

BAB II

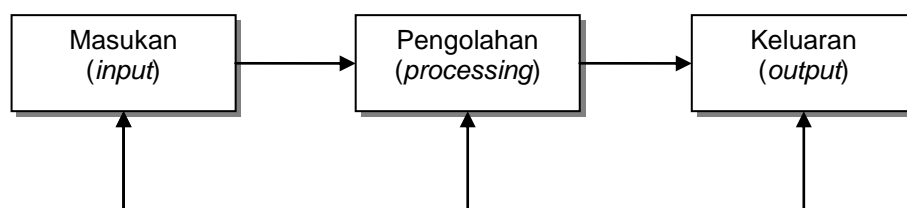
TINJAUAN PUSTAKA

II.1. Konsep Dasar Sistem

II.1.1. Pengertian Sistem

Menurut Hanif Al Fatta (2007 : 3) secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel- variable yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain. Sedangkan menurut Anastasia Diana,dkk (2011: 3) sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu.

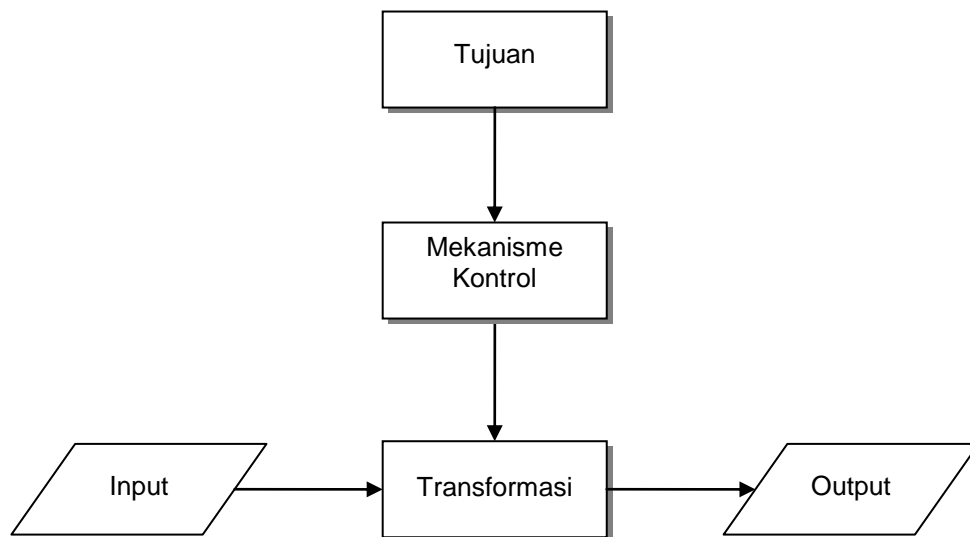
Sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*processing*), serta keluaran (*output*). Ciri pokok sistem menurut Gapsper ada empat, yaitu sistem itu beroperasi dalam suatu lingkungan, terdiri atas unsur-unsur, ditandai dengan saling berhubungan, dan mempunyai satu fungsi atau tujuan utama.



Gambar II.1. Model Sistem

Gambar di atas menunjukkan bahwa sistem atau pendekatan sistem minimal harus mempunyai empat komponen, yakni masukan, pengolahan, keluaran, dan balikan atau *control*.

Sementara *Mc. Leod* pada tahun (1995) mendefinisikan sistem sebagai sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Sumber daya mengalir dari elemen output dan untuk menjamin prosesnya berjalan dengan baik maka dihubungkan dengan mekanisme *control*. Untuk lebih jelasnya elemen sistem tersebut dapat di lihat pada gambar II.2 :



Gambar II.2. Model hubungan elemen-elemen sistem

Banyak ahli mengajukan konsep sistem dengan deskripsi yang berbeda, tetapi pada prinsipnya hampir sama dengan konsep dasar sistem umumnya. *Schröderberg* pada tahun (1971) dan *Suradinata* pada tahun (1996) secara ringkas menjelaskan bahwa sistem adalah :

1. Komponen-komponen yang saling berhubungan satu sama lain.
2. Suatu keseluruhan tanpa memisahkan komponen pembentuknya.
3. Bersama-sama dalam mencapai tujuan.
4. Memiliki input dan output yang dibutuhkan oleh sistem lainnya.

5. Terdapat proses yang mengubah input menjadi output.
6. Menunjukkan adanya entropi.
7. Memiliki aturan.
8. Memiliki subsistem yang lebih kecil.
9. Memiliki deferensi antar subsistem.
10. Memiliki tujuan yang sama meskipun mulainya berbeda. (Hanif Al Fattah, 2007 : 3)

II.1.2. Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem yang lainnya :

1. Batasan (*boundary*) : Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*environment*) : Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*input*) : Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*output*) : sumber daya atau produk (informasi, laporan, dokumen, tampilan layer computer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.

5. Komponen (*component*) : Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*).
Komponen ini bisa merupakan subsistem dari sebuah sistem.
6. Penghubung (*interface*) : Tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*storage*) : Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama. (Hanif Al Fattah, 2007 : 5)

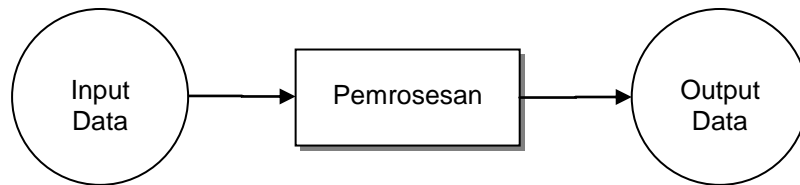
II.1.3. Pengertian Informasi

Menurut Hanif Al Fatta (2007 : 9) informasi adalah data yang telah di olah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.

