webster's Unbrided* adalah elemen-elemen yang saling berhubungan dan membentuk satu kesatuan atau organisasi (Hanif Al Fatta ; 2007 : 3).

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

1. Komponen Sistem

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

2. Batasan Sistem

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

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

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

5. Masukan Sistem

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

6. Keluaran Sistem

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

7. Pengolah Sistem

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

8. Sasaran Sistem

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya (Sulindawati ; 2010 : 135).

II.2. Pengertian Informasi

Informasi adalah data yang berguna yang telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Informasi sangat penting bagi organisasi. Pada dasarnya informasi adalah penting seperti sumber daya yang lain, misalnya peralatan, bahan, tenaga, dsb.

Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi. Dalam sistem informasi akuntansi, kualitas dari informasi yang disediakan merupakan hal penting dalam kesuksesan sistem.

Secara konseptual seluruh sistem organisasional mencapai tujuannya melalui proses alokasi sumber daya, yang diwujudkan melalui proses pengambilan keputusan manajerial. Informasi memiliki nilai ekonomis pada saat ia mendukung

keputusan alokasi sumber daya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan.

Pemakai informasi akuntansi dapat dibagi dalam dua kelompok besar: ekstern dan intern. Pemakai ekstern mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat pekerja, dan masyarakat. Pemakai intern terutama para manajer, kebutuhannya bervariasi tergantung pada tingkatannya (Agustinus ; 2012 : 1).

II.3. Sistem Informasi Akuntansi

Informasi akuntansi diberikan sebagai alat atau sarana untuk membantu manajer menjalankan fungsi-fungsi manajemen sehingga tujuan organisasi dapat tercapai. Fungsi manajemen, seperti perencanaan (*planning*), pengorganisasian (*organizing*), pengarahan karyawan (*directing*) dan pengendalian (*controlling*) tidak dapat dilakukan tanpa informasi yang memadai. Informasi dalam sebuah organisasi merupakan perekat yang mengikat fungsi-fungsi manajemen dalam sebuah sistem sehingga memungkinkan organisasi bertindak koheren dan harmonis antar berbagai fungsi.

1. Perencanaan (*Planning*)

Tahap pertama dalam perencanaan adalah mengidentifikasi alternatif alternatif yang tersedia, kemudian memilih salah satu dari berbagai alternatif tersebut yang paling baik dan cocok dengan tujuan organisasi. Rencana manajemen biasanya diekspresikan dalam sebuah bentuk formal yang disebut dengan anggaran (*budgets*). Oleh karena itu, istilah penganggaran (*budgeting*)

sering pula disebut sebagai proses perencanaan. Anggaran biasanya dibuat di bawah kendali dan koordinasi seorang manajer atau kepala bagian akuntansi. Umumnya anggaran dibuat setiap tahun dan mencerminkan rencana manajemen dalam format yang spesifik dan kuantitatif. Anggaran ini dibuat berdasarkan data yang dikumpulkan, dianalisis dan diikhtisarkan oleh akuntan manajemen.

2. Pengorganisasian Sumber-sumber Daya (*Organizing*)

Langkah selanjutnya adalah pengorganisasian sumber-sumber daya, dimana manajer menetapkan koordinasi antar bagian (*organizationchart*). Perkembangan Pengorganisasian hanya merupakan alat untuk mencapai tujuan yangtelah ditentukan.

3. Pengarahan Karyawan (*DirectingandMotivating*)

Setelah rencana kerja untuk periode mendatang disusun, manajer harusmelakukan pemantauan terus menerus terhadap kegiatan sehari-hari danmenjaga agar organisasi dapat berfungsi sebagaimana mestinya. Untukmelakukan hal tersebut, maka manajer harus memiliki kemampuan untuk memotivasi dan mengarahkan karyawan secara efektif.

4. Pengendalian dan Pengawasan (*Controlling*)

Dalam melaksanakan tugas pengendalian, manajer sebenarnya mencarijaminan bahwa rencana yang telah disusun dalam bentuk anggaran sudah dilaksanakan sebagaimana mestinya (Suprihatmi : 2009 : 57).

