

BAB II

TINJAUAN PUSTAKA

II.1. Multimedia

Multimedia dapat diartikan sebagai penggunaan beberapa media yang berbeda-beda untuk menggabungkan dan menyampaikan informasi dalam bentuk teks, audio, grafik, animasi dan video.

Beberapa definisi :

- a. Kombinasi dari komputer dan video.
- b. Kombinasi dari tiga elemen yaitu suara, gambar dan teks.
- c. Kombinasi dari paling sedikit dua media input dan output. Media ini dapat berupa audio (suara, musik) animasi, video, teks, grafik dan gambar.
- d. Alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang dapat mengkombinasikan teks, grafik, animasi, audio dan video.
- e. Multimedia dalam konteks komputer adalah pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, audio dan video dengan menggunakan *tool* yang memungkinkan pemakai berinteraksi dan berkomunikasi. (Antonius Rachmat & Alphone Roswanto; 2006:23).

Sistem multimedia dapat dibagi menjadi :

- a. Sistem Multimedia *Stand Alone*

Sistem ini berarti merupakan sistem komputer multimedia yang memiliki minimal *storage* (*harddisk*, CD-ROM/DVD-ROM/CD-RW/DVD-RW), alat

input (*keyboard, mouse, scanner, mic*), dan output (*speaker, monitor, LCD Proyektor*), VGA dan *soundcard*.

b. Sistem Multimedia Berbasis Jaringan

Sistem ini harus terhubung melalui jaringan yang mempunyai *bandwidth* yang besar. Perbedaannya adalah adanya *sharing* sistem dan pengaksesan terhadap sumber daya yang sama. Contoh : *video conference* dan *video broadcast*. Permasalahan : bila *bandwidth* kecil maka akan terjadi kemacetan jaringan, *delay* dan masalah infrastruktur yang belum siap.

II.2. Jenis-Jenis Media Pembelajaran

Menurut Azhar Arsyad (2011), berdasarkan perkembangan teknologi tersebut, media pengajaran dikelompokkan kedalam empat bagian, yaitu :

a. Media Hasil Teknologi Cetak

Teknologi cetak adalah cara untuk menghasilkan atau menyampaikan materi, seperti buku dan materi *visual* statis terutama melalui proses pencetakan mekanis atau fotografis. Kelompok media hasil teknologi cetak meliputi teks, grafik, foto atau representasi fotografik dan reproduksi. Materi cetak dan visual merupakan dasar penggabungan materi dan pengajaran lainnya. Teknologi ini menghasilkan materi dalam bentuk salinan tercetak.

b. Media Hasil Teknologi Audio-Visual

Teknologi Audio-Visual adalah cara menghasilkan atau menyampaikan materi dengan menggunakan mesin-mesin mekanis dan elektronik untuk mengajukan pesan-pesan audio dan visual. Pengajuan melalui audio-visual

bercirikan pemakaian perangkat keras selama proses belajar, seperti mesin *proyektor film*, *tape recorder* dan *proyektor visual*.

c. Media Hasil Teknologi Berbasis Komputer.

Teknologi berbasis komputer merupakan cara menghasilkan atau menyampaikan materi dengan menggunakan gambar-gambar yang berbasis *microprocessor*. Berbeda antara media yang dihasilkan oleh teknologi berbasis komputer dengan dua teknologi lainnya adalah karena informasi/materi disampaikan dalam bentuk digital, bukan dalam bentuk cetakan atau *visual*. Berbagai jenis aplikasi teknologi berbasis komputer dalam pengajaran umumnya dikenal sebagai *Computer Assisted Instruction* (pengajaran berbantuan komputer). Aplikasi tersebut meliputi *drills* dan *practice* (latihan untuk membantu siswa menguasai materi yang telah dipelajari sebelumnya), tutorial (penyajian materi pelajaran secara bertahap), permainan dan simulasi (latihan mengaplikasikan pengetahuan dan keterampilan yang baru dipelajari).

d. Media Hasil Gabungan Teknologi Cetak dan Komputer

Teknologi hasil gabungan adalah cara untuk menghasilkan dan menyampaikan materi yang menggabungkan pemakaian beberapa bentuk media yang dikenalkan komputer. Media pembelajaran yang akan dibuat dalam tulisan ini adalah media pembelajaran berbasis komputer.