II.1.4. Pengertian Sistem Informasi

Sistem informasi adalah kesatuan elemen- elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi yang akan mendukung pembuatan keputusan dan melakukan kontrol terhadap jalannya perusahaan. Sistem informasi dapat didefinisikan sebagai alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya.

Sistem informasi berdasarkan konsep (*input, processing, output – IPO*) dapat dilihat pada gambar :



Gambar II.3. Konsep sistem informasi (Hanif Al Fattah, 2007 : 9)

II.1.5. Komponen Sistem Informasi

Stair menjelaskan bahwa sistem informasi berbasis komputer (CBIS) dalam suatu organisasi terdiri dari komponen-komponen berikut :

1. Perangkat keras, yaitu perangkat keras komponen untuk melengkapi kegiatan memasukkan data, memproses data, dan keluaran data.
2. Perangkat lunak, yaitu program dan instruksi yang diberikan ke komputer.
3. *Database*, yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.
4. Telekomunikasi, yaitu komunikasi yang menghubungkan antara pengguna sistem dengan sistem komputer secara bersama-sama ke dalam suatu jaringan kerja yang efektif.
5. Manusia, yaitu personel dari sistem informasi, meliputi manajer, analis, programmer, dan operator, serta yang bertanggung jawab terhadap perawatan sistem.
6. Prosedur, yakni tata cara yang meliputi strategi, kebijakan, metode, dan peraturan-peraturan dalam menggunakan sistem informasi berbasis komputer.

Sementara *Burch dan Grudnitski* pada tahun (1986) berpendapat, sistem informasi yang terdiri dari komponen-komponen di atas disebut dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), dan blok kendali (*control block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sarannya.

1. Blok Masukan. Input mewakili data yang masuk ke dalam sistem informasi. Input di sini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan yang dapat berupa dokumen-dokumen dasar.
2. Blok Model. Blok ini terdiri dari kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.
3. Blok Keluaran. Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen serta semua pemakai sistem.
4. Blok Teknologi. Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan sekaligus mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.
5. Blok Database. Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali. Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

Sementara, menurut pendapat *Davis* pada tahun (1995), sistem informasi manajemen terdiri dari elemen-elemen berikut :

1. Perangkat keras komputer (*hardware*)
2. Perangkat lunak (*software*), yang terdiri dari perangkat lunak sistem umum, perangkat lunak terapan, dan program aplikasi.
3. Database
4. Prosedur
5. Petugas operasional. Hanif Al Fattah (2007 : 9)

II.1.6. Definisi Sistem Informasi Akuntansi

Sistem Informasi Akuntansi (SIA) adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Anastasia Diana,dkk (2011 : 4).

Pengguna informasi akuntansi dapat di kelompokkan menjadi dua kelompok besar antara lain :

1. Pengguna *Eksternal*, yang terdiri dari pemegang saham, investor, kreditor, agen pemerintah, konsumen, vendor, pesaing, serikat kerja, dan masyarakat secara luas.
2. Pengguna *Internal*, terdiri dari para manajer.

Fungsi penting yang dibentuk *Sistem Informasi Akuntansi* pada sebuah organisasi antara lain :

1. Mengumpulkan dan menyimpan data tentang aktivitas dan transaksi.
2. Memproses data menjadi into informasi yang dapat digunakan dalam proses pengambilan keputusan.
3. Melakukan kontrol secara tepat terhadap aset organisasi.

Subsistem *Sistem Informasi Akuntansi* memproses berbagai transaksi keuangan dan transaksi non keuangan yang secara langsung memengaruhi pemrosesan transaksi keuangan.

Sistem Informasi Akuntansi terdiri dari 3 subsistem:

1. Sistem pemrosesan transaksi ialah mendukung proses operasi bisnis harian.
2. Sistem buku besar/ pelaporan keuangan ialah menghasilkan laporan keuangan, seperti laporan laba/rugi, neraca, arus kas, pengembalian pajak.
3. Sistem pelaporan manajemen ialah yang menyediakan pihak manajemen internal berbagai laporan keuangan bertujuan khusus serta informasi yang dibutuhkan untuk pengambilan keputusan, seperti anggaran, laporan kinerja, serta laporan pertanggung jawaban.

Adapun manfaat dari sistem informasi akuntansi adalah :

1. Menyediakan informasi yang akurat dan tepat waktu sehingga dapat melakukan aktivitas utama pada *value chain* secara efektif dan efisien.
2. Meningkatkan kualitas dan mengurangi biaya produk dan jasa yang dihasilkan.
3. Meningkatkan efisiensi.
4. Meningkatkan kemampuan dalam pengambilan keputusan.
5. Meningkatkan *sharing knowledge*.

6. Menambah efisiensi kerja pada bagian keuangan.

II.2. Definisi Penerimaan dan Pengeluaran Kas

II.2.1. Penerimaan dan Pengeluaran Kas

Prosedur penerimaan kas perusahaan perlu dirancang dengan sedemikian rupa sehingga kemungkinan tidak diterimanya uang yang seharusnya diterima dapat dikurangi menjadi sekecil mungkin.

