

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Perancangan Program Aplikasi

Menurut Bahra, perancangan adalah suatu kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari penelitian alternatif sistem yang terbaik.

Menurut Jogiyanto, program adalah kegiatan menulis kode program yang akan dieksekusi oleh komputer. Menurut Binanto, program adalah himpunan atau kumpulan instruksi tertulis yang dibuat oleh *programmer* atau suatu bagian *executable* dari sebuah *software*.

Menurut Jogiyanto, aplikasi adalah penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan ke dalam suatu sarana yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, pekerjaan itu sendiri. (Anis Setyo, Hal:13, 2005)

II.2. Sejarah Permainan Komputer

Manusia telah mengenal dan memainkan permainan sejak dahulu. Di sahara ditemukan sebuah papan permainan yang terbuat dari batu yang mirip seperti *Mancala* yang telah berusia ± 5000 tahun. Menurut *David Fox* dan *Roman Verhosek*, permainan yang populer di Negara-negara *oriental*, sebenarnya telah

ada sejak 2000 SM. Bahkan sebuah permainan yang mirip *Backgamon* seperti *Tabula* dan *Nard* telah tercatat pada *skrip* Romawi kuno .

Pada tahun 1952, seorang mahasiswa *Cambridge University* bernama *Gouglas* membuat permainan *Tic tac toe* dalam versi grafik. Permainan ini dia kembangkan ketika hendak mendemonstrasikan tesisnya tentang interaksi antara manusia dan komputer. Pada tahun 1966 permainan *digital* pertama kali dibuat oleh *Ralph Baer* bersama timnya yang berjumlah 500 orang insinyur. Permainan ini hanya dapat dimainkan dengan komputer seharga 40.000 *Dollars*. Unsur edukasi menjadi tujuan utama dalam permainan ini. Permainan dalam bentuk permainan antara papan dan bola tersebut diperuntukkan untuk membantu pasukan belajar strategi dan melatih kemampuan refleks pemainnya.

Pada tahun 1972, muncul permainan baru yang disebut permainan *Arcade*, yang dipelopori oleh *Nolan Bushnel* dengan permainannya berjudul *Pong*. Mesin untuk memainkan permainan ini disebut mesin *Arcade*. Pemain yang ingin bermain diharuskan untuk memasukkan koin kedalam mesin. Pada hari kedua mesin ini diletakkan pada suatu bar, orang-orang mengantri untuk memainkan permainan *Pong*. Tidak mau tertinggal dengan sistem *Arcade*, perusahaan *game* lain seperti *Magnavox Odyssey*, *Atari 2006*, *Mattel Intelvision*, *Calleco Vision* dan *Nintendo Entertainment*, menciptakan permainan yang dapat dimainkan di rumah. Permainan yang paling menghebohkan orang-orang dengan tampilan grafik dan permainan *play* yang luar biasa pada sistem konsol tersebut adalah *Super Mario Brothers* yang diciptakan oleh *Nintendo*.

Pada perkembangannya, permainan komputer berkembang dengan pesatnya seiring perkembangan perangkat keras yang mendukung. Hal ini dibuktikan dengan program permainan yang lebih kompleks dan tampilan grafik tiga dimensi. (David Fox dan Roman Verhosek, 1-2, 2002)

II.2.1 Jenis Jenis Permainan

Adapun Beberapa jenis-jenis permainan adalah :

1. Olahraga (*Sport*) adalah permainan ini merupakan adaptasi dari kenyataan, membutuhkan kelincahan dan juga strategi dalam memainkannya. Contoh permainan : *Winning Eleven* dan *Basket Ball*.
2. Pertarungan (*Fighting*) adalah permainan yang biasanya ada dua karakter yang bertarung untuk memperoleh kemenangan atau tercapainya suatu misi. Contoh permainan : *Weird Fighters*, *Space Quest* dan *Black Belt*.
3. Petualangan (*Adventure*) adalah sebuah permainan dimana *user* dapat mengeksplorasi 1 tempat atau lebih, melawan musuh yang ditemui, berinteraksi dengan karakter yang ditemui, dan terkadang memecahkan teka-teki. Contoh permainan : *Summoner Saga* dan *Apple Maze*.
4. Tembak-tembakan (*Shooting*) adalah permainan yang bertipe menembak musuh atau sasaran tertentu. Contoh permainan : *Duck Hunt* dan *House of The Dead*.
5. Logika (*Logic*) adalah suatu permainan memecahkan teka-teki yang ada dalam permainan tersebut. Contoh permainan : *Minesweeper* dan *Bejeweled*.