II.3. Peranan Komputer Dalam Meningkatkan Mutu Pendidikan

Perkembangan teknologi informasi (TI) yang sangat pesat merupakan potensi untuk meningkatkan kualitas pendidikan. Internet sebagai anak kandung

dari teknologi informasi menyimpan informasi tentang segala hal yang tidak terbatas, yang dapat digali untuk kepentingan pengembangan pendidikan. Dengan internet belajar tidak lagi dibatasi ruang dan waktu. Keberadaan teknologi informasi bagi dunia pendidikan berarti tersedianya saluran atau sarana yang dapat dipakai untuk menyiarkan program pembelajaran baik secara searah maupun secara interaktif. Pemanfaatan teknologi informasi ini penting mengingat kondisi geografis Indonesia secara umum berada pada daerah pegunungan yang terpencar kedalam banyak pulau-pulau. Dengan adanya teknologi informasi memungkinkan diselenggarakannya pendidikan jarak jauh, sehingga memungkinkan terjadinya pemerataan pendidikan diseluruh wilayah bumi Indonesia, baik yang sudah dijangkau transportasi darat maupun yang belum dapat dijangkau transportasi darat. Dengan demikian pemanfaatan teknologi informasi dalam pendidikan mempunyai arti penting terutama dalam rangka pemerataan pendidikan dan peningkatan kualitas serta efektifitas penyelenggaraan pendidikan di Indonesia. (Ali Muhtadi; 2006:4).

II.4. Aspek-Aspek Yang Mempengaruhi Minat Belajar

Motivasi belajar mahasiswa dapat didefinisikan sebagai suatu keadaan dalam diri mahasiswa yang mendorong dan mengarahkan perilakunya kepada tujuan yang ingin dicapainya dalam mengikuti pendidikan tinggi. Idealnya tujuan mahasiswa dalam mengikuti pendidikan tinggi adalah untuk menguasai bidang ilmu yang dipelajarinya, sehingga dalam mempelajari setiap bahan pembelajaran, mahasiswa terdorong untuk menguasai bahan pembelajaran tersebut dengan baik,

dan bukan hanya untuk sekedar lulus meski hanya dengan nilai sangat baik sekalipun. Meski secara konseptual tidak ada perbedaan antara menguasai bahan pembelajaran dengan baik dengan mendapat nilai baik untuk bahan pembelajaran tersebut, namun dalam dunia pendidikan tinggi swasta di Indonesia dewasa ini, seorang mahasiswa yang lulus dalam suatu bahan pembelajaran dengan nilai baik, belum tentu menguasai bahan pembelajaran tersebut dengan baik. Sebaliknya jika seorang mahasiswa mampu menguasai suatu bahan. (Arko Pujadi;2007).

Motivasi belajar mahasiswa diindikasikan dari kepemilikan buku wajib yang dianjurkan dosen, keseriusan dalam mengikuti perkuliahan, frekuensi bertanya pada dosen, kerajinan dalam mengerjakan tugas, sikap dalam perkuliahan serta tingkat kesadaran dalam kegiatan perkuliahan. Meski demikian motivasi belajar tersebut juga dapat dikatakan tinggi, mengingat mahasiswa hanya terkadang memiliki buku wajib dan bertanya pada dosen. Demikian pula, mereka kadang-kadang masih suka ngobrol dan berbuat berisik didalam kelas pada saat perkuliahan sedang berlangsung. Dengan karakteristik seperti itu, maka dapatlah dikatakan bahwa motivasi belajar mahasiswa fakultas ekonomi Universitas Bunda Mulia tidak rendah tetapi tidak juga tinggi atau juga bisa disebut kurang tinggi untuk menimbulkan konotasi bahwa motivasi belajar tersebut masih perlu ditingkatkan.

II.5. Teori Pembelajaran Konvensional

Seorang guru dituntut untuk menguasai berbagai model pembelajaran, dimana melalui model pembelajaran yang digunakan akan dapat memberikan nilai

tambah bagi siswa atau peserta didik. Selanjutnya yang tidak kalah penting dari proses pembelajaran adalah hasil belajar yang optimal atau maksimal. Banyak terdapat model pembelajaran, salah satunya pembelajaran konvensional.