II.3.1. Komponen Sistem Informasi Akuntansi

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

1. Orang-orang 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 (Kusrini ; 2007 : 10).

II.3.2. Modernisasi Akuntansi

Di dalam suatu entitas, sistem akuntansi atau sistem informasi merupakan salah satu bidang fungsional atau subunit organisasi yang tugasnya adalah menyediakan informasi bagi bidang lain. Sistem informasi akuntansi bertanggung jawab melaporkan hasil operasi perusahaan kepada pihak yang berkepentingan. Dengan demikian, bidang kajian sistem informasi akuntansi ini terkait baik dengan akuntansi manajemen maupun akuntansi keuangan. Informasi akuntansi yang dimaksud tidak hanya terbatas pada laporan keuangan yang dipublikasi untuk pihak eksternal. Slip gaji yang diterima karyawan, bukti pembayaran yang diterima pembeli di pasar swalayan, bukti penarikan dana yang keluar dari mesin ATM, atau laporan manajerial, bisa dikategorikan sebagai informasi akuntansi (Indra Bastian ; 2012 : 208).

II.3.3. Faktor-faktor Akuntansi

Informasi akuntansi diberikan sebagai alat atau sarana untuk membantu manajer menjalankan fungsi-fungsi manajemen sehingga tujuan organisasi dapat tercapai. Fungsi manajemen, seperti perencanaan (*planning*), pengorganisasian (*organizing*), pengarahan karyawan (*directing*) dan pengendalian (*controlling*) tidak dapat dilakukan tanpa informasi yang memadai. Informasi dalam sebuah organisasi merupakan perekat yang mengikat fungsi-fungsi manajemen dalam sebuah sistem sehingga memungkinkan organisasi bertindak koheren dan harmonis antar berbagai fungsi (Suprihatmi : 2009 : 57).

II.4. Pengolahan Proyek Kontruksi

Proyek konstruksi berkembang semakin besar dan rumit dewasa ini baik dari segi fisik maupun biaya. Pada prakteknya suatu proyek mempunyai keterbatasan akan sumber daya, baik berupa manusia, material, biaya ataupun alat. Hal ini membutuhkan suatu manajemen proyek mulai dari fase awal proyek hingga fase penyelesaian proyek. Dengan meningkatnya tingkat kompleksitas proyek dan semakin langkanya sumberdaya maka dibutuhkan juga peningkatan sistem pengelolaan proyek yang baik dan terintegrasi (Biemo W. Soemardi ; 1).

II.5. Metode Single Step

Laporan laba/rugi adalah laporan yang menunjukkan besarnya pendapatan dan beban pada akhir periode akuntansi. Penyusunan laporan laba/rugi yang bersumber dari kertas kerja, datanya diambil dari kolom perhitungan laba/rugi.

Penyusunan laporan laba/rugi dapat dibedakan atas dua bentuk, yaitu bentuk langsung (*single step*) dan bentuk bertahap (*multiple step*).

Laporan laba rugi *single step*, adalah laporan laba rugi yang mencatat seluruh penghasilan terlebih dahulu kemudian dikurangi dengan seluruh beban (Ryan Ariefriansyah ; 2013 : 14).

Contoh:

Berdasarkan kertas kerja Perusahaan ADHI JAYA Medan per 31 Desember 2005 (Tabel 8.3), maka laporannya dapat disusun sebagai berikut.

Perusahaan ADHI JAYA, Medan			
LAPORAN LABA/RUGI			
Untuk Periode yang Berakhir 31 Desember 2005			
Pendapatan jasa		Rp	27.500.000,00
Pendapatan lain-lain		Rp	—
Jumlah pendapatan		Rp	27.500.000,00
Beban usaha:			
Beban gaji	Rp	10.000.000,00	
Beban perjalanan dinas	Rp	900.000,00	
Beban iklan	Rp	600.000,00	
Beban telepon	Rp	400.000,00	
Beban listrik dan air	Rp	600.000,00	
Beban Perlengkapan kantor	Rp	400.000,00	
Beban sewa	Rp	250.000,00	
Beban penyusutan peralatan kantor	Rp	500.000,00	
Beban lain-lain	Rp	—	
Jumlah beban usaha		(Rp	13.650.000,00)
Laba bersih		Rp	13.850.000,00

Gambar II.1. Laporan Laba/Rugi
(Sumber : Ryan Ariefriansyah ; 2013 : 14)

II.6. Pengertian Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai jenis komputer dan berbagai sistem operasi termasuk telepon genggam. Java dikembangkan oleh *Sun Microsystem* dan dirilis tahun 1995. Java merupakan suatu teknologi perangkat lunak yang digolongkan *multi platform*. Selain itu, Java juga merupakan suatu *platform* yang memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan suatu program.