Adapun beberapa prosedur penerimaan kas yang perlu diperhatikan yaitu:

1. Terdapat pemisahan tugas antara yang menyimpan, yang menerima dan yang mencatat penerimaan uang.
2. Setiap penerimaan uang langsung disetor ke bank sebagaimana adanya.

Seperti halnya penerimaan uang, prosedur pengeluaran kas perlu dirancang sedemikian rupa sehingga hanya pengeluaran-pengeluaran yang telah disetujui dan betul-betul untuk kegiatan perusahaan saja yang dicatat dalam pembukuan perusahaan.

Adapun prosedur pengeluaran kas harus memperhatikan hal-hal sebagai berikut:

1. Semua pengeluaran dilakukan cek pengeluaran dalam jumlah kecil melalui dana kas kecil.
2. Semua pengeluaran kas harus memperoleh persetujuan dari yang berwenang terlebih dahulu.

3. Terdapat pemisahan tugas antara yang berhak menyetujui pengeluaran kas, yang menyimpan uang kas dan melakukan pengeluaran serta yang mencatat pengeluaran kas.

II.3. Basis Data (*Database*)

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.

Pada era kemajuan teknologi seperti sekarang ini, nilai informasi sangatlah penting, terlebih bagi kemajuan perusahaan. Oleh karena itu penggunaan dan penguasaan database sangat penting. Dalam database ada sebutan-subutan untuk satuan data yaitu:

1. Karakter adalah satuan data terkecil, data terdiri atau susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
2. Field adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama siswa, tanggal lahir dan lain-lain. Dalam dunia perancangan database, field juga disebut atribut. Bila dipandang dari sudut pemrograman berorientasi obyek maka name dan property type. Properti name atau nama adalah properti dari field yang berisi field yang mewakili data sejenis yang disimpannya. Sedangkan properti type adalah property yang mengatur tipe data dari data yang akan ditampungnya. Misalnya

nama fieldnya adalah nama siswa maka tipe datanya adalah char, bila nama fieldnya adalah tanggal lahir maka tipe datanya adalah date. Field dilihat seperti kolom.

3. Record adalah kumpulan dari field. Pada record anda dapat menemukan banyak sekali informasi penting dengan cara mengombinasikan field-field yang ada.
4. Tabel adalah sekumpulan dari record-record yang memiliki kesamaan entity dalam dunia nyata. Kumpulan dari table adalah database, wujud fisik sebuah database dalam komputer adalah sebuah file yang didalamnya terdapat berbagai tingkatan data yang telah disebutkan di atas.
5. File adalah bentuk fisik dari penyimpanan data. File database berisi semua data yang telah disusun dan diorganisasikan sedemikian rupa sehingga memudahkan pemberian informasi. Wahana Komputer (2010 : 24)

II.3.1. Entity Relationship Diagram

Pada dasarnya ERD (*Entity Relationship Diagram*) adalah sebuah diagram yang secara konseptual memetakan hubungan antar entitas pada diagram DFD di atas. ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaannya ERD ini dilakukan untuk mengurangi tingkatan kerumitan penyusunan sebuah database yang baik.

Entity dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen-komponen data (atribut atau field) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia database entity memiliki atribut yang menjelaskan karakteristik dari entity

tersebut. Ada dua macam atribut yang di kenal deskriptif. Hal ini berarti setiap entity memiliki himpunan yang diperlukan sebuah primary key untuk membedakan anggota-anggota dalam himpunan tersebut.

Atribut dapat memiliki sifat-sifat sebagai berikut:

1. *Atomic* adalah sifat dari atribut yang menggambarkan bahwa atribut tersebut berisi nilai yang spesifik dan tidak dapat dipecah lagi.
2. *Multivalued*, sifat ini menandakan atribut ini bila bisa memiliki lebih dari satu nilai untuk tiap *entity* tertentu.
3. *Composite* atribut yang bersifat komposit adalah atribut yang nilainya adalah bangunan dari beberapa atribut yang bersifat atonik. Wahana Komputer (2010 : 30)

II.3.2. Kamus Data

Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *systems data dictionary* adalah catalog fakta tentang data dan kebutuhan- kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, analis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Kd harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk maksud keperluan ini, maka KD harus memuat hal-hal berikut ini.

1. Nama arus data.

Karena KD dibuat berdasarkan arus data yang mengalir di DAD, maka nama dari arus data juga harus dicatat di KD, Sehingga mereka yang membaca DAD dan memerlukan penjelasan lebih lanjut tentang suatu

arus data tertentu di DAD dapat langsung mencarinya dengan mudah di KD.

2. Alias.

Alias atau nama lain dari data dapat dituliskan bila nama ini ada. Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya.

3. Bentuk data.

Telah diketahui bahwa arus data dapat mengalir;

- a. Dari kesatuan luar ke suatu proses, data yang mengalir ini biasanya tercatat di suatu dokumen atau formulir;
- b. Hasil dari suatu proses ke satuan luar, data yang mengalir ini biasanya terdapat di media laporan atau *query* tampilan layar atau dokumen hasil cetakan komputer;
- c. Hasil suatu proses ke proses yang lain, data yang mengalir ini biasanya dalam bentuk variable atau parameter yang dibutuhkan oleh proses penerimanya;
- d. Hasil suatu proses yang direkam ke simpanan data, data yang mengalir ini biasanya berbentuk suatu variable;
- e. Dari simpanan data dibaca oleh suatu proses, data yang mengalir ini biasanya berupa suatu *field* (item data).

