

BAB II

TUNJAUAN PUSTAKA

II.1. Perancangan dan Aplikasi

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem dapat dirancang dalam bentuk bagan alir sistem (*system flowchart*), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem. (Muhammad Fadlan, 2004:151)

Adapun pengertian aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

II.2. Sejarah Game

Game berasal dari kata bahasa Inggris yang memiliki arti dasar Permainan. Permainan dalam hal ini merujuk pada pengertian “kelincahan intelektual” (*intellectual playability*). *Game* juga bisa diartikan sebagai arena keputusan dan

aksi pemainnya. Ada target-target yang ingin dicapai pemainnya.(Afrian, 2014:45)

Menurut *John von Neumann* dan *Oskar Morgenstern* tahun 1944, *game* adalah : “Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi.”

Game atau permainan dapat diklasifikasikan menjadi dua bagian besar *game* fisik dan *game* elektronik. *Game* fisik mungkin sudah sering kita lakukan dalam kehidupan sehari-hari sewaktu masih anak-anak. Seperti lompat tali, petak umpet dan sebagainya. Dan *game* elektronik merupakan fenomena yang sangat menarik saat ini. Bahkan dapat dikatakan bahwa hampir semua kalangan menyukai *game* elektronik. *Electronic Game* atau selanjutnya dapat disebut sebagai *Video Game* pertama sekali ditemukan oleh *Thomas T. Goldsmith Jr* dan *Estle Ray Mann*. Penemuan ini dipatenkan pada Januari 1947. Yang mendasari perkembangannya saat ini adalah ketika mereka menemukan *Cathode-Ray* sebuah tabung *vacuum* yang digunakan sebagai media untuk membuat simulasi kecepatan tembakan dan arah tembakan sebuah roket. Pada Februari 1947, *Christopher Strachey* memulai pengembangannya ke arah pemrograman yang mulai menggunakan memori di mana aplikasinya diterapkan untuk kebutuhan para pilot.

Dan penemuan baru terus berkembang hingga tahun 1959. Namun era perkembangan konsol *game* di mulai setelah masa ini. (Ivan C. Sibero, 2009:9)

II.2.1. Generasi Perkembangan Game

II.2.1.1. Generasi Pertama (1972-1977)

Generasi pertama ini diawali dari ide *Ralph Baer* seorang teknisi televisi. Dia memulai membuat *game* dua pemain sederhana yang bernama *chese* dimana 2 buah titik saling berkejaran. Generasi pertama ini merupakan awal interaktif *game*. Perkembangan penemuan *Baer* ini membuat beberapa penemuan lain mulai ikut bergabung dengannya sehingga ditemukan sebuah mesin baru yang bernama *Magnavox* dan mulai dirilis pada tahun 1972. (Ivan C. Sibero, 2009:10)

II.2.1.2. Generasi Kedua (1976-1983)

Generasi kedua ini sering juga disebut generasi konsol *bit*. Di mana pemrograman *video game* dibuat lebih *advance*. *Farichild Channel F* merupakan sebuah produk yang dirilis sekitar tahun 1976. Tahun 1977 atari memulai konsol baru dengan dasar *CPU*. Merena menamakan produk ini *Video Computer System* (VCS) yang dikenal dengan *ATARI 2600*. Pada masa itu, *ATARI* cukup dikenal masyarakat dibandingkan dengan produk lainnya. *Magnavox* malah mulai dengan konsol *base CPU* pada tahun 1978 dengan produknya *Odyssey 2*. Di Amerika dan Kanada, *Philips Electronics*, sebuah perusahaan yang cukup dikenal hingga saat ini, mulai memproduksi *Philips G7000* dan *Interton VC 4000*. Namun begitu banyak konsol baru yang muncul pada saat itu, *ATARI* tetap masih menjadi magnet *game* bagi masyarakat dengan *arcade game* nya yang cukup dikenal

Space Invaders. Walaupun dengan keterbatasan *hardware* memori yang sangat minim sekitar 2-31 kb, namun perkembangannya cukup pesat. (Ivan C. Sibero, 2009:11)

II.2.1.3. Generasi Ketiga (1983-1992)

Jika sebelumnya perkembangan *game* banyak dilakukan di Eropa dan Amerika, pada generasi ini Jepang juga mulai merilis konsol barunya, *Famicom*, dan berubah nama menjadi *Nintendo*. *Nintendo Entertainment System (NES)* mendominasi pasarnya hingga Amerika Utara, selain *NES*, *Sega Master System* juga berkembang pesat di Eropa dan Brasil. Dan *Atari* pun membuat produk barunya *ATARI 7800*. Pada akhir generasi ini, *Nintendo* memproduksi *Game Boy* yang cukup fenomenal hingga bertahan sampai 15 tahun lebih. (Ivan C. Sibero, 2009:12)

II.2.1.4. Gerenasi Keempat (1987-1996)

Sejarah generasi ini lebih dikenal dengan generasi 16 bit dimana pengembangannya dimulai sekitar Oktober 1987. Generasi awal *PC* yang dikembangkan *Nippon Electric Company's (NEC)*. Produknya yang cukup dikenal di Amerika yaitu *TurboGrafx-16*. Walaupun *NEC* mulai dengan konsol barunya *Nintendo* dan *Sega* masih tetap menjadi konsol terbaik bagi para penggila *game*. (Ivan C. Sibero, 2009:13)

II.2.1.5. Generasi Kelima (1993 – 2002)