Bahasa pemrograman java pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan *versi* yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, serta Sembilan pemrograman lainnya dari *Sun Microsystem*. Salah satu hasil proyek ini adalah mascot Duke yang dibuat oleh Joe Palrang (Wahana Komputer ; 2010 : 1).

II.7. Pengertian NetBeans

NetBeans merupakan salah satu proyek *open source* yang disponsori oleh *Sun Microsystem*. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu NetBeanss IDE dan NetBeans Platform. NetBeans IDE merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-*compile*, mencari kesalahan dan mendistribusikan program. Sedangkan NetBeans Platform adalah sebuah modul yang merupakan kerangka awal / pondasi dalam bangun aplikasi *desktop* yang besar.

NetBeans juga menyediakan paket yang lengkap dalam pemrograman dari pemrograman standar (aplikasi *desktop*), pemrograman *enterprise*, dan pemrograman perangkat *mobile*. Saat ini NetBeans telah mencapai versi 6.8 (Wahana Komputer ; 2010 : 15).

II.8. Pengertian Database

Secara sederhana *database* (basis data/pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Pengertian akses dapat mencakup pemerolehan data maupun manipulasi data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *harddisk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk *database*.

Pengaplikasian *database* dapat kita lihat dan rasakan dalam keseharian kita. *Database* ini menjadi penting untuk mengelola data dari berbagai kegiatan. Misalnya, kita bisa menggunakan mesin ATM (*anjungan tunai mandiri / automatic teller machine*) bank karena bank telah mempunyai *database* tentang nasabah dan rekening nasabah. Kemudian data tersebut dapat diakses melalui mesin ATM ketika bertransaksi melalui ATM. Pada saat melakukan transaksi, dalam konteks *database* sebenarnya kita sudah melakukan perubahan (*update*) data pada *database* di bank. Ketika kita menyimpan alamat dan nomor telepon di

HP, sebenarnya juga telah menggunakan konsep *database*. Data yang kita simpan di HP juga mempunyai struktur yang diisi melalui formulir(*form*) yang disediakan. Pengguna dimungkinkan menambahkan nomor HP, nama pemegang, bahkan kemudian dapat ditambah dengan alamat *email*, alamat *web*, nama kantor, dan sebagainya (Agustinus Mujilan ; 2012 : 23).

II.9. Pengertian MySQL

MySQL adalah suatu sistem manajemen basis data relasional (RDBMS-*Relational Database Management System*) yang mampu bekerja dengan cepat, kokoh, dan mudah digunakan. Contoh RDBMS lain adalah *Oracle*, *Sybase*. Basis data memungkinkan anda untuk menyimpan, menelusuri, menurutkan dan mengambil data secara efisien. *Server MySQL* yang akan membantu melakukan fungsionalitas tersebut. Bahasa yang digunakan oleh MySQL tentu saja adalah *SQL-standar* bahasa basis data relasional di seluruh dunia saat ini.

MySQL dikembangkan, dipasarkan dan disokong oleh sebuah perusahaan Swedia bernama MySQL AB. RDBMS ini berada di bawah bendera GNU GPL sehingga termasuk produk *Open Source* dan sekaligus memiliki lisensi komersial. Apabila menggunakan MySQL sebagai basis data dalam suatu situs Web. Anda tidak perlu membayar, akan tetapi jika ingin membuat produk RDBMS baru dengan basis MySQL dan kemudian mengualnua, anda wajib bertemu mudah dengan lisensi komersial (Antonius Nugraha Widhi Pratama ; 2010 : 10).

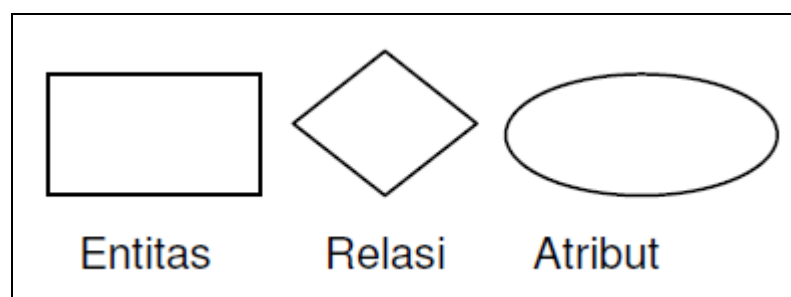
II.10. *Entity Relationship Diagram (ERD)*

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

Model *entity-relationship* pertama kali diperkenalkan oleh Peter Chen pada tahun 1976. Dalam pemodelan ini dilakukan dengan tahapan sebagai berikut:

- a. Memilih entitas-entitas yang akan disusun dalam basis data dan menentukan hubungan antar entitas yang telah dipilih.
- b. Melengkapi atribut-atribut yang sesuai pada entitas dan hubungan sehingga diperoleh bentuk tabel normal penuh (ternormalisasi).