4. Arus data

Arus data menunjukkan dari mana data mengalir dan ke mana data akan menuju. Keterangan arus data ini perlu dicatat di KD supaya memudahkan mencari arus data ini di DAD.

5. Penjelasan

Untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat di KD, maka bagian dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

6. Periode

Periode ini menunjukkan kapan terjadinya arus data ini. Periode perlu dicatat di KD karena dapat digunakan untuk mengidentifikasi kapan input data harus dimasukkan ke sistem, kapan proses dari program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

7. Volume

Volume yang perlu dicatat di KD adalah tentang volume rata-rata volume puncak dari arus data. Volume rata-rata menunjukkan banyak rata-rata arus data yang mengalir dalam satu periode tertentu dan volume puncak menunjukkan volume yang terbanyak.

8. Struktur data

Struktur data menunjukkan arus data yang dicatat di KD terdiri dari item-item data apa saja. Prof. Dr. Jogiyanto Hm, MBA, Akt. (2005 : 726-727)

II.3.3. Normalisasi

Setelah memulai tahapan di atas atau ERD, maka hasil oada diagram tersebut memulai direlasasikan pada table-tabel database. Untuk itu diperlukan sebuah tahapan yang disebut normalisasi. Normalisasi data adalah proses dimana table-tabel pada database dices dalam hal kesaling tergantungan di antara *field-field* pada sebuah table. Misalnya jika pada sebuah table terdapat ketergantungan terhadap lebih dari satu *field* dalam table tersebut, maka table tersebut harus dipecah menjadi banyak table.

Ada beberapa langkah dalam normalisasi table, yaitu:

1. Decomposition, dekomposisi adalah proses mengubah bentuk table supaya memenuhi syarat tertentu sebagai table yang baik. Dekomposisi dapat dikatakan berisi jika table yang dikenal dekomposisi bila digabungkan kembali dapat menjadi table awal sebelum di dekomposisi.
2. Bentuk tidal normal, pada bentuk ini semua data yang ada pada tiap *entity* (diambil artibutnya) masih ditampung dalam satu table besar. Data yang ada pada table ini masih ada yang redudansi dan ada yang kosong.
3. Normal *Form* pertama (1st Normal *Form*), pada tahapan ini table di dekomposisi dari table bentuk tidak normal yang kemudian dipisahkan menjadi table-tabel kecil yang memiliki criteria tidak memiliki artibut yang bernilai ganda dan komposit.
4. Normal *Form* kedua(2nd Normal *Form*), pada tahapan ini table dianggap memenuhi normal kedua jika pada table tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer table tersebut.

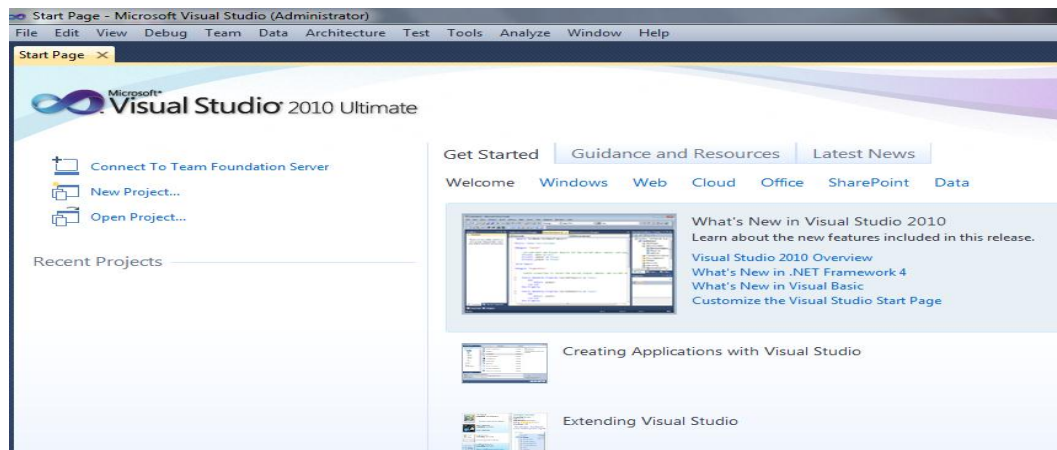
5. Normal *Form* ketiga(3rd Normal *Form*), setiap atribut pada table selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada table yang belum efisien. Wahana Komputer (2010 : 32)

II.4. Definisi Visual Basic

Microsoft Visual Basic 2010 merupakan salah satu aplikasi pemrograman visual yang paling banyak digunakan dan cukup populer karena kemudahan penggunaannya.

Microsoft Visual Basic 2010 adalah salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio 2010*. Sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang dikeluarkan oleh *Microsoft*, *Visual Studio 2010* menambahkan perbaikan-perbaikan fitur dan fitur baru yang lebih lengkap dibandingkan versi *Visual Studio* pendahulunya, yaitu *Microsoft Visual 2008*. Wahana Komputer (2010 : 2)

Adapun Tampilan Awal *Microsoft Visual Basic* 2010 adalah sebagai berikut :



Gambar II.4. Tampilan Awal Microsoft Visual Basic 2010

(Sumber : Wahana Komputer; 2010 : 12)

II.5. Definisi Microsoft SQL Server 2008

SQL Server merupakan suatu *Relational Database Management system (RDBMS)* yang di gunakan untuk menyimpan data. Data yang di simpan dalam *database* bisa dalam skala kecil maupun besar. Dengan menggunakan *SQL Server* 2008, hal tersebut bukanlah masalah karena *SQL Server* 2008 memiliki *resource* yang mumpuni dalam mengolah data. Pada *SQL Server* 2008, kita bisa mengimplementasikan beberapa fitur baru yang dapat meningkatkan aktivitas dan performa *database* pada *SQL Server*.