Pembelajaran konvensional adalah metode pembelajaran tradisional atau disebut juga dengan metode ceramah. Maka pada dasarnya pembelajaran konvensional merupakan pembelajaran yang tidak dapat menarik siswa untuk belajar lebih aktif, karena guru hanya berceramah kepada siswa atau peserta didiknya. Ciri-ciri pembelajaran konvensional atau tradisional yaitu siswa adalah penerima secara pasif, siswa belajar secara individual, pembelajaran sangat abstrak dan teoritis, perilaku dibangun atas dasar kebiasaan, keterampilan dikembangkan atas dasar latihan, hasil belajar hanya diukur dengan tes, dan pembelajaran hanya terjadi di dalam kelas. (Kokom Komalasari; 2010:18).

II.6. Hewan Vertebrata dan Invertebrata

Berdasarkan ada atau tidaknya tulang belakang, *kingdom Animalia* (dunia hewan) dibedakan menjadi Vertebrata dan Invertebrata/Avertebrata.

a. Hewan Bertulang Belakang (Vertebrata)

Semua hewan yang memiliki tulang belakang digolongkan kedalam subfilum Vertebrata. Hewan vertebrata memiliki struktur tubuh dan organ tubuh yang lebih kompleks. Selain memiliki tulang belakang, Vertebrata memiliki rangka dalam yang terdiri dari tulang-tulang. Vertebrata adalah kelompok hewan bertulang belakang atau disebut pula hewan tingkat tinggi.

Filum Vertebrata dibedakan menjadi kelompok ikan (*Pisces*, terdiri dari beberapa kelas), kelas amfibi (*Amphibia*), kelas reptilia (*Reptilia*), kelas burung (*Aves*), dan kelas mamalia (*Mamalia*).

Semua Vertebrata berkembang biak secara seksual. Hewan jantan memiliki testis yang menghasilkan sperma. Hewan betina mempunyai ovarium yang menghasilkan sel telur. Bersatunya sel telur dan sperma disebut pembuahan (*fertilisasi*) yang menghasilkan satu sel zigot. Selanjutnya, zigot berkembang menjadi embrio dan akhirnya menjadi individu baru.

Proses pembuahan vertebrata dibedakan menjadi dua macam, yaitu pembuahan di luar tubuh (*fertilisasi eksternal*) dan di dalam tubuh (*fertilisasi internal*). Pada ikan dan amfibi terjadi pembuahan diluar tubuh. Pada reptilia, burung dan mamalia terjadi pembuahan didalam tubuh.

b. Hewan Tak Bertulang Belakang (Invertebrata)

Invertebrata adalah kelompok hewan yang tak bertulang belakang atau disebut pula hewan tingkat rendah. Avertebrata atau Invertebrata terdiri dari beberapa filum, antara lain *Porifera* (hewan berpori), *Cnidaria*, *Ctenophora*, *Platyhelminthes* (cacing pipih), *Nematoda* (cacing gilik), *Annelida* (cacing gelang), *Echinodermata* (hewan kulit duri), *Mollusca* (hewan lunak), dan *Arthropoda* (hewan kaki berbuku-buku). (Istamar Syamsuri;2006).

II.7. Visual Basic .NET 2005

Visual Basic .NET 2005 (VB.NET 2005) atau biasa disebut Visual Basic 8 adalah teknologi pemrograman *Microsoft* yang dapat digunakan untuk membuat

aplikasi di lingkungan kerja berbasis windows. Visual Basic .NET 2005 adalah pengembangan dari Visual Basic sebelumnya. Kelebihan VB.NET 2005 terletak pada tampilannya yang lebih canggih dibandingkan edisi Visual Basic sebelumnya. Selain memiliki kelebihan VB .Net 2005 juga memiliki kekurangan. Kekurangan VB .NET 2005 yang terlihat jelas adalah beratnya aplikasi ini apabila dijalankan di komputer yang memiliki spesifikasi sederhana.