Konsol 64 bit mulai muncul pada era ini. Tidak banyak perubahan dari generasi sebelumnya. Pengembang konsol generasi keempat tetap eksis dengan produk terbarunya. Grafis *3D* mulai menjadi alasan untuk perang konsol ini. *Nintendo* dengan *Super Nintendo* dan *Sega* dengan *Sega Saturn*. Para pengembang *game* lebih menyukai membuat *game* untuk konsol *Nintendo* dan produk *Sony* yang cukup terkenal yaitu *Playstation*. (Ivan C. Sibero, 2009:14)

II.2.1.6. Generasi Keenam (1996-2006)

Generasi 128 bit. Teknologi semakin maju membuat beberapa konsol baru mulai muncul dan semuanya memiliki pasar sendiri. Namun para pabrikan konsol generasi sebelumnya sampai sekarang masih ikut dalam perang konsol ini. *PC* konsol masih menjadi lirikan para penggila *game*. Dengan kemajuan teknologi ini membuat era ini perkembangan *game PC* lebih cepat dibanding generasi sebelumnya. Kartu grafis yang mulai memiliki kecepatan tinggi sehingga membuat para *game* dimanjakan. Para pengembang *game PC* juga mulai bertumbuh. Pada generasi ini, beberapa produk mutakhir mulai bermunculan seperti *Sony Playstation 2* dari *Sony*, *Sega's Dreamcast*, *Xbox* dari *Microsoft*, dan *Nintendo* dengan produk *Game Cube*-nya. (Ivan C. Sibero, 2009:15)

II.2.1.7. Generasi Ketujuh (2004)

Pada masa ini beberapa pengembang konsol mulai fokus dengan pengembangan saja. Mulai tahun 2004 hanya beberapa perusahaan konsol yang masih bertahan dengan penembangannya seperti *Nintendo*, *Sony*, dan *Microsoft*.

Diluar itu, perusahaan konsol mulai membuat konsol yang lebih menarik dan dengan fitur baru dan *game-game* barunya yang menarik. Beberapa konsol baru yang ada dipasaran saat ini masih dibuat oleh perusahaan konsol generasi sebelumnya, seperti *Nintendo Wii* oleh *Nintendo*, *Xbox 360* dari *Mricosoft* dan *Playstation 3* dari *Sony*. Perkembangan konsol ini juga mempengaruhi tumbuhnya para pengembang *Software game*. (Ivan C. Sibero, 2009:17)

II.2.2. *Personal Computer (PC) Game*

Perkembangan *PC game* pada masa perang konsol masih sangat lambat karena *PC* lebih sering digunakan untuk peralatan kantor. Namun perkembangan *hardware* komputer membuat para *game developer* melirik *PC* sebagai konsol *game*. Perlu diingat perkembangan *game PC* cukup baik pada era 80-an. *Game Digger* menjadi salah satu *game* yang cukup laris. Ingin bermain dengan tampilan yang realistis dengan *genre* yang berbeda, ini mungkin jawaban bagi mereka yang mulai membuat *game PC*. (Ivan C. Sibero, 2009:18)

II.2.3. *Genre Game*

Genre game adalah klasifikasi *game* yang didasari interaksi pemainnya. Visualisasi juga menjadi ukuran klasifikasi *genre* ini. Namun untuk beberapa kasus pengembang *game* membuat kompilasi antar berbaai *genre* ini. Tentu saja variasi format *game* lebih banyak. (Ivan C. Sibero, 2009:18)

II.2.3.1. *Action*

Genre ini mungkin gaya permainan yang paling diminati para *player*. Dibutuhkan kecermatan reaksi waktu dan gerak. *Genre* ini memiliki banyak

rintangan di dalamnya. Jenis *game* ini seperti menembak, perkelahian dan banyak lagi. (Ivan C. Sibero, 2009:18)

II.2.3.2. *Ball And Paddle*

Jenis *game* ini merupakan jenis *game* yang pertama kali dibuat. Bola dan penangkisnya menjadi alat dalam permainan ini. *Game Ping Pong* adalah salah satu contoh *genre game* ini. Memang terlihat sederhana, namun dapat dikembangkan dengan visualisasi yang lebih menarik, misal objek bola diganti menjadi buah-buahan atau lainnya. Contoh *game genre* ini adalah *Arkanoid* dan *Block Buster*. (Ivan C. Sibero, 2009:18)

II.2.3.3. *Beat'em up, hack and slash*

Jenis *game* ini menekankan pada reaksi pemain. Berpetualang sambil bertempur merupakan aksi dari *genre* ini. Aksi pukul-tendangan, penggunaan pedang dan beberapa alat lain menjadi hiburan tersendiri buat pemain. Lawan pemain bisa berupa apa saja dan biasanya cukup banyak. Dilihat dari aksi yang cukup banyak ini dibutuhkan kontrol yang cukup banyak juga. *Double Dragon* merupakan *game* dari *genre Beat'em up, hack and slash* yang cukup terkenal pada tahun 1987. (Ivan C. Sibero, 2009:19)

II.2.3.4. *Fighting*

Jenis ini sekilas hampir mirip, namun perbedaan terlihat dari pertempurannya. Di sini pemain berhadapan satu lawan satu dengan musuh yang memiliki beragam keahlian. Karena karakter lawan digerakkan oleh komputer maka keahlian bertarung mereka menggunakan kontrol yang begitu banyak

sampai harus dihafal. *Street Fighter* adalah *game* yang cukup populer dari *genre* ini yang sampai sekarang masih banyak dimainkan. (Ivan C. Sibero, 2009:19)