6. Strategi (*Strategy*) adalah jenis ini memerlukan koordinasi dan strategi dalam memainkan permainan ini. Kebanyakan *game* strategi adalah *game* perang. Contoh permainan: *Warcraft* dan *Age of Empires*.
7. Edukasi (*Education*) adalah sebuah permainan ini dibuat dengan tujuan spesifik sebagai alat pendidikan. Contoh permainan : *Bobi Bola* dan *Tracy*.
8. Simulasi (*Simulation*) adalah sebuah permainan ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detil berbagai faktor. Contoh permainan : *The Sims* dan *Roller Coster Tycoon*. (Suindarti, Hal : 5-6, 2011)

II.2.2 Game Berburu Bebek

Game berburu bebek atau *duck hunt* merupakan suatu *game shooting*, *game duck hunt* pertama kali dibuat oleh perusahaan *nintendo* pada akhir 1980-an. Permainan ini terinspirasi dari kebiasaan orang eropa zaman dahulu yang sering berburu bebek liar yang di rawa dan hutan.

II.3. Visual Basic 2008

Visual Basic 2008 merupakan bagian dari kelompok bahasa pemrograman *Visual Studio 2008* yang diproduksi oleh *Microsoft*. *Visual Studio 2008* terdiri dari beberapa bahasa pemrograman diantaranya adalah *Microsoft Visual Studio 2008*, *Microsoft C# 2008*, *Microsoft C++ 2008*, *Microsoft Visual J#*, dan *Visual Web Developer 2008*. *Visual Studio* ini telah mengalami perubahan versi mulai dari *Visual Basic 6.0*, *Visual Studio 2003*, *Visual Studio 2005*, dan yang terakhir adalah *Visual Studio 2008*.

Microsoft Visual Basic 2008 disebut juga *Microsoft Visual Basic 9.0* yang memiliki kelebihan-kelebihan, yaitu *suport* dengan bahasa *query Language-Integrated Query (LINQ)* dan *suport* dengan *database Microsoft SQL Server Compact 3.5*. Selain itu, kelebihan lain adalah memiliki *Object Relational Designer (O/R Designer)* untuk membantu *editing LINQ* ke *SQL* yang akan dihubungkan dengan *database* dan *fiture* lain, seperti *WPF* dan *WCF (Windows Communication Foundation)*. (Hendarayudi, Hal:1-2, 2011)

II.3.1. IDE (Integrated Development Environment) Visual Studio 2008

Untuk mengembangkan aplikasi menggunakan *Visual Basic 2008*, terlebih dahulu adalah mengenal *IDE* dari *Visual Studio 2008*. Pada waktu *Visual Studio 2008* dijalankan, maka akan tampil sebuah tampilan awal atau *Start Page* dari *Visual Studio 2008*.