Menurut Wahana Komputer (2010 : 2) *SQL Server* 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang *Database*. *SQL Server* adalah sebuah *DBMS (Database Management System)*.

II.6. UML (*Unified Modeling Language*)

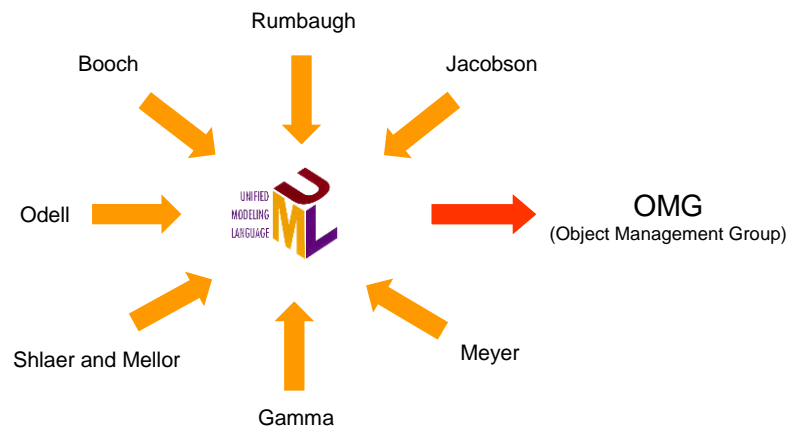
II.6.1. Pengertian UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri dalam visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembangan sistem yang membuat *blue print* atas visinya dalam bentuk yang baku. Yuni Sugiarti, S.T., M.Kom (2013 : 34)

II.6.2. Diagram-Diagram *UML*

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis.



Gambar II.5. Metodologi pemodelan berorientasi objek
(Sumber : Yuni Sugiarti, S.T., M.Kom (2013 : 3))

II.6.3. Konsepsi Dasar UML

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan dalam gambar dibawah.

Major Area	View	Diagrams	Main Concepts
structural	static view	class diagram	class, association, generalization, dependency, realization, interface
	use case view	use case diagram	use case, actor, association, extend, include, use case generalization
	implementation view	component diagram	component, interface, dependency, realization
	deployment view	deployment diagram	node, component, dependency, location
dynamic	state machine view	statechart diagram	state, event, transition, action
	activity view	activity diagram	state, activity, completion transition, fork, join
	interaction view	sequence diagram	interaction, object, message, activation
collaboration diagram		collaboration, interaction, collaboration role, message	
model management	model management view	class diagram	package, subsystem, model
extensibility	all	all	constraint, stereotype, tagged values

Gambar II.6. konsepsi dasar UML
(Sumber : Yuni Sugiarti, S.T., M.Kom (2013 : 3))

Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari *Diagrams. Main concepts* bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagram tersebut.

Lalu darimana kita mulai ? Untuk menguasai UML, sebenarnya cukup dua hal yang harus kita perhatikan:

1. Menguasai pembuatan diagram UML
2. Menguasai langkah-langkah dalam analisa dan pengembangan dengan UML

Tulisan ini pada intinya akan mengupas kedua hal tersebut.

Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram-diagram sebagai berikut:

1. *use case diagram*
2. *class diagram*
3. *statechart diagram*
4. *activity diagram*
5. *sequence diagram*
6. *collaboration diagram*
7. *component diagram*
8. *deployment diagram*. Sri Dharwiyanti (2006: 7)

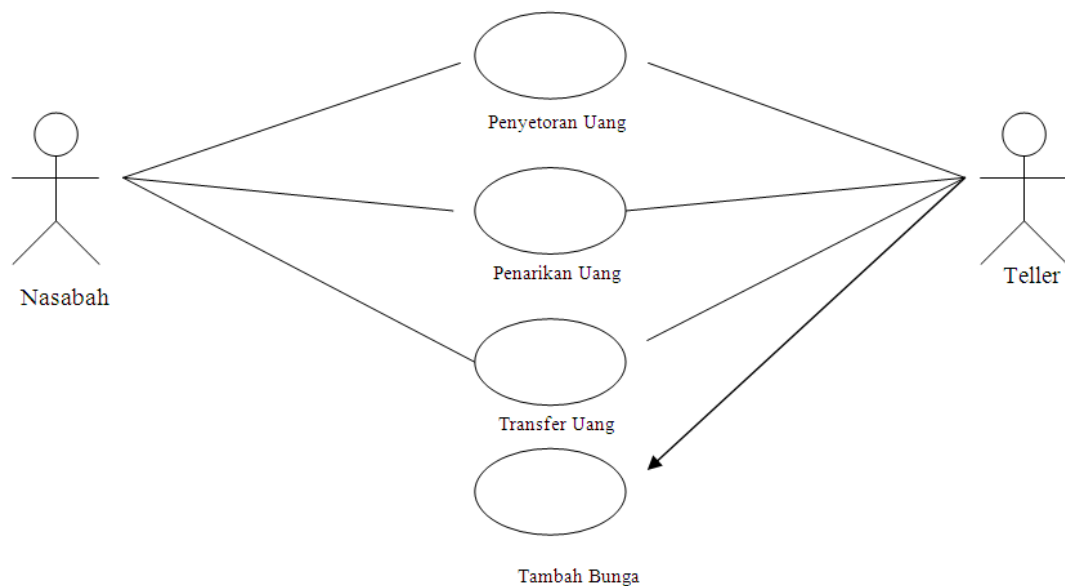
1. *Diagram Use Case (Use Case Diagram)*

Use Case menggambarkan *external view* dari sistem yang akan kita buat modelnya. Menurut Pooley (2005:15) mengatakan bahwa model *use case* dapat dijabarkan dalam diagram, tetapi yang perlu diingat, diagram tidak indentik dengan model karena model lebih luas dari diagram.

komponen pembentuk diagram *use case* adalah :

- a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use Case*, aktivitas/ sarana yang disiapkan oleh bisnis/sistem.