Elemen-elemen dalam model ER dapat digambarkan pada gambar diagram di bawah ini :



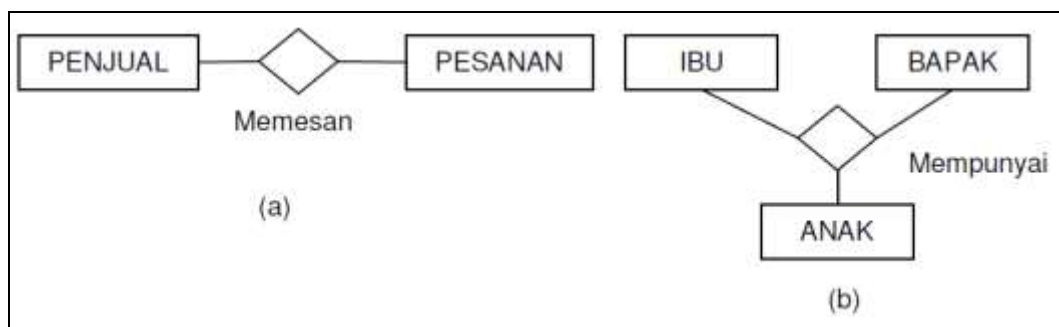
Gambar II.2. Elemen ERD
(Sumber : Haidar Dzacko, 2007 : 21).

Entitas merupakan sesuatu yang dapat diidentifikasi dalam lingkungan kerja pengguna. Entitas yang diberikan tipe dikelompokkan ke kelas entitas. Perbedaan antara kelas entitas dan instansi entitas adalah sebagai berikut:

- a. Kelas entitas adalah kumpulan entitas dan dijelaskan oleh struktur atau format entitas di dalam kelas.
- b. Instansi kelas merupakan bentuk penyajian dari fakta entitas.

Umumnya terdapat banyak instansi entitas di dalam setiap entitas kelas. Setiap entitas kelas memiliki atribut yang menjelaskan karakteristik dari entitas tersebut, sedangkan setiap instansi entitas mempunyai identifikasi yang dapat bernilai unik (mempunyai nilai yang berbeda untuk setiap identifikasinya) atau non-unik (dapat bernilai sama untuk setiap identifikasinya).

Antara entitas diasosiasikan dalam suatu hubungan (*relationship*). Suatu relasi dapat memiliki beberapa atribut. Jumlah kelas entitas dalam suatu relasi disebut derajat relasi. Gambar di bawah ini merupakan contoh dari relasi berderajat dua dan relasi berderajat tiga (Haidar Dzacko ; 2007 : 21).



Gambar II.3 (a) Relasi Berderajat Dua (b) Relasi Berderajat Tiga
 (Sumber : Haidar Dzacko, 2007 : 21).

II.11. Kamus Data

Kamus data (*data dictionary*) mencakup definisi-definisi dari data yang disimpan di dalam basis data dan dikendalikan oleh sistem manajemen basis data. Figur 6.5 menunjukkan hanya satu tabel dalam basis data jadwal. Struktur basis data yang dimuat dalam kamus data adalah kumpulan dari seluruh definisi *field*, definisi tabel, relasi tabel, dan hal-hal lainnya. Nama *field* data, jenis data (seperti teks atau angka atau tanggal), nilai-nilai yang valid untuk data, dan karakteristik-karakteristik lainnya akan disimpan dalam kamus data. Perubahan-perubahan pada struktur data hanya dilakukan satu kali di dalam kamus data, program-program aplikasi yang mempergunakan data tidak akan ikut terpengaruh (Raymond McLeod ; 2008 : 171).

II.12. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

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

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

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada

pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

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

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

3. Bentuk normal tahap ketiga (3rd normal form)

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

4. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan

fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 76).