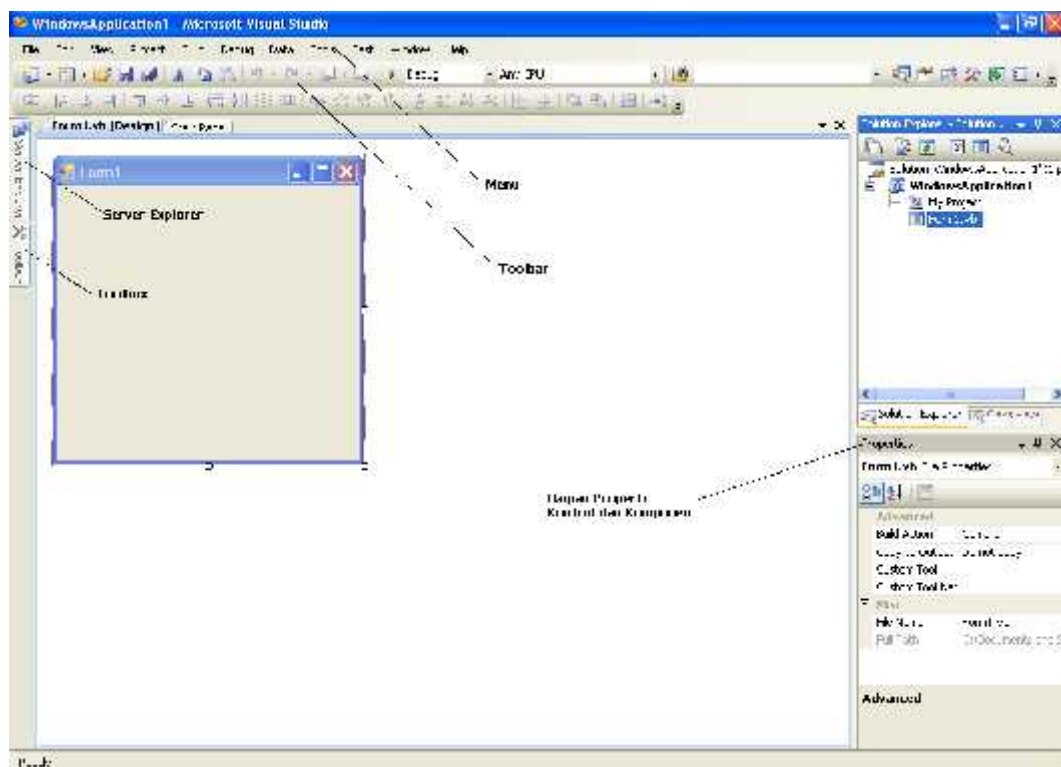


Gambar II.1. Tampilan Awal Visual Studio 2008

Sumber : Hendarayudi, Hal : 2, 2011

Setelah itu akan muncul kotak dialog *New Project*. Pada kotak dialog *New Project* terdapat beberapa pilihan *tool* untuk pengembangan aplikasi, seperti *Visual Basic*, *Visual C#* dan *Visual C++*. Pilih *Visual Basic* kemudian pilih *Windows Form Application*. Beri nama *project* yang akan dibuat pada bagian Nama dan *direktori* tempat menyimpan *project* pada bagian *Location*.

Klik OK pada kotak dialog *New Project* tersebut. Selanjutnya muncul *Visual Basic IDE* tempat untuk membangun aplikasi *Visual Basic 2008*.



Gambar II.2. IDE Visual Studio 2008

Sumber : Hendarayudi, Hal : 3, 2011

Pada *project Visual Basic* untuk *Windows Application* secara *default* telah terdapat sebuah *form*. *Form* tersebut bernama *Form1*. Pada *form* inilah tempat untuk meletakkan kontrol-kontrol atau komponen-komponen untuk membuat

sebuah aplikasi *windows*. *Form* dan kontrol-kontrol dari program aplikasi inilah yang biasanya disebut dengan *GUI* (*Graphical User Interface*) atau antar muka dari program. Jadi *user* akan berinteraksi dengan sebuah program aplikasi melalui *GUI*. Pada *IDE Visual Basic 2008* terdapat *Menu*, *Toolbar*, *Toolbox*, *Server Explorer*, *Solution Explorer* dan *Properties Window*.

II.3.2. Menu

Menu adalah bagian dari *IDE* yang terdiri dari perintah-perintah untuk mengatur *IDE*, mengembangkan, memelihara dan mengeksekusi program. Di dalam *menu*, perintah-perintah dikelompokkan ke dalam beberapa bagian sesuai jenis perintah tersebut.

II.3.3. Toolbar

Toolbar fungsinya sama seperti fungsi dari *menu*, hanya saja pada *toolbar* pilihan-pilihan berbentuk *icon*. Untuk memilih suatu proses yang akan dilakukan, anda tinggal mengklik *icon* yang sesuai dengan proses yang anda inginkan.

II.3.4. Toolbox

Toolbox adalah kumpulan komponen-komponen yang digunakan untuk membuat aplikasi. Untuk membuat objek kontrol dan komponen pada *form* program aplikasi diambil dari kontrol-kontrol yang ada pada *toolbox*. Untuk menampilkan *windows toolbox*, klik pada tombol *toolbox* yang terdapat pada *toolbar*. Tampilan *Toolbox* pada *Visual Basic* dapat dilihat pada gambar II.3.



Gambar II.3. Toolbox

Sumber : Hendarayudi, Hal : 3, 2011

II.3.5. Solution Explorer

Solution Explorer adalah tempat untuk mengatur perintah atau komponen yang terdapat di dalam *form*. Pada *windows solution explorer* terdapat beberapa tombol pada *toolbar* dan *tree* yang berisi daftar *file* yang digunakan dalam *project*. Tampilan *Solution Explorer* pada *Visual Basic* dapat dilihat pada gambar II.4.