Gambar II.1. di bawah ini merupakan salah satu contoh bentuk diagram *use case*. Prabowo Pudji Widodo, Herlawati (2011 : 15-16).

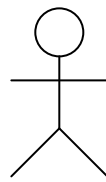


Gambar II.7. Diagram *Use Case*

(Sumber : Prabowo Pudjo Widodo, Herlawati; 2011:17)

a. Aktor

Menurut Chonoles (2003 :17) menyarankan sebelum membuat use case dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder* untuk melihat gambar aktor, lihat pada gambar II.2. sebagai berikut :

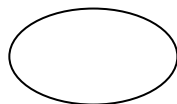


Gambar II.8. Aktor

(Sumber : Prabowo Pudjo Widodo, Herlawati ; 2011:17)

b. Use Case

Menurut Pilone (2005 : 21) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. Sedangkan menurut Whitten (2004 : 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario) baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval* untuk melihat gambar simbol *use case*, lihat pada gambar II.3. sebagai berikut :

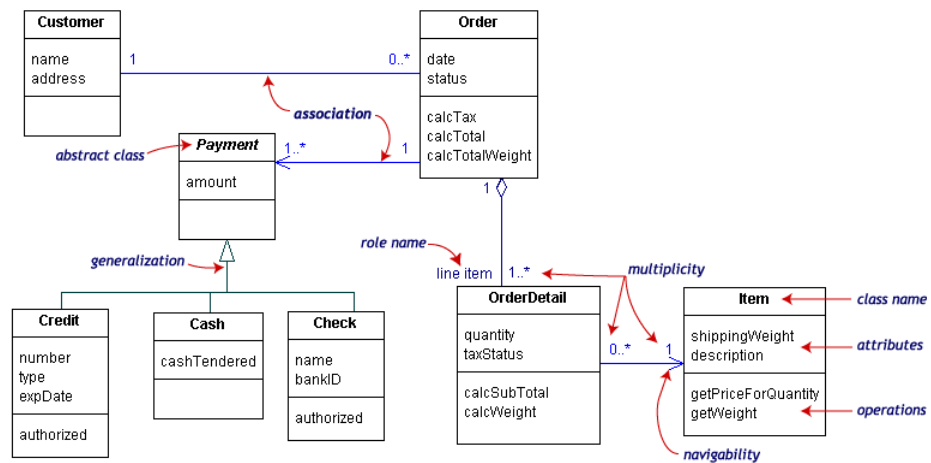


Gambar II.9. Simbol Use Case

(Sumber : Prabowo Pudjo Widodo, Herlawati; 2011:22)

2. Diagram Kelas (*Class Diagram*)

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini *forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model. Probowo Pudji Widodo, Herlawati (2011 : 37) untuk melihat gambar hubungan diagram kelas dengan diagram *uml* lainnya, lihat pada gambar II.4. sebagai berikut :



Gambar II.10. Class Diagram

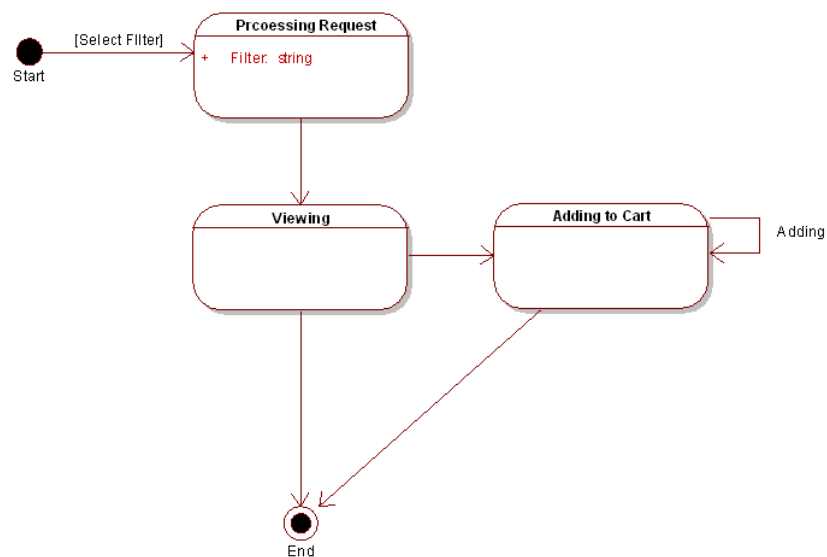
(Sumber : Sri Dharwiyanti; 2006 : 6)

3. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard*

yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring.

Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah. Sri Dharwiyanti (2006: 7)



Gambar II.11. Statechart Diagram

(Sumber : Sri Dharwiyanti; 2006 : 7)

4. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini

menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian internal misalnya penggajian tiap jumat sore.