Komponen-komponen lingkungan kerja VB .NET 2005 adalah :

1. *Toolbox* adalah bagian yang berguna sebagai tempat meletakkan objek-objek yang menjadi alat untuk memprogram. Dalam istilah pemrograman, yang dimaksud alat tersebut adalah sebuah *Class*. Objek-objek yang tersedia di VB .NET 2005 sangat banyak seperti *Textbox*, *ComboBox*, *ListBox* dan lain sebagainya.
2. *Solution Explorer* adalah bagian yang berfungsi melihat item-item penyusun sebuah proyek atau *solution*. Sebuah proyek dapat disusun oleh berbagai macam item, yaitu :
 - a. *Form* dan kodenya, memiliki ekstensi *.vb*
 - b. *Class*, juga memiliki ekstensi *.vb*
 - c. *Module*, juga memiliki ekstensi *.vb*
 - d. *Class Diagram*, memiliki ekstensi *.cd*
 - e. *File XML*, memiliki ekstensi *.xml*
 - f. *File proyek*, memiliki ekstensi *.vbproj*
 - g. *File laporan Cristal Report*, memiliki ekstensi *.rpt*

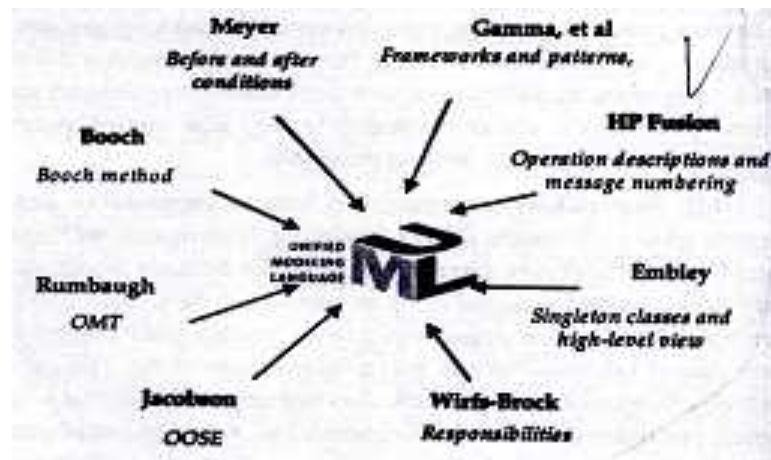
3. *Properties* berguna untuk melihat nilai properti dari sebuah *objek/class*. Misalnya properti *Name*, *Text* dan lain sebagainya.
4. Area Kerja adalah jendela yang berguna untuk melakukan kegiatan pengisian kode ketika berupa tampilan kode (*code view*) dan juga tempat untuk mengatur desain form ketika berupa tampilan desainer (*designer view*).
5. *Main Menu* berguna sebagai tempat mengakses fungsi-fungsi di VB .NET.
6. *Toolbar* berguna sebagai *shortcut* bagi fungsi-fungsi yang sering diakses di *Main Menu*. Berbeda dengan *Main Menu* yang berupa menu yang terjulur kebawah. *Toolbar* berwujud urutan tombol-tombol dengan ikon berupa gambar.
7. *Server Explorer* adalah bagian yang berfungsi sebagai tempat untuk *manajemen database*. Bagian ini berguna jika VB .NET sedang digunakan untuk memprogram *database*. (Andi ; 2006).

II.8. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia perkembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi perkembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan dengan baik (Munawar ; 2005 : 17).

UML merupakan kesatuan bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan interatif, yaitu: identifikasi kelas-kelas dan objek-objek, identifikasi semantik dari hubungan objek dan kelas tersebut, perincian *interface* dan implementasi. Keunggulan metode Booch adalah pada detail dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur dan pemodelan *entity-relationship*. Tahapan utama dalam metodologi ini adalah analisis, disain sistem, desain objek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO. Metode OOSE dari Jacobson lebih memberi penekanan dan *use case*. OOSE memiliki tiga tahapan yaitu membuat model *requirement* dan analisis, desain dan implementasi dan model pengujian (test Model). Keunggulan metode ini adalah mudah dipelajari karena memiliki notaasi sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Dengan UML, metode Booch, OMT dan OOSE digabungkan dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam dari pada metode lainnya. Unsur-unsur yang membentuk UML ditunjukkan dalam Gambar II.1.