Gambar II.4. Solution Explorer

Sumber : Hendarayudi, Hal : 4, 2011

II.3.6. Properties Window

Properties windows adalah tempat untuk mengatur perintah atau komponen yang terdapat di dalam *form*. *Properties window* juga dipakai untuk mengatur properti objek kontrol dan komponen yang dipakai. Dengan *properties window*, Anda dapat mengubah properti yang nantinya akan dipakai sebagai *default* objek kontrol dan komponen pada waktu pertama kali program dieksekusi. Tampilan *Properties Window* pada *Visual Basic* dapat dilihat pada gambar II.5.



Gambar II.5. Properties Window

Sumber : Hendarayudi, Hal : 4, 2011

II.3.7. Form

Form adalah tempat Anda membuat tampilan (*user interface*) untuk program aplikasi Anda. Pada *form*, Anda dapat meletakkan atau menambahkan objek kontrol maupun komponen.

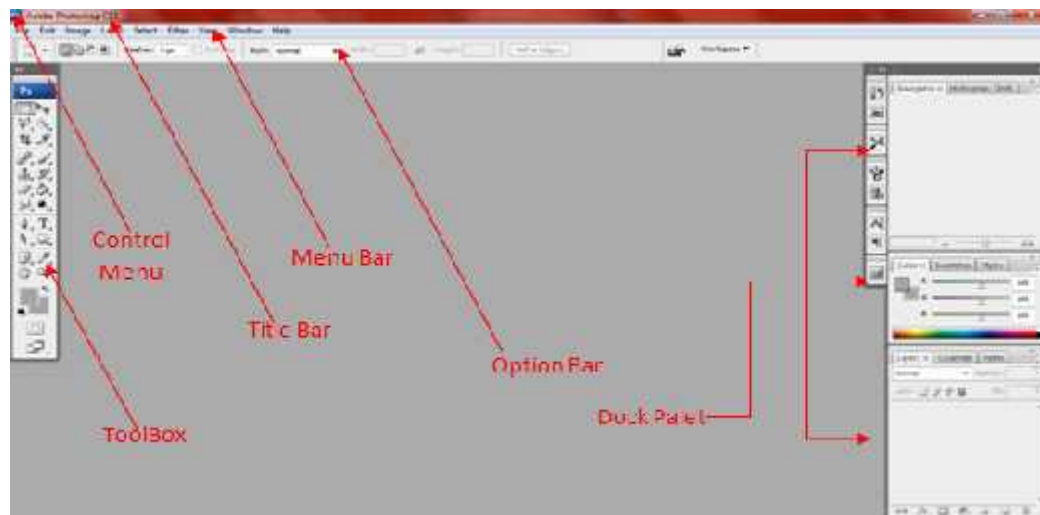
II.3.8. Kode Editor

Kode editor adalah tempat dimana Anda meletakkan atau menuliskan kode program dari program aplikasi Anda. Pada *kode editor* juga terdapat bagian objek dan *event* dari kontrol.

II.4. Adobe Photoshop CS3

Adobe Photoshop CS3 merupakan seri terbaru dari program sebelumnya yaitu *Adobe Photoshop CS2*. Dengan perubahan penampilan yang menarik dan fitur terbarunya, menjadikan *Adobe Photoshop CS3* sebagai sebuah program pengolahan gambar dan photo yang semakin digemari oleh para *designer* dan *Photografer*. Beberapa perubahan dan fitur terbaru memberikan banyak kemudahan dalam hal pengolahan *file* dan *editing* maupun modifikasi sebuah photo. Serta penambahan beberapa fasilitas baru, menjadikan *Adobe Photoshop CS3* sebagai salah satu program pengolahan yang semakin handal. (Agnes Heni T. Sri Mulanto, Hal : 1, 2008).

Pada saat pertama kali Anda menjalankan *Adobe Photoshop CS3*, maka akan tampil jendela program seperti yang terlihat Gambar II.6.