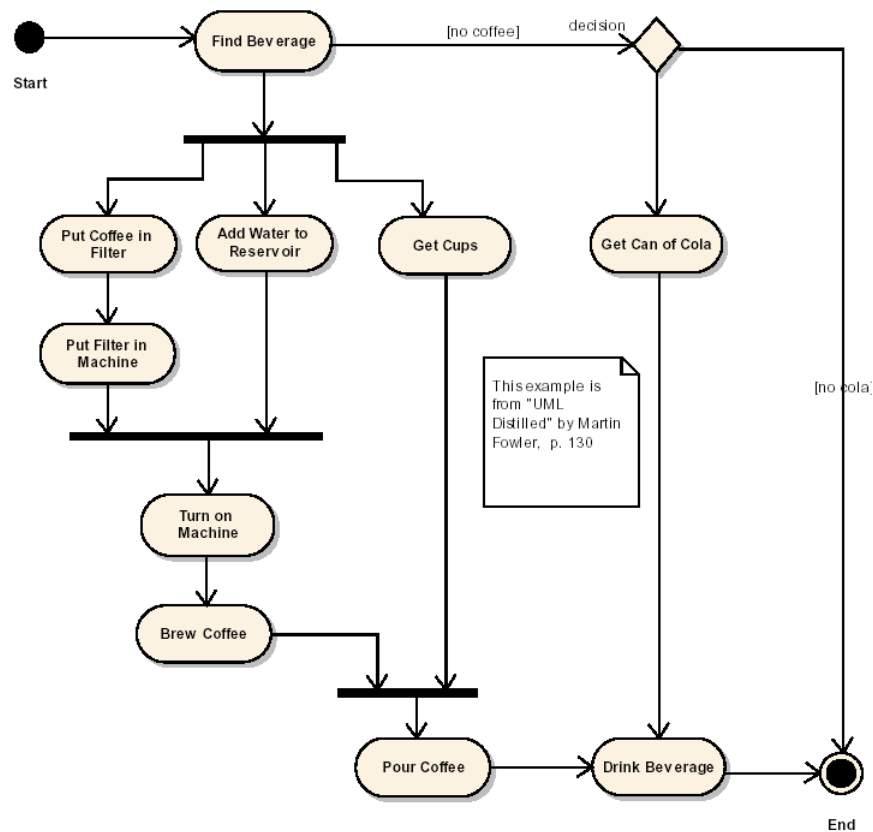
Prabowo Pudjo Widodo, Herlawati (2011 : 143-145)

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi nelakukan langka sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

- a. Fungsi Matematika
- b. Pemanggilan Perilaku
- c. Pemrosesan Data

Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classifier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan dengan model proses bisnis, informasi itu biasanya disebut *process-relevant data*. Aktivitas diharapkan dapat digunakan ulang dalam suatu aplikasi, sedangkan aksi biasanya *specific* dan digunakan hanya untuk aktivitas tertentu.

Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya. Untuk melihat gambar aktivitas sederhana tanpa rincian, lihat pada gambar II.12. sebagai berikut :



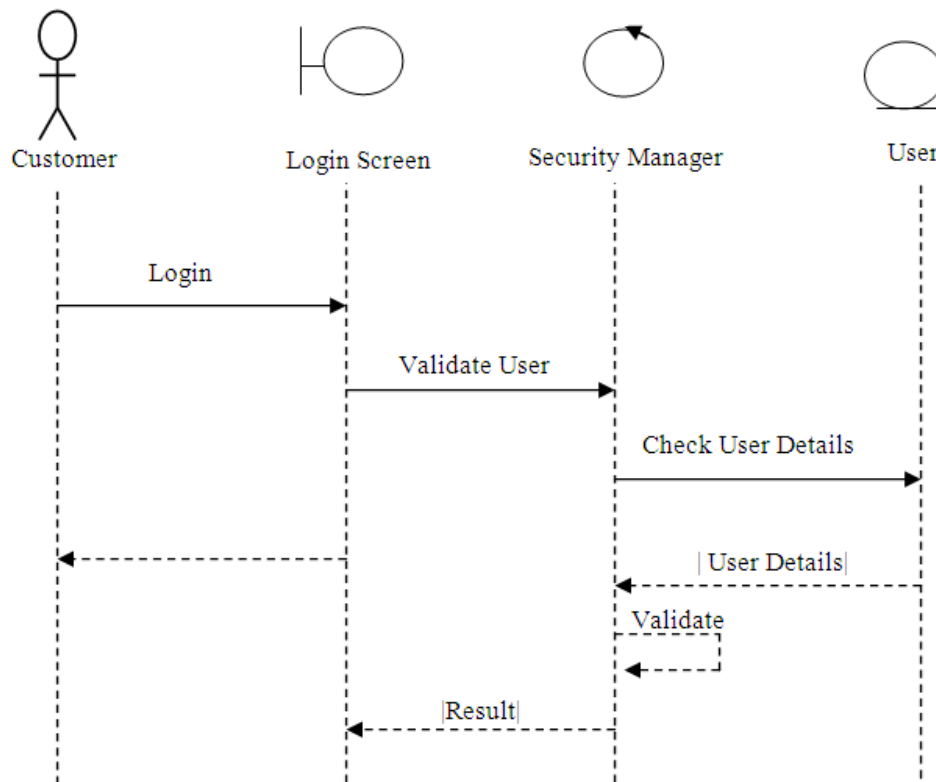
Gambar II.12. Activity Diagram

(Sumber : Sri Dharwiyanti; 2006 : 7)

5. Sequence Diagram

Menurut Douglas (2004 :174) menyebutkan ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*) dan diagram komunikasi (*communication diagram*).