II.2.3.5. Maze

Genre ini sudah tidak asing lagi bagi para *gamer*, Beberapa pengembang *game* menyatakan bahwa *genre* ini cukup bertahan dan menjadi dasar untuk pengembangan beberapa *game*. Diawali dengan *game Pacman* dan *Digger* (*game PC* pertama), *game* ini cukup mudah untuk dimainkan, dan cukup menarik. Reaksi gerakan pemain harus cepat agar terhindar dari musuh menjadi dasar dari *game* ini. (Ivan C. Sibero, 2009:20)

II.2.3.6. Pinball

Jenis *game* ini merupakan aplikasi dari permainan *pinball table*. Kontrol *game* ini cukup sederhana, berupa dua buah lengan pemukul bola. Bola yang memantul memberikan nilai yang berbeda pada tiap zona. Mungkin sedikit sulit mencapai nilai yang tinggi karena bola yang dipukul tidak dapat dikontrol gerakannya. (Ivan C. Sibero, 2009:20)

II.2.3.7. Game Tebak Gambar Dan Puzzle

Tebak gambar adalah permainan asah otak ringan, kumpulan gambar disusun sedemikian rupa sehingga bisa menimbulkan sebuah kosakata baru yang diadaptasi dari istilah sehari-hari, ungkapan unik dan lucu, ataupun berupa isu dan peristiwa yang sedang terjadi. *Puzzle* yang merupakan permainan asah otak yang menantang ketrampilan pemainnya, sepertinya tidak pernah kehilangan

popularitasnya dan tidak pernah termakan usia. *Puzzle* merupakan salah satu jenis *game* yang cukup memeras otak untuk menyelesaikannya.(Ashari,2014:1)

II.3. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi. Antarmuka pengguna android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan didunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek pada layar. Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalisasi perangkat, umumnya ditulis dalam versi bahasa pemrograman Java yang telah di-*custom*.(wahana komputer,2014:1-2)

Android adalah istilah dalam bahasa Inggris yang berarti “Robot yang menyerupai manusia”. Logo “Android” sendiri, dicerminkan seperti sebuah robot berwarna hijau, yang mengacu kepada arti kata Android.



Gambar II.1. Logo Android

(Sumber:Beginning Android Programming with ADT Bundle: 2014: 2-4)

Android adalah sistem operasi yang bersifat *open source* (sumber terbuka). Disebut *open source* karena *source code* (kode sumber) dari sistem operasi Android dapat dilihat, di-*download*, dan dimodifikasi secara bebas. Paradigma *open source* ini memudahkan pengembangan teknologi Android, karena semua pihak yang tertarik dapat memberikan kontribusi, baik pada pengembangan sistem operasi maupun aplikasi. (Alfa dan eva, 2014: 2-4).

II.3.1. Versi dan Jenis-Jenis Android

Pengembangan android dimulai dengan berdirinya Android, Inc. pada Oktober 2003 dengan tujuan membuat *Mobile Device* yang lebih *Smart* untuk menyaingi Symbian dan Windows Mobile yang populer saat itu (iPhone dan Blackberry). Pada tahun 2005, Android Inc. diakuisisi oleh Google. Pengembangan terus dilanjutkan sampai Android versi beta diluncurkan pada tanggal 5 November 2007, bersamaan dengan berdirinya OHA (Open Handset Alliance). Sampai saat ini tanggal 5 November diperingati sebagai hari jadi Android. Seminggu setelahnya, pada tanggal 12 November 2007 Android SDK (Software Development Kit) diluncurkan, sehingga pengguna dapat membuat dan mengembangkan aplikasi Android mereka sendiri.

Android telah dikembangkan dan diupdate beberapa kali sejak rilis pertamanya. Tabel dibawah ini memperlihatkan versi Android semenjak pertama kali dirilis.

Tabel II.1.Versi Android

Versi	Nama	Tanggal Rilis	Catatan
1.1	-	9 Februari 2009	-
1.5	Cupcake	30 April 2009	Mulai pakai kode nama
1.6	Donut	15 September 2009	-
2.0 / 2.1	Éclair	26 Oktober 2009	-
2.2	Froyo	20 Mei 2010	-
2.3	Gingerbread	6 Desember 2010	Masih banyak digunakan di smartphone jenis lama
3.0	Honeycomb	22 Februari 2011(3.0) 10 Mei 2011 (3.1) 15 Juli 2011 (3.2)	Hanya untuk tablet
4.0	Ice Cream Sandwich	19 Oktober 2011	Smartphone dan Tablet
4.1- 4.3	Jelly Bean	9 Juli 2012 (4.1) 13 November 2012(4.2) 24 Juli 2013 (4.3)	Update untuk memperbaiki dan menambah fitur-fitur ICS
4.4	Kit Kat	-	Diperkenalkan November 2013

(Sumber : Beginning Android Programming with ADT Bundle: 2014: 7)

II.3.2. Tool Pengembangan Android

1. Eclipse

Sejak berkurangnya standarisasi pengujian aplikasi pada android, membangun aplikasi pada *operating system* tersebut menjadi lebih mudah dan fleksibel dibandingkan sebelumnya. Sehingga semakin banyak programmer baru yang lahir dengan beragam kreatifitas dalam mengembangkan aplikasi. Dalam mengembangkan sebuah aplikasi sendiri, dibutuhkan software pendukung, Begitu pula dalam mengembangkan aplikasi pada android, *software* yang diperlukan seperti Eclipse sebagai IDE(*Integrated Development Environment*)atau program komputer dengan beberapa fasilitas yang diperlukan dalam mengembangkan perangkat lunak.(Alfa dan Eva, 2014;12)