Faktur Pembelian Barang				
PT. Ayo Bersama				
Jl. Senopati 12 yk				
Kode Suplier : G01			Tanggal : 07/10/04	
Nama Suplier : Gober Nustra			Nomor : 004	
Kode	Nama Brg	Qty	Harga	Jml
A01	AC model 1	10	1000000	10000000
A02	AC model 2	5	900000	4500000
		Total Faktur		14500000
Jatuh Tempo 15/10/04				

Gambar II.4. Dokumen Dasar
(Sumber : Janner Simarmata ; 2010 : 76).

II.13. UML (*Unified Modeling Language*)

Menurut Haviluddin (2011 ; 3) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

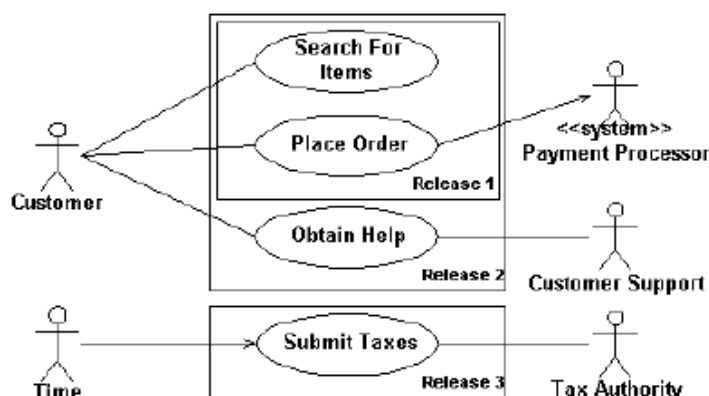
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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

1. 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

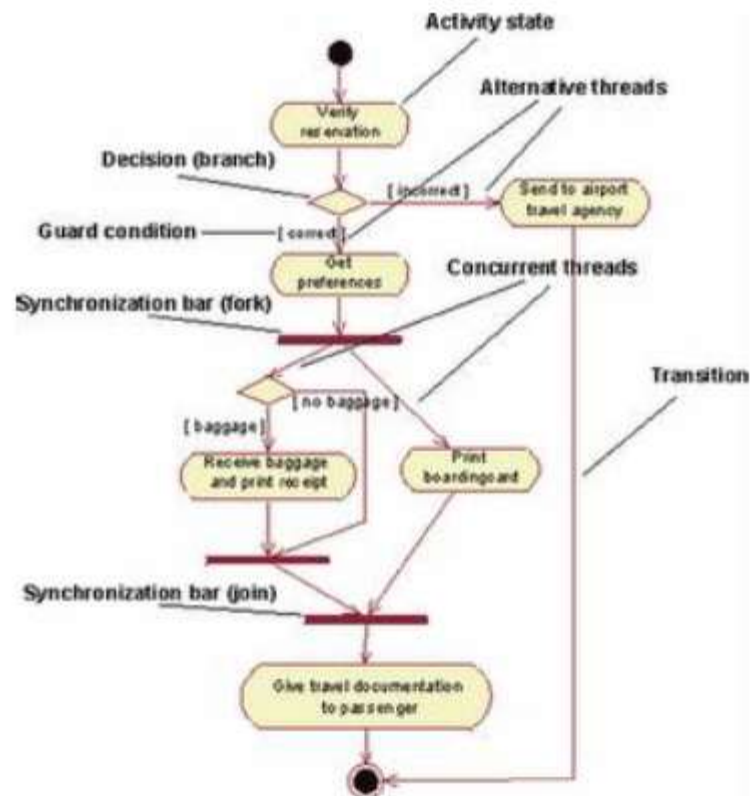
- a. *System use case*; interaksi dengan sistem.
- b. *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata



Gambar II.5. Contoh Usecase diagram
Sumber : Haviluddin ; 2011 : 3

2. Activity Diagram

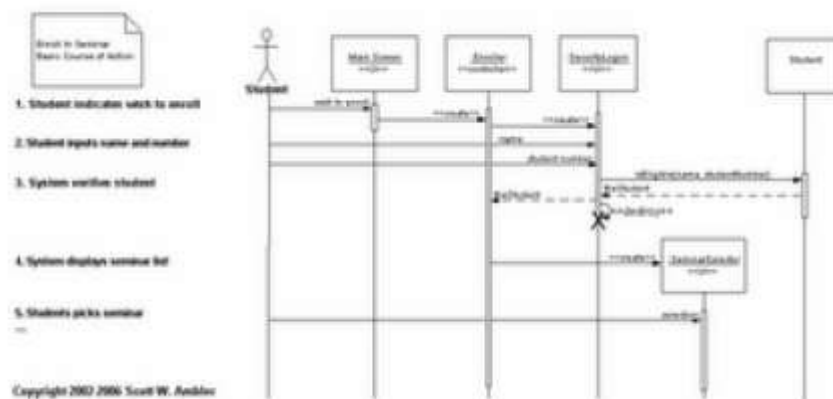
Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.