Gambar II.6 Jendela Program *Adobe Photoshop CS3*

Sumber : T.Agnes Heni, Sri Mulanto, Hal : 3, 2008

Penjelasan mengenai gambar di atas dapat dilihat pada Tabel II.1 berikut ini :

Tabel II.1 Bagian-Bagian Adobe Photoshop CS3

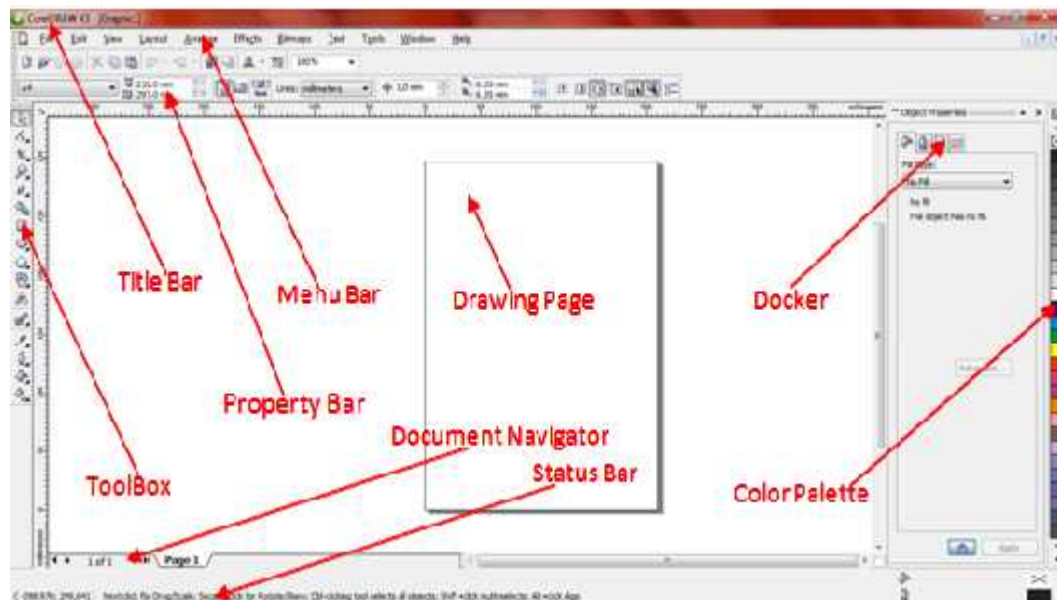
Nama Bagian	Keterangan
<i>Control Menu</i>	<i>Control Menu</i> adalah lambang atau simbol dari aplikasi
<i>Title Bar</i>	<i>Title bar</i> adalah tempat nama program aplikasi yang sedang aktif saat ini dan nama gambar sebelum disimpan.
<i>Menu Bar</i>	<i>Menu bar</i> merupakan batang menu yang berfungsi untuk menampilkan pilihan menu atau perintah untuk mengoperasikan <i>Adobe Photoshop</i> . Misalnya <i>menu File, Edit, Image, Layer, Select, Filter</i> , dan sebagainya.
<i>Option Bar</i>	<i>Option bar</i> digunakan untuk mengubah perilaku <i>tool</i> yang sedang digunakan.
<i>Toolbox</i>	<i>Toolbox</i> merupakan sebuah kotak yang berisi berbagai piranti untuk memanipulasi dan menyunting sebuah tampilan.
<i>Dock palet</i>	<i>Palet</i> dalam <i>Photoshop</i> digunakan untuk mengontrol sifat dan cara kerja dari tombol-tombol yang ada di dalam <i>ToolBox</i> .

Sumber : T.Agnes Heni, Sri Mulanto, 2008

II.5. Corel DRAW X3

CorelDRAW adalah aplikasi *grafis* yang sangat populer di kalangan *desainer* untuk menyelesaikan pekerjaan yang berhubungan dengan suatu desain. Program ini dirancang untuk memenuhi permintaan para *professional grafis*. Bagi yang bekerja di bidang periklanan, percetakan, penerbitan, pembuatan rambu-rambu, pengukir atau bidang *manufactur*. *CorelDRAW* menawarkan alat yang Anda butuhkan untuk membuat garis ilustrasi secara akurat dan layout halaman yang *professional*. *CorelDRAW X3* banyak memberikan kemudahan kepada pemakainya dengan menawarkan beberapa fasilitas baru yang tidak terdapat dalam generasi *CorelDRAW* versi sebelumnya. (A. Taufik Hidayatullah : 2007 : 1)

Pada saat pertama kali Anda menjalankan *CorelDRAW X3*, maka akan tampil elemen lembar kerja program seperti yang terlihat Gambar II.7.