2. **Android SDK (Software Development Kit)**

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. Beberapa fitur-fitur *Android* yang paling penting adalah mesin *Virtual Dalvik* yang dioptimalkan untuk perangkat *mobile*, *integrated browser* berdasarkan *engine open source WebKit*, Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi *opengl ES 1.0* (Opsional akselerasi perangkat keras), kemudian *SQLite* untuk penyimpanan data (*database*). Fitur-fitur *android* lainnya termasuk media yang mendukung *audio*, *video*, dan gambar, juga ada fitur *bluetooth*, EDGE, 3G dan WiFi, dengan fitur kamera, GPS, dan kompas. Selanjutnya fitur yang juga turut disediakan adalah lingkungan *Development* yang lengkap dan kaya termasuk perangkat emulator, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE *Eclipse*. (Alicia Sinsuw ; 2013 : 2)

3. **AVD (Android Virtual Device)**

Android Virtual Device merupakan emulator untuk menjalankan aplikasi *android*, yang tampilannya dapat dilihat pada gambar 1. Setiap AVD terdiri dari sebuah profil perangkat keras yang dapat mengatur pilihan untuk menentukan fitur *hardware emulator*. Misalnya, menentukan apakah menggunakan perangkat kamera, apakah menggunakan *keyboard* QWERTY fisik atau tidak, berapa banyak memori internal, dan lain-lain. AVD juga memiliki sebuah pemetaan versi *Android*, maksudnya kita menentukan versi dari *platform Android* akan berjalan pada emulator. Pilihan lain dari AVD, misalnya menentukan *skin* yang kita ingin

gunakan pada emulator, yang memungkinkan untuk menentukan dimensi layar, tampilan, dan sebagainya. Kita juga dapat menentukan *SD Card virtual* untuk digunakan dengan di emulator. (Alicia Sinsuw ; 2013 : 2)



Gambar II.2. Tampilan AVD
(Sumber : Alicia Sinsuw ; 2013 : 3)

II.4. Adobe Flas CS6

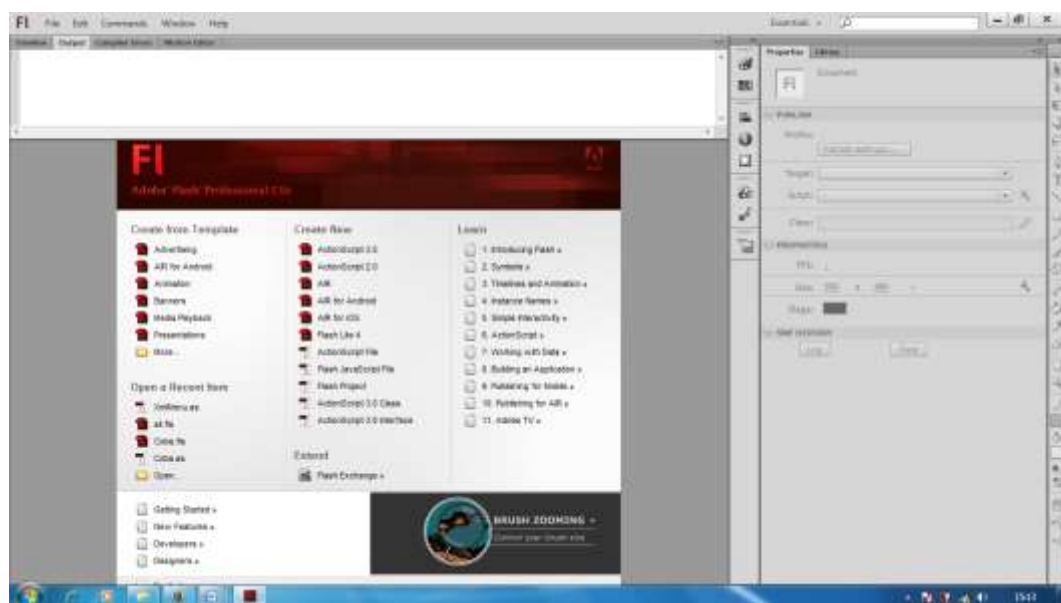
Adobe flas CS6 (dahulu bernama *Macromedia Flash*) adalah salah satu perangkat lunak komputer yang merupakan produk unggulan *Adobe System*. *Adobe Flash* digunakan untuk membuat gambar *vector* maupun animasi gambar tersebut. Bekas yang dihasilkan dari perangkat lunak ini mempunyai file *extension* .swf dan dapat diputar di penjelajah web yang telah dipasang *Adobe Flash Player*. Flash menggunakan bahasa pemrograman bernama *Action Script* yang muncul pertama kalinya adalah *Adobe Flash CS5*. (Kristo Radion, S..ST, 2012:3)

II.4.1. Area Kerja Adobe Flash CS6

Langkah untuk menjalankan program *Adobe Flash Pro CS6*, tekan tombol **Start ► All Programs ► Adobe ► Adobe Flash CS6** sehingga tampil **Welcome Screen** seperti tampak pada gambar berikut ini :



Gambar II.3. Tampilan Awal Adobe Flash CS6
(Sumber : Adobe Flash Professional CS6)



Gambar II.4. Tampilan Area Kerja Adobe Flash CS6
(Sumber : Adobe Flash Professional CS6)

Di dalam *Welcome Screen* menampilkan empat pilihan perintah untuk Adobe Flash CS6 yaitu sebagai berikut :

1. **Create from Template**, berguna untuk membuka lembar kerja dengan template yang tersedia dalam program *Adobe Flash Pro CS6*.
2. **Open a Recent Item**, berguna untuk membuka kembali *file* yang pernah anda simpan atau pernah buka sebelumnya.
3. **Create New**, berguna untuk membuka lembar kerja baru dengan beberapa pilihan script yang tersedia.
4. **Learn**, bertujuan untuk membuka jendela *Help* yang berfungsi untuk mempelajari suatu jendela perintah.