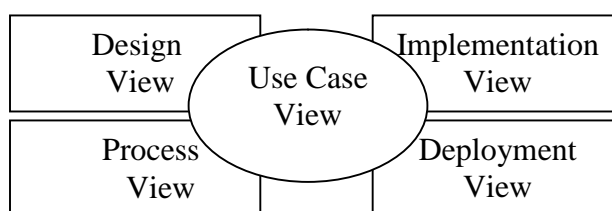


Gambar II.1. Unsur-unsur yang membentuk UML

Sumber : Pemodelan Visual dengan UML, Munawar, 2005 : 18

UML adalah hasil kerja dari konsorsium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam OOAD (*Object Oriented Analysis dan Design*). UML tidak hanya domain dalam penotasian dilingkungan OO tetapi juga populer di luar lingkungan OO. Ada tiga karakter penting yang melekat di UML yaitu sketsa, cetak biru dan bahasa *pemrograman*. Sebagai sebuah sketsa UML bisa berfungsi sebagai sebuah cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi detail tentang coding program (*Forward engineering*) atau bahkan membaca program dan mengimplementasikannya kembali ke dalam diagram (*reverse engineering*). *Reverse engineering* sangat berguna pada situasi dimana kode program yang tidak terdokumentasi asli hilang atau bahkan belum pernah dibuat sama sekali. Sebagai bahasa pemrograman, UML dapat diterjemahkan diagram yang ada di UML menjadi kode program siap untuk dijalankan.

UML dibangun atas model 4+1 *view*. Model ini didasarkan pada fakta bahwa struktur sebuah sistem dideskripsikan dalam *view* dimana salah satu diantaranya *use case view*. *use case view* ini memegang peran khusus untuk mengintegrasikan *content* ke *view* yang lain. Model 4+1 *view* ditunjukkan pada gambar II.2.



Gambar II.2. Model 4+1 View

Sumber : Pemodelan Visual dengan UML, Munawar, 2005 : 20

Kelima *view* tersebut tidak berhubungan dengan diagram yang dideskripsikan di UML. Setiap *view* berhubungan dengan perspektif tertentu dimana sistem akan diuji. *View* yang berbeda akan menekankan pada aspek yang berbeda dari sistem yang mewakili tentang sistem bisa dibentuk dengan menggabungkan informasi-informasi yang ada pada kelima *view* tersebut.

Use case view mendefinisikan perilaku eksternal sistem. Hal ini menjadi daya tarik bagi *end user*, analis dan tester. Pandangan ini mendefinisikan kebutuhan sistem karena mengandung semua *view* yang lain yang mendeskripsikan aspek-aspek tertentu dari peran dan sering dikatakan yang mendrive proses perkembangan perangkat lunak.

Design view mendeskripsikan struktur logika yang mendukung fungsi-fungsi yang dibutuhkan di *use case*. *Design view* ini berisi definisi komponen program, class-class utama bersama-sama dengan spesifikasi data, perilaku dan interaksinya. Informasi yang terkandung di *view* menjadi pergantian para progremer karena menjelaskan secara detil bagaimana fungsionalitas sistem akan diimplementasikan.

Implementasi *view* menjelaskan komponen-komponen visi yang akan dibangun. Hal ini berbeda dengan komponen logic yang dideskripsikan pada *design view*. Termasuk disini diantaranya *file exe*, *library* dan *database*. Informasi yang ada di *view* dan integrasi sistem.

Proses *view* berhubungan dengan hal-hal yang berkaitan dengan *concurrency do* dalam sistem. Sedangkan *deployment view* menjelaskan bagaimana komponen-komponen fisik didistribusikan ke lingkungan fisik seperti jaringan komputer dimana sistem akan dijalankan. Kedua *view* ini menunjukkan kebutuhan non fungsional dari sistem seperti toleransi kesalahan dan hal-hal yang berhubungan dengan kinerja (Munawar;2005:17-21).

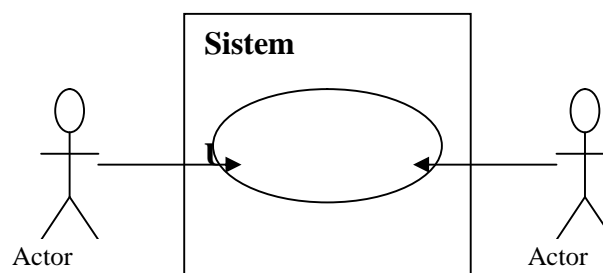
II.8.1. Use Case Diagram

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara deskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem yang disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem

yang lain, perangkat keras dan urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian *scenario* yang digabungkan bersama-sama oleh pengguna tujuan umum pengguna.

Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan *actor*. *Actor* adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model *use case* adalah bagian dari model *requirement*. Termasuk disini adalah problem domain object dan penjelasan tentang *user interface*. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari *perspektif user*.

Notasi *use case* menunjukkan 3 aspek dari sistem yaitu *actor use case* dan *system / sub system boundary*. *Actor* mewakili peran orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*. Ilustrasi *actor*, *use case* dan *system* ditunjukkan pada gambar II.3.



Gambar II.3. Use Case Diagram

Sumber : Pemodelan Visual dengan UML, Munawar, 2005 : 64

Untuk mengidentifikasi *actor*, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. *Actor* adalah *abstraction* dari orang dan sistem yang lain mengaktifkan fungsi dari target

sistem. Orang atau sistem bila muncul dalam beberapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan use case, tetapi tidak memiliki kontrol atas use case.

Use case adalah abstraksi dari interaksi antara sistem dan *actor*. Oleh karena itu sangat penting untuk memilih abstraksi yang cocok. Use case dibuat berdasarkan keperluan actor. Use case harus merupakan 'apa' yang dikerjakan software aplikasi, bukan 'bagaimana' software aplikasi mengerjakannya. Setiap use case harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan actor. Namun use case boleh terdiri dari beberapa kata dan tidak boleh ada dua use case yang memiliki nama yang sama (Munawar ; 2005 : 63-66).

II.8.2. Class Diagram

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Kelas memiliki apa yang disebut *Atribut* dan metode atau operasi :

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Susunan kelas suatu sistem yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main, kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem, kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case*, kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.
4. Kelas yang diambil dari pendefinisian data, kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

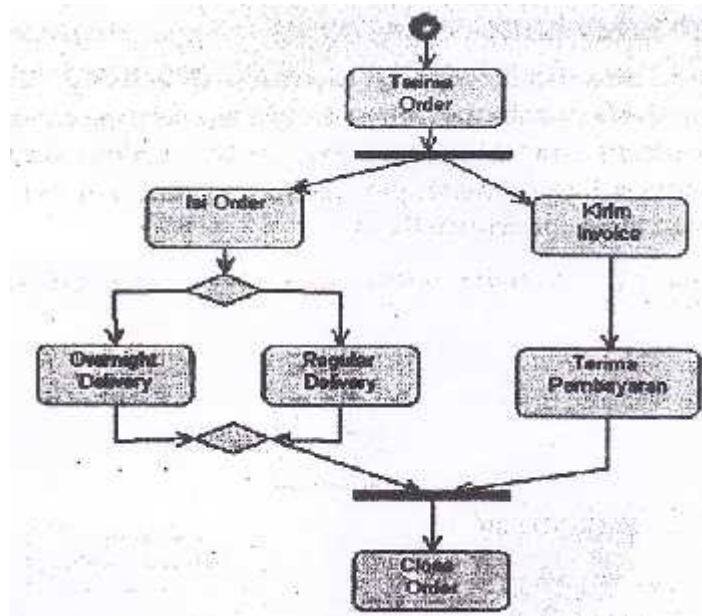
Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan yang lain dalam sebuah kelas. Sebagai aturan secara umum maka sebuah metode yang dibuat harus memiliki kadar *cohesion* yang kuat dan kadar *coupling* yang lemah (Rosa A.S dan M. Shalahuddin ; 2011 : 122-123).

II.8.3. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaanya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak

bisa(Munawar ; 2005 : 87). Contoh *activity diagram* sederhana ditunjukkan pada gambar II.4.



Gambar II.4. Contoh Activity Diagram Sederhana

Sumber : Pemodelan Visual dengan UML, Munawar, 2005 : 111

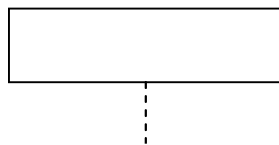
II.8.4. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sebuah contoh objek dan pesan yang diletakkan diantara objek-objek ini didalam *use case*.