Gambar II.7 Elemen Lembar Kerja *CorelDRAW X3*

Sumber : A. Taufik Hidayatullah, Hal: 4, 2007

Penjelasan mengenai gambar diatas dapat dilihat pada Tabel II.2 berikut ini :

Tabel II.2 Bagian-Bagian *CorelDRAW X3*

Nama Bagian	Keterangan
<i>Title Bar</i>	<i>Title bar</i> adalah tempat nama program aplikasi yang sedang aktif saat ini dan nama gambar sebelum disimpan.
<i>Menu Bar</i>	<i>Menu bar</i> adalah kumpulan perintah yang dapat Anda gunakan untuk membuat dan memanipulasi gambar desain. Misalnya <i>menu File, Edit, View, Layout, Arrange, Effects</i> , dan sebagainya.
<i>Toolbar</i>	<i>Toolbar</i> adalah perintah pada menu bar yang ditampilkan dalam bentuk gambar-gambar kecil yang selanjutnya disebut ikon.
<i>Toolbox</i>	<i>Toolbox</i> adalah salah satu jenis toolbar yang digunakan untuk keperluan membuat atau mengedit obyek/gambar.

<i>Property Bar</i>	Isi bagian ini berubah-ubah sesuai dengan <i>tool</i> yang sedang digunakan.
<i>Drawing Page</i>	Digunakan untuk membuat atau mengedit gambar
<i>Docker</i>	Ekspansi dari beberapa perintah menu. <i>Docker</i> dapat dibuat melayang atau tergabung dengan jendela utama.
<i>Document Navigator</i>	Digunakan untuk mengatur halaman dari dokumen yang memiliki banyak halaman.
<i>Status Bar</i>	<i>Status bar</i> adalah baris bagian paling bawah aplikasi <i>CorelDRAW X3</i> yang menampilkan informasi yang berhubungan dengan desain yang sedang dikerjakan antara lain adalah informasi tentang cara memilih obyek, warna dan <i>type</i> warna, posisi <i>pointer mouse</i> dalam <i>drawing window</i> serta langkah yang dapat dilaksanakan dalam suatu proses.
<i>Color Palette</i>	<i>Color Palette</i> adalah daftar pilihan warna yang tersedia dalam sistem dan berguna untuk mengisi warna <i>fill</i> dan mengubah warna <i>outline</i> obyek yang terpilih.

Sumber : A. Taufik Hidayatullah, 2007

II.7. UML (*Unified Modeling Language*)

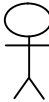


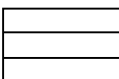
UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk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





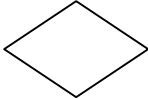


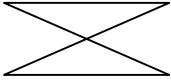
II.7.1 Defenisi Diagram – diagram UML

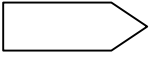
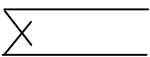

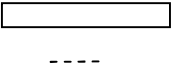
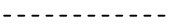
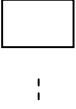
a. Activity Diagram

Activity Diagram adalah teknik untuk mendiskusikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity* diagram banyak mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity* diagram biasa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Di bawah ini Terdapat beberapa simbol-simbol yang sering digunakan pada saat pembuatan UML, dapat dilihat pada Tabel II.3.

Tabel II.3. Simbol-simbol dalam UML

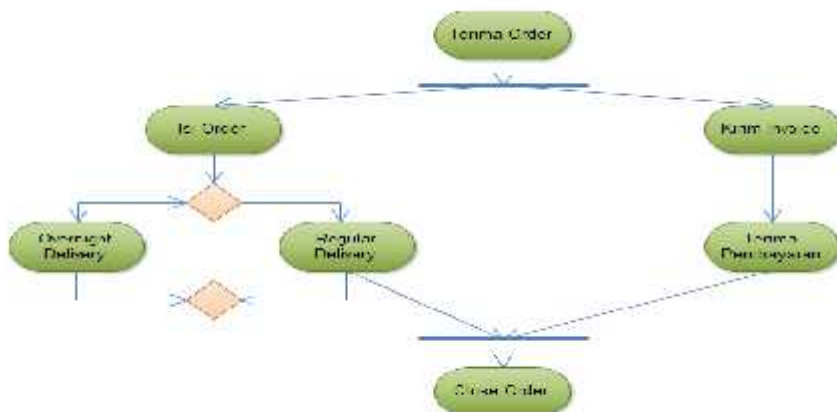
Simbol	Keterangan
	<i>Actor</i> ; mewakili peran orang
	<i>Use Case</i> ; Interaksi antara <i>system</i> dan <i>actor</i>
	<i>System</i>
	<i>Class</i>