II.4.2. Toolbox

Toolbox adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain animasi mulai dari tombol seleksi, pen, pensil, *Text*, *3D Rotation*, dan lain-lain.



Gambar II.5. Tampilan Menu Bar
(Sumber : Adobe Flash Professional CS6)

Tabel II.2. Fungsi Tombol Toolbox

Nama Tombol	Fungsi
Selection Tool (V)	Untuk menyeleksi objek.
Subselection Tool (A)	Untuk menyeleksi bagian objek untuk proses editing.
Free Transform Tool (Q)	Untuk mengubah bentuk objek secara bebas.
GradientTransform Tool(F)	Untuk mengubah transformasi warna gradasi sebuah objek.
3D Rotation Tool (W)	Untuk melakukan transformasi bentuk dan posisi 3D pada objek berdasarkan sumbu X, Y dan Z.
3D Translation Tool (G)	Untuk melakukan transformasi bentuk dan posisi 3D pada simbol movie clip dengan acuan tiga sumbu X, Y dan Z.
Lasso Tool (L)	Untuk menyeleksi objek dengan pola seleksi bebas.
Pen Tool (P)	Untuk menggambar objek
Add Anchor Point Tool (=)	Untuk menambah titik anchor pada sebuah path.
Delete Anchor Point Tool (-)	Untuk menghapus titik Anchor.
Convert Anchor Point Tool (C)	Untuk mengubah sudut lancip dari sebuah path menjadi sudut lengkung.
Text Tool (T)	Untuk mengetik text dan paragraf.
Line Tool (N)	Untuk menggambar objek garis lurus.
Rectangle Tool (R)	Untuk menggambar objek kotak.
Oval Tool (O)	Untuk menggambar objek oval atau lingkaran.
RectanglePrimitive Tool (R)	Untuk menggambar objek kotak dengan sudut dapat di lengkungkan.
Oval Primitive Tool (O)	Untuk menggambar objek lingkaran dengan berbagai variasi.
PolyStar Tool	Untuk menggambar objek poligon dan bintang.
Pencil Tool (Y)	Untuk menggambar dengan bentuk goresan pensil.
Brush Tool (B)	Untuk menggambar dengan bentuk polesan kuas.
Spray Brush Tool (B)	Untuk menggambar dengan spary, yaitu menyemprotkan warna atau simbol.

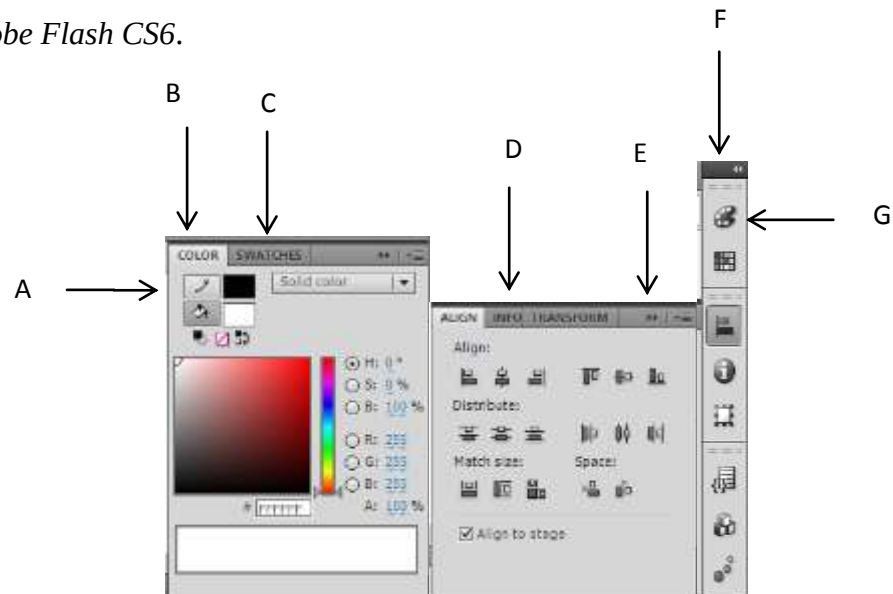
Tabel II.2. Fungsi Tombol Toolbox (Lanjutan)

Nama Tombol	Fungsi
Deco Tool (U)	Untuk menggambar corak dekorasi dengan menggunakan simbol graphic.
Bone Tool (X)	Membuat animasi pertulangan dengan menggunakan titik sendi pada objek.
Bind Tool (Z)	Melakukan pengeditan dan modifikasi titik sendi dari piranti Bone Tool.
Ink Bottle Tool (S)	Untuk memberi warna dan bentuk garis outline pada sebuah objek.
Paint Bucket Tool (K)	Untuk memberi warna bidang objek.
Eyedropper Tool (I)	Untuk mengambil sample warna dari sebuah objek.
Eraser Tool (E)	Untuk menghapus bidang objek.
Hand Tool (H)	Untuk menggeser area lembar kerja atau stage.
Zoom Tool (M,Z)	Untuk memperbesar atau memperkecil tampilan lembar kerja atau stage.
Stroke Color	Untuk menentukan warna garis.
Fill Color	Untuk menentukan warna bidang objek.
Black and White	Untuk mengubah warna garis dan bidang menjadi hitam dan putih.
Swap Colors	Untuk membalik warna antara warna garis dan warna bidang objek.
No Color	Untuk menghapus warna garis atau warna bidang objek.
Snap to Objects	Untuk mengaktifkan atau mematikan fungsi Snap to Objects.