Menurut Pilone (2005 : 174) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.13. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nantinya. (Prabowo Pudji Widodo, Herlawati; 2011 : 174-175)



Gambar II.13. Diagram Urutan

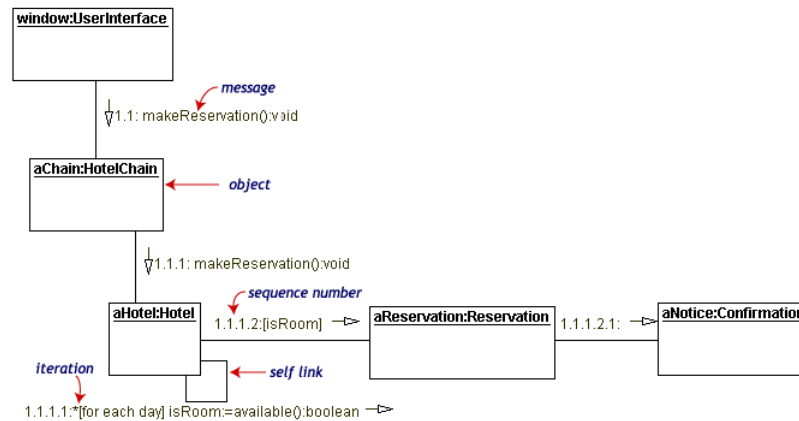
(Sumber : Prabowo Pudjo Widodo, Herlawati; 2011:175)

6. Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.

Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.

Sri Dharwiyanti (2006: 9)



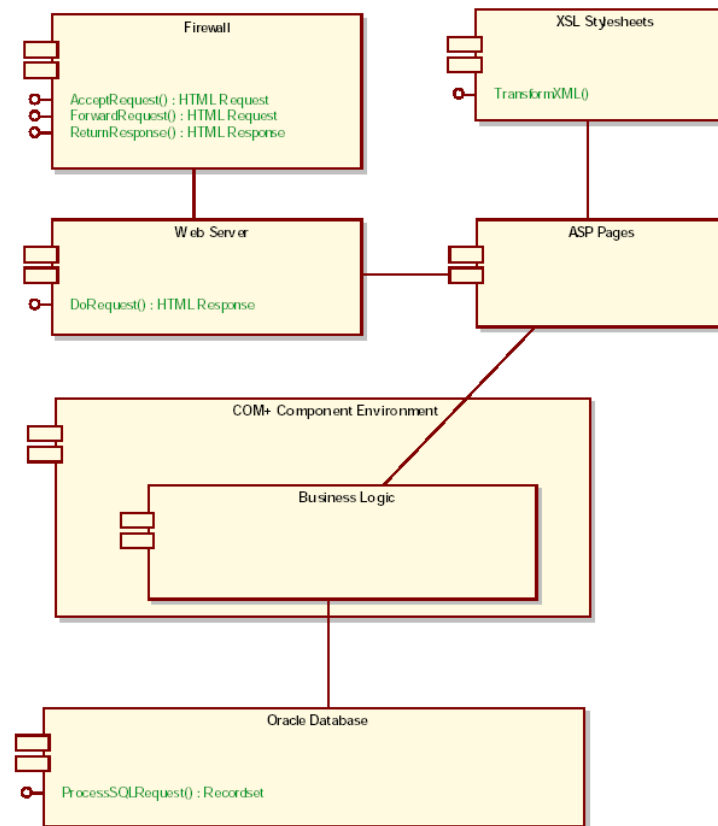
Gambar II.14. Collaboration Diagram

(Sumber : Sri Dharwiyanti; 2006 : 9)

7. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain. Sri Dharwiyanti (2006: 8)



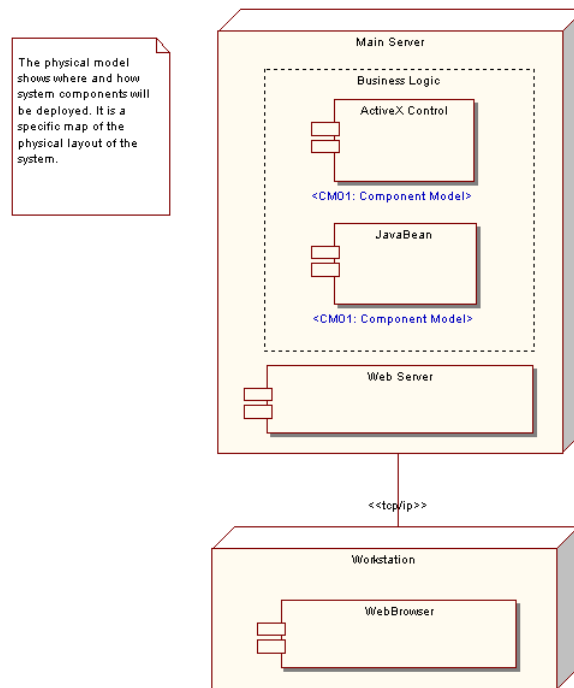
Gambar II.15. Component Diagram

(Sumber : Sri Dharwiyanti; 2006 : 10)

8. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya

TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini. Sri Dharwiyanti (2006: 10)



Gambar II.16. Deployment Diagram
(Sumber : Sri Dharwiyanti; 2006 : 11)