Gambar II.6. Contoh Activity Diagram
Sumber : Haviluddin ; 2011 : 3

3. 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



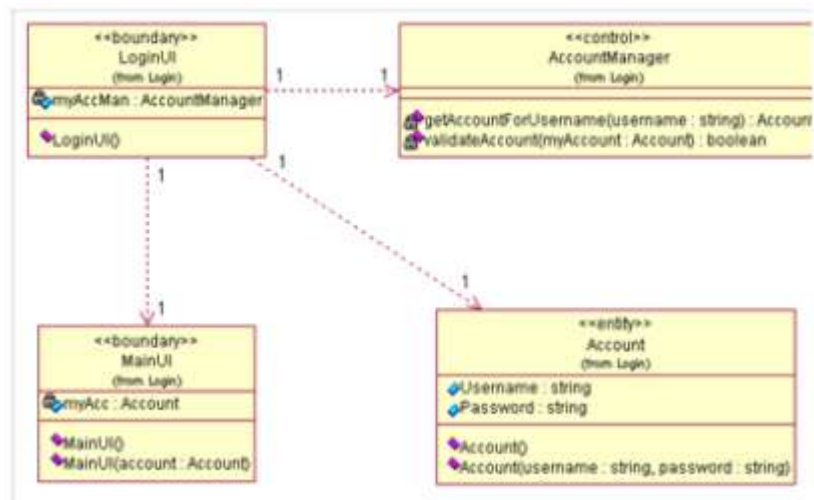
Gambar II.7. Contoh Sequence Diagram

Sumber : Haviluddin ; 2011 : 3

4. 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 :

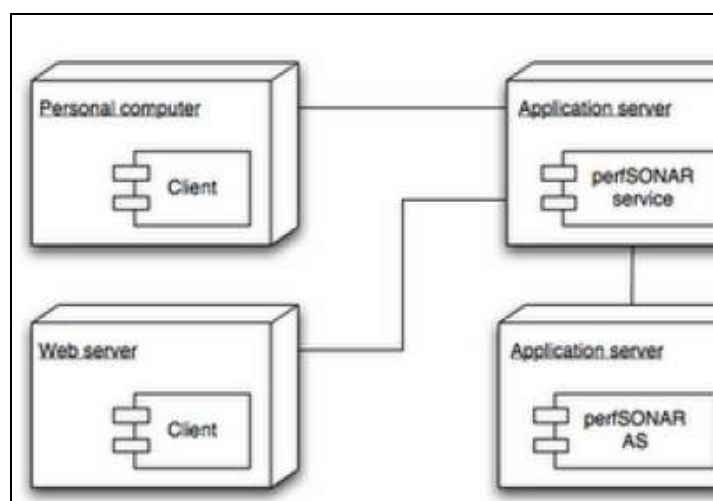
- Nama (dan *stereotype*)
- Atribut
- Metoda



Gambar II.8. Contoh Class Diagram
Sumber : Haviluddin ; 2011 : 3

5. Deployment Diagram

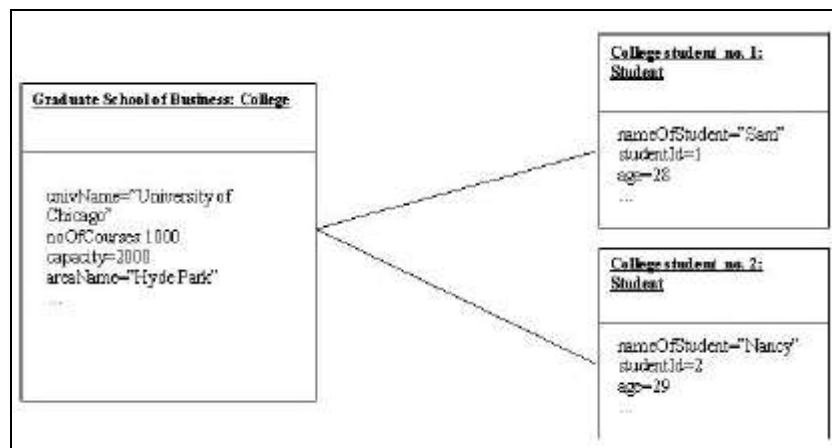
Deployment diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment* diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan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