	<i>Message</i>
	Titik awal
	Titik akhir
	<i>Activity</i>
	Pilihan untuk pengambilan keputusan
	<i>Fork</i> ; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan <i>pararel</i> menjadi satu
	<i>Rake</i> ; menunjukkan adanya <i>dekomposisi</i>
	Tanda waktu

	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (<i>Flow Final</i>)
	<i>Activation</i> (Mewakili sebuah eksekusi operasi dari <i>participant</i>)
	<i>Lifeline</i> (Setiap <i>participant</i> terhubung dengan garis titik-titik)
	<i>Participant</i>

Sumber : Munawar, *Pemodelan Visual dengan UML* ; 2005 : 09

Dan berikut ini dapat dilihat bentuk dari *Activity Diagram* pada Gambar II.9.

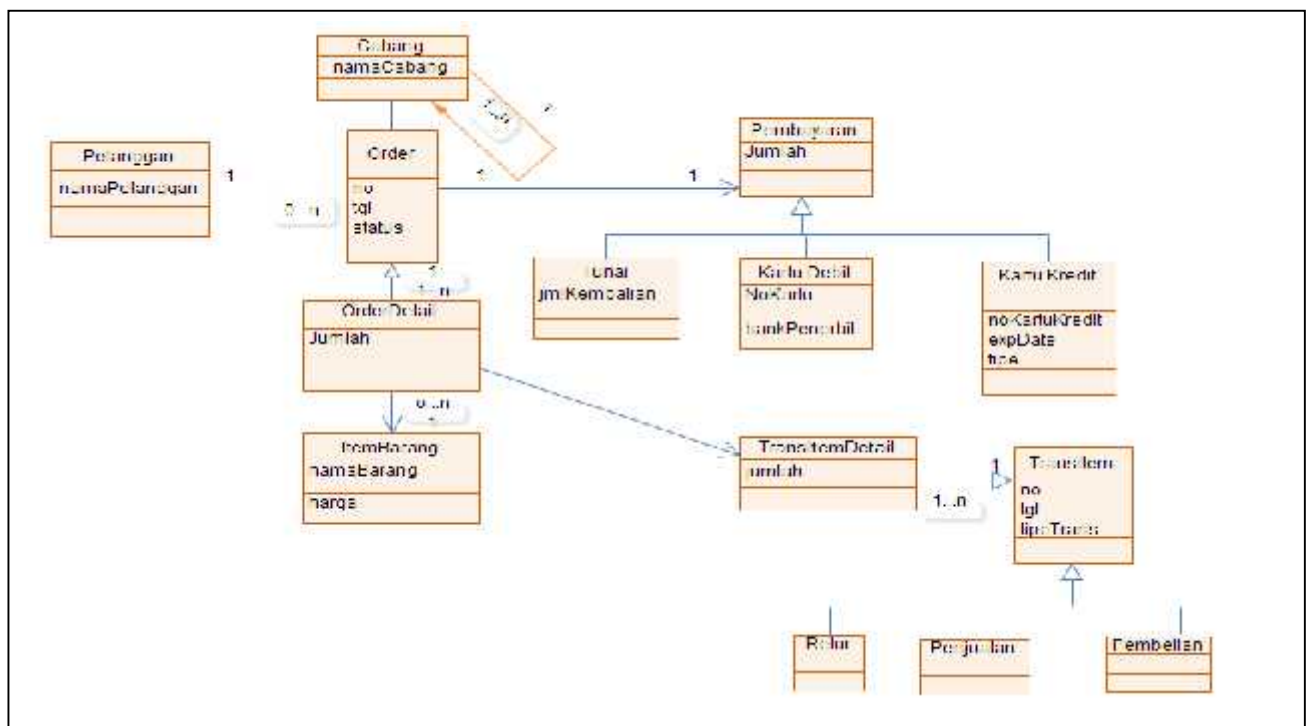


Gambar II.8. Activity Diagram

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

b. Class Diagram

Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena *class* adalah deskripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Disamping itu *class* diagram bisa memberikan pandangan global atas sebuah sistem. Hal tersebut tercemin dari *class-class* yang ada dan relasinya satu sama lainnya. Itulah sebabnya *class* diagram menjadi diagram paling populer di UML. Berikut ini merupakan contoh dari *class diagram* dan dapat dilihat pada Gambar II.10.