II.4.3. Tampil Panel

Panels, berisi kontrol fungsi yang di pakai dalam flash, yang berfungsi untuk mengganti dan memodifikasi berbagai atribut dari objek atau animasi secara cepat dan mudah. *Panels* biasanya terletak di bagian kanan area *Flash*. Untuk menampilkan panel tersebut, klik menu *Window>(nama panel)*.

Program *Adobe Flash Pro CS6* menampilkan lembar kerja yang sangat menarik serta dapat di ubah menurut selera Anda. Dengan perubahan tersebut, Anda lebih mudah mengatur dan menyusun objek yang akan dianimasikan. Perhatikan gambar berikut serta keterangannya untuk mengatur tampilan lembar kerja *Adobe Flash CS6*.



Gambar II.6. Tampilan Panel
(Sumber : Adobe Flash Professional CS6)

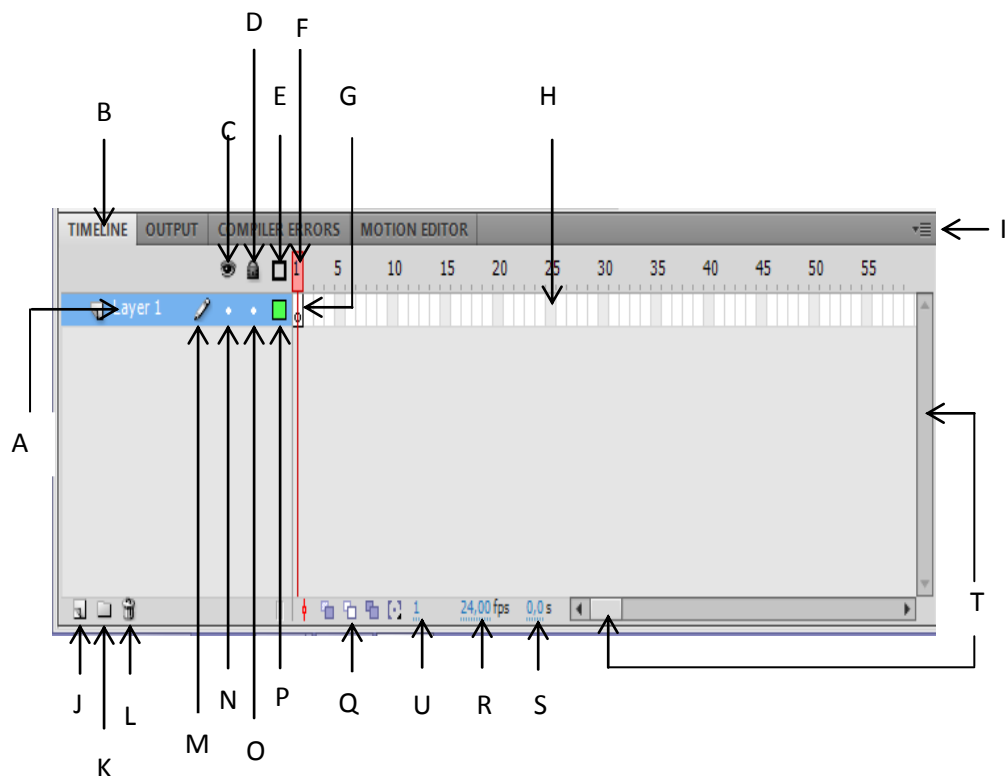
Tabel II.3. Keterangan Tampilan Panel

Abjad	Keterangan
A	Klik tahan untuk mengatur lembar panel.
B	Klik untuk menutup uraian panel aktif.
C	Klik untuk menutup panel aktif.
D	Klik dan geser keluar untuk melepas panel dari grupnya.
E	Klik untuk menutup uraian panel kedalam dock panel.
F	Klik untuk menguraikan semua dock panel.
G	Klik ikon panel untuk menguraikan panel dock terpilih.

(Sumber : Adobe Flash Professional CS6)

II.4.4. Timeline

Timeline berguna untuk menentukan durasi animasi, jumlah layer, frame, menempatkan script dan beberapa keperluan animasi lainnya. Semua bentuk animasi yang Anda buat akan diatur dan ditempatkan pada layer dalam timeline.



Gambar II.7. Tampilann Timeline CS6

(Sumber : Adobe Flash Propesional CS6)

Tabel II.4. Keterangan Tampilan Timeline

Abjad	Nama	Keterangan
A	Layer	Layar kerja yang menampung objek yang akan dianimasika di dalam Timeline.
B	Timeline	Tabulasi dari lembar kerja atau Stage yang sedang dikerjakan.
C	Show/Hide All Layers	Untuk menyembunyikan atau menampilkan semua isi layer.

Tabel II.4. Keterangan Tampilan Timeline (Lanjutan)

Abjad	Nama	Keterangan
D	Lock/Unlock All Layers	Untuk mengunci atau melepas kunci objek dari semua layer.
E	Show All Layer as outlines	Untuk menampilkan objek pada semua layer dalam bentuk outline.
F	Playhead	Jarum untuk membaca Frame pada saat animasi dijalankan.
G	Blank Keyframe	Sebuah simbol lingkaran kosong yang menampung suatu objek.
H	Frame	Suatu bagian dari layer yang digunakan untuk mengatur pembuatan animasi.
I	Menu	Untuk mengatur tampilan Frame.
J	New Layer	Untuk menambahkan layer baru.
K	New Folder	Untuk menambahkan folder baru.
L	Delete	Untuk menghapus layer
M	Simbol Pensil	Menunjukkan bahwa layer dalam kondisi terpilih atau aktif.
N	Titik Show or Hide	Klik untuk menampilkan atau menyembunyikan layer aktif.
O	Titik Kunci	Klik untuk mengunci atau melepaskan kunci layer yang aktif.
P	Kotak Outline	Klik untuk menampilkan objek dalam layer aktif menjadi bentuk outline.
Q	Onion skinning button	Untuk mengatur tampilan animasi didalam stage.
R	Frame Rate	Untuk mengatur kecepatan gerak animasi dalam tiap detiknya.
S	Elapsed Time	Menunjukkan durasi atau lamanya animasi.
T	Scrollbar	Menggulung jendela Timeline secara vertikal dan horisontal.
U	Current Frame	Menunjukkan posisi Frame aktif.