Gambar II.9. Contoh Deployment Diagram
Sumber : Haviluddin ; 2011 : 3

6. Objek Diagram

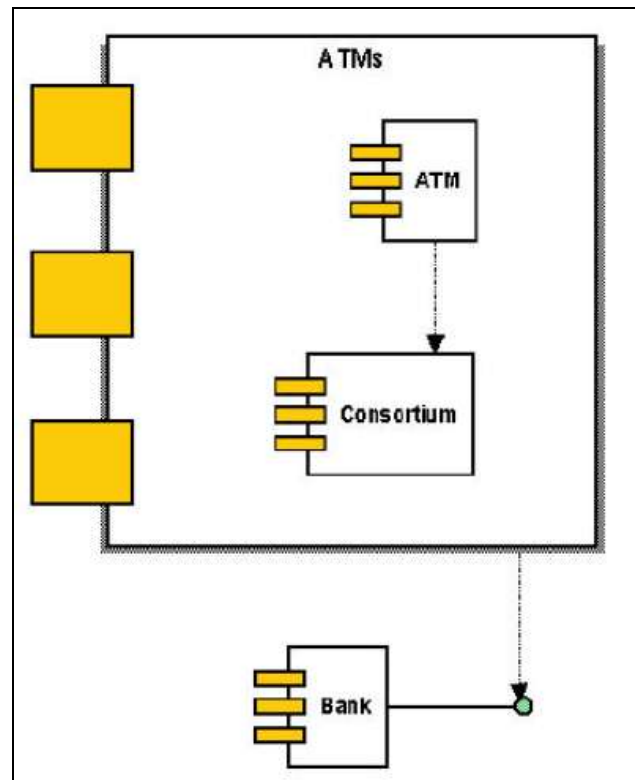
Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.



Gambar II.10. Contoh Objek Diagram
Sumber : Haviluddin ; 2011 : 3

7. Component diagram

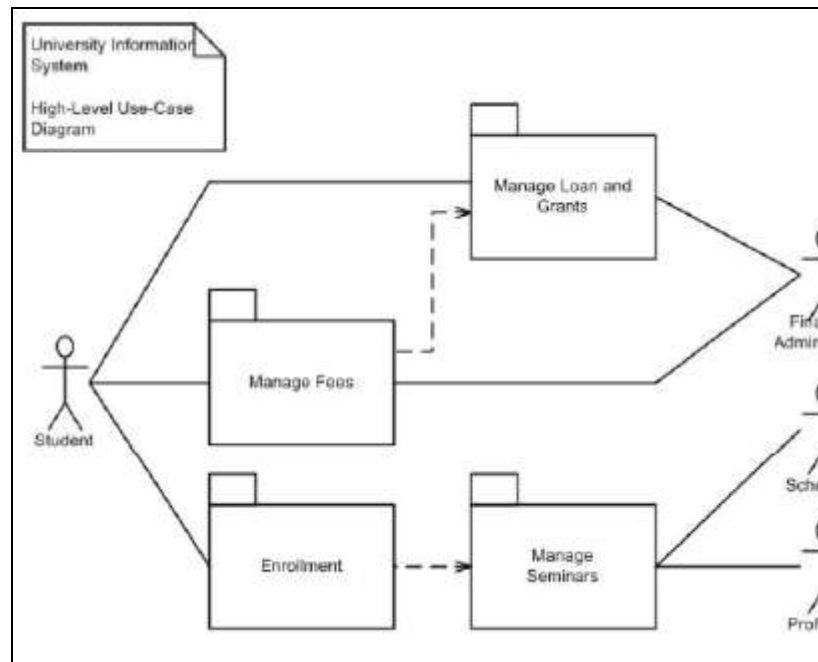
Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.



Gambar II.11. Contoh *Component Diagram*
Sumber : Haviluddin ; 2011 : 3

8. *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.12. Contoh Package Diagram
Sumber : Haviluddin ; 2011 : 3