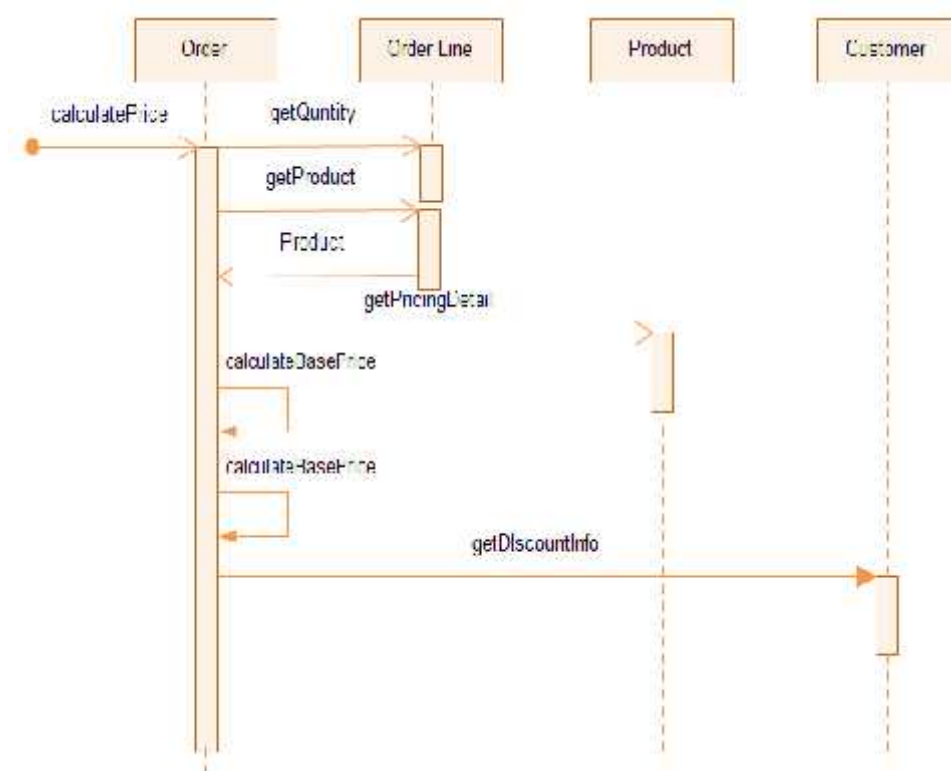
Gambar II.9. Class Diagram VCD Gallery

Sumber :Munawar, *Pemodelan Visual dengan UML* ; 2005 : 220

c. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan)

yang diletakkan diantara obyek-obyek ini di dalam *use case*. Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress* vertikal. Berikut ini merupakan contoh *Sequence diagram*, dapat dilihat pada Gambar II.10



Gambar II.10. Sequence Diagram dengan kontrol tersentralisasi

Sumber :Munawar, *Pemodelan Visual dengan UML* ; 2005 : 91

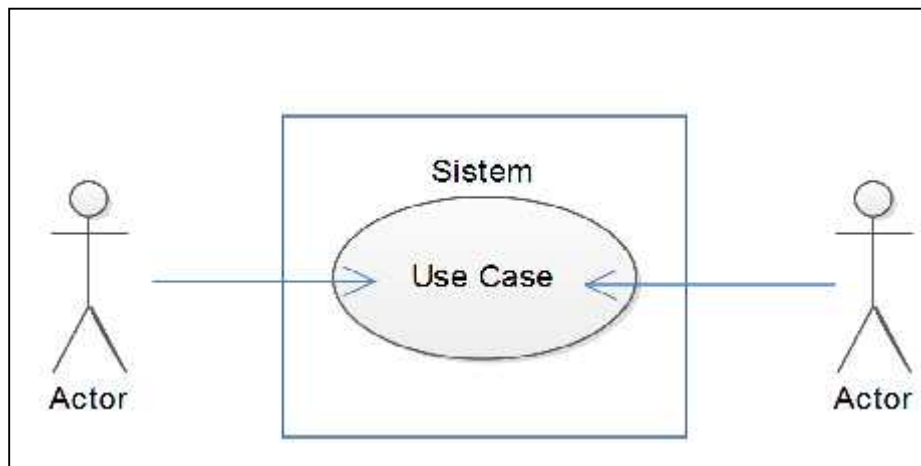
d. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”.

Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah *entitas* manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien dan merancang test *case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan caramenarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Berikut ini merupakan gambar *use case diagram* dapat dilihat pada Gambar II.11.



Gambar II.11. Use Case Model

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