(Sumber : Adobe Flash Profesional CS6)

II.4.5. Panel Propertis

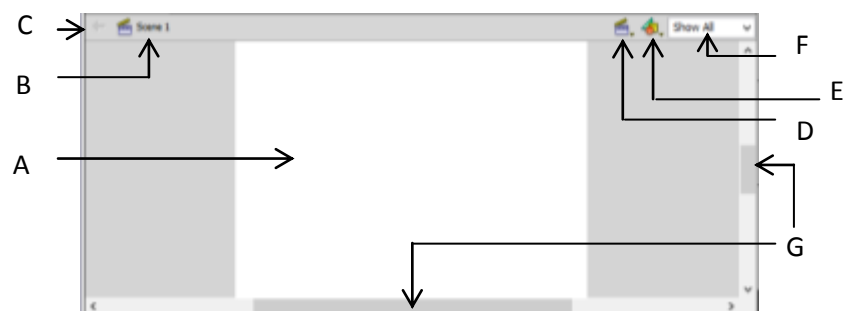
Panel Propertis berguna untuk menampilkan parameter dari sebuah tombol yang terpilih sehingga Anda dapat memodifikasi dan memaksimalkan fungsi dari tombol tersebut. Panel Propertis menampilkan parameter sesuai dengan tombol yang terpilih.



Gambar II.8. Tampilan Properties
(Sumber : Adobe Flash Propesional CS6)

II.4.6. Stage

Stage adalah lembar kerja yang di gunakan untuk membuat atau mendesain objek yang akan dianimasikan. Objek yang dibuat dalam lembar kerja dapat berupa objek *Vektor*, *Movie clip*, *Text*, *Button*, dan lain-lain.



Gambar II.9. Tampilan Stage
(Sumber : Adobe Flash Propesional CS6)

Tabel II.5. Keterangan Tampilan Stage

Abjad	Keterangan
A	Stage , lembar kerja untuk menyusun objek yang akan dianimasikan.
B	Scene , menunjukkan nama scene yang aktif.
C	Panah yang digunakan untuk berpindah dari lembar kerja simbol ke lembar kerja utama.
D	Edit Scene , untuk memilih nama scene yang akan diedit.
E	Edit Symbols , untuk memilih nama simbol yang akan diedit.
F	Zoom , untuk mengatur besarnya tampilan stage atau lembar kerja.
G	Scrolber , untuk menggulung lembar kerja secara horisontal dan vertikal.

(Sumber : Adobe Flash Profesional CS6)

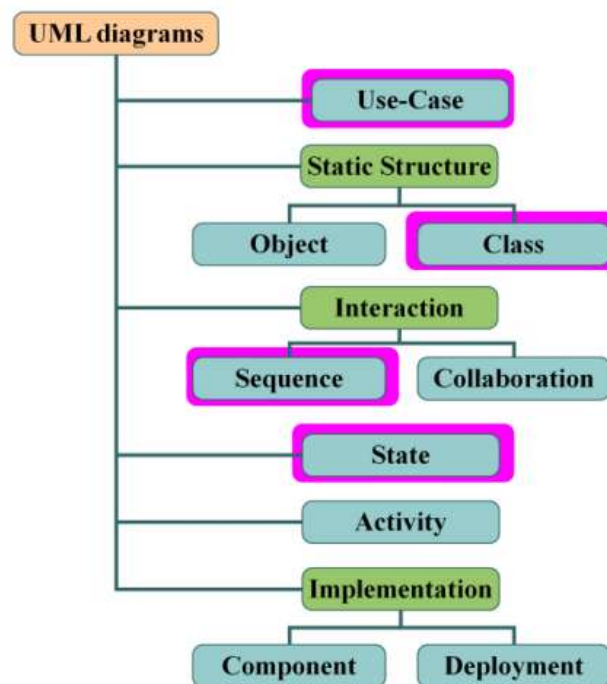
II.5. UML(Unified Modelling Language)

Unified Modelling Language merupakan alat perancangan sistem yang berorientasi pada objek. Secara filosofi kemunculan UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* (OO), karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik maka OO memiliki proses standard dan bersifat independen.

UML diagram memiliki tujuan utama untuk membantu tim pengembangan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu *Grady Booch*, OOD (*Object-Oriented Design*), *Jim Rumbaugh*, OMT (*Object Modelling Technique*), dan *Ivar Jacobson* OOSE (*Object-Oriented Software Engineering*).

UML mempunyai tiga kategori utama yaitu *struktur diagram*, *behaviour diagram* dan *interaction diagram*. Dimana masing-masing kategori tersebut memiliki diagram yang menjelaskan arsitektur sistem dan saling terintegrasi. (Haviluddin, 2011)

Menurut Haviluddin (2011) Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi olehobyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. Berikut gambar dari diagram UML.



Gambar II.10. Diagram UML
(Sumber : Haviluddin , 2011 ; 2)

Komponen-komponen UML

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

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu :

A. Struktur Diagram

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem.