Komponen utama *Sequence diagram* terdiri dari atas objek yang dituliskan dengan kotak segiempat bernama. *Messege* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical* (Munawar ; 2005 : 109).

1. Objek / *participant*

Objek diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Mereka diatur dalam urutan guna menyederhanakan diagram. Setiap *participant* dihubungkan garis titik-titik yang disebut *lifeline*. Sepanjang *lifeline* ada kotak yang disebut *activation*. *Activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*. *Activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*. Bentuk *participant* dapat dilihat pada gambar II.5.



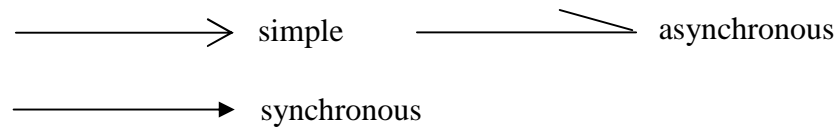
Gambar II.5. Bentuk *Participant*

Sumber : Pemodelan Visual dengan UML, Munawar, 2005 : 88

2. *Message*

Sebuah *message* bergerak dari suatu *participant* ke *participant* yang lain dan dari *lifeline* ke *lifeline* yang lain. Sebuah *participant* bisa mengirim sebuah *message* kepada dirinya sendiri. Sebuah *message* bisa jadi *simple*, *synchronous* atau *asynchronous*. *Message* yang *simple* adalah sebuah perpindahan (transfer), contoh dari satu *participant* ke *participant* yang lainnya. Jika suatu *participant* mengirimkan sebuah *message* tersebut akan ditunggu sebelum di proses dengan urusannya. Namun jika *message asynchronous* yang dikirimkan, maka

jawabannya atas *message* tersebut tidak perlu ditunggu. Simbol *message* pada *sequence diagram* dapat dilihat pada gambar II.6.

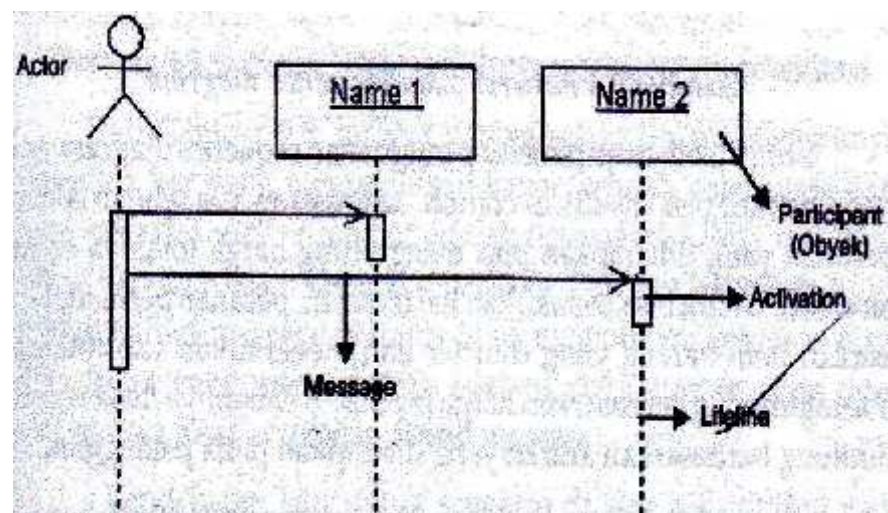


Gambar II.6. Bentuk Message

Sumber : *Pemodelan Visual dengan UML, Munawar, 2005 : 88*

3. Time

Time adalah diagram yang mewakili waktu pada arah vertikal. Waktu dimulai dari atas ke bawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dibanding *message* yang lebih dekat kebawah. Terdapat dua dimensi pada *sequence diagram* yaitu dimensi dari kiri ke kanan menunjukkan tata letak participant dan dimensi dari atas ke bawah menunjukkan lintasan waktu. Simbol-simbol yang ada pada *sequence diagram* ditunjukkan pada gambar II.7.



Gambar II.7. Bentuk Time

Sumber : *Pemodelan Visual dengan UML, Munawar, 2005 : 89*