Struktur diagram dalam UML terdiri atas :

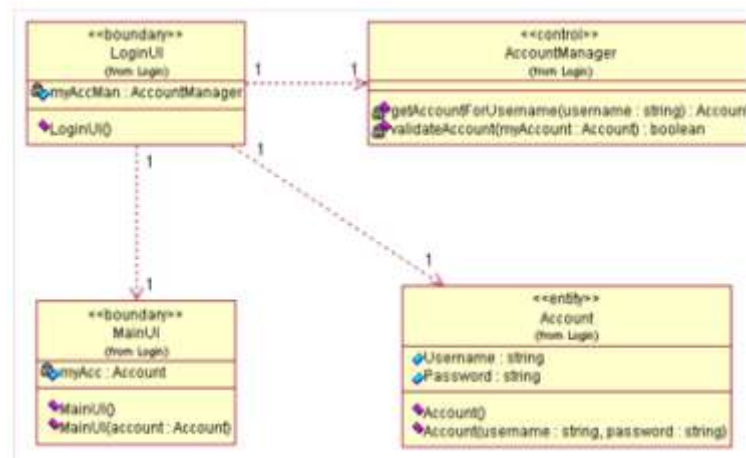
1. *Class Diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas.

Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Class memiliki tiga area pokok :

- a. Nama (dan stereotype)
- b. Atribut
- c. Metoda



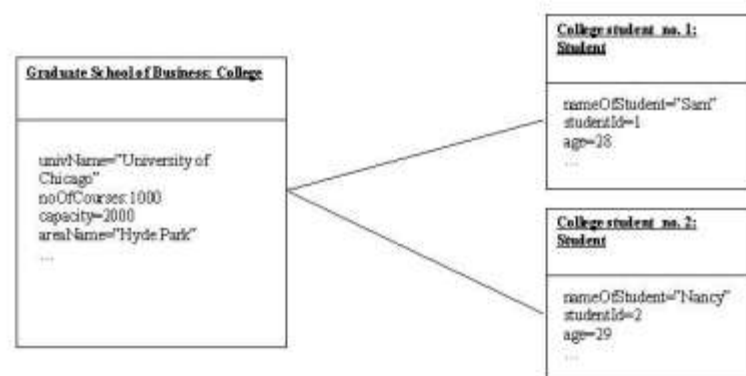
Gambar II.11. Contoh Notasi Class Diagram

(Sumber : Haviluddin , 2011 ; 3)

2. Object diagram

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

Berikut notasi *object diagram* :

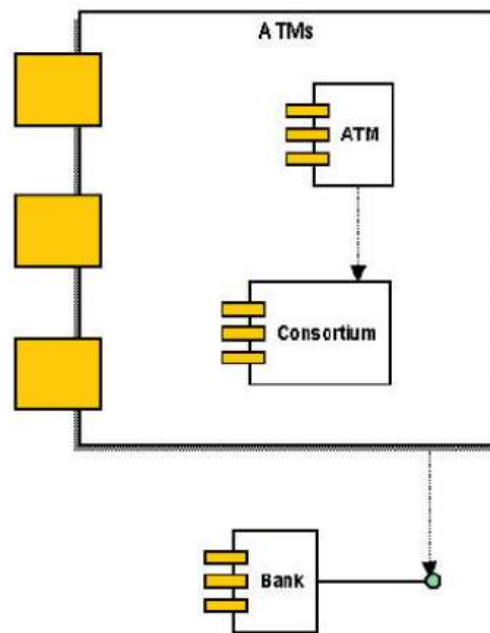


Gambar II.12. Contoh Notasi Object Diagram

(Sumber : Haviluddin , 2011 ; 3)

3. *Component diagram*

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

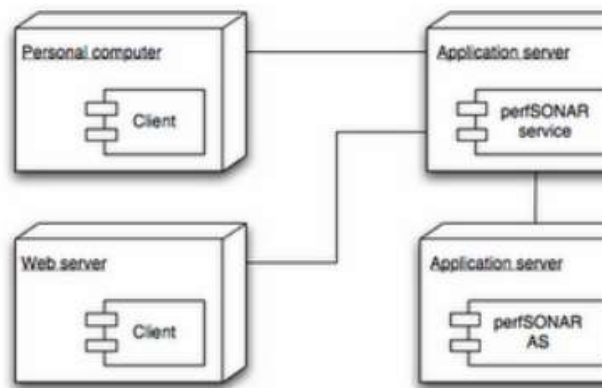


Gambar II.13. Contoh Notasi *Object Diagram*

(Sumber : Haviluddin , 2011 ; 3)

4. *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. (Haviluddin , 2011 ; 3).



Gambar II.14. Contoh Notasi *Deployment Diagram*
(Sumber : Haviluddin , 2011 ; 4)

5. *Composite structure diagram*

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

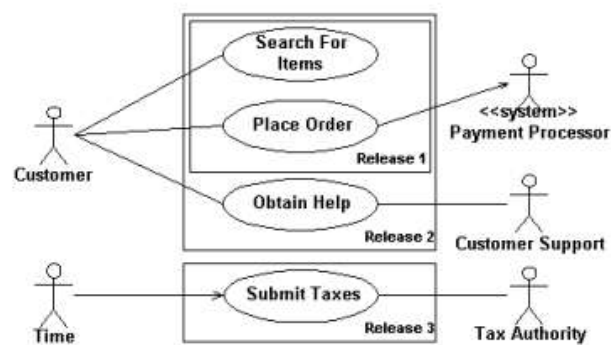
6. *Package diagram*

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

B. Behavior Diagram

1. Usecase 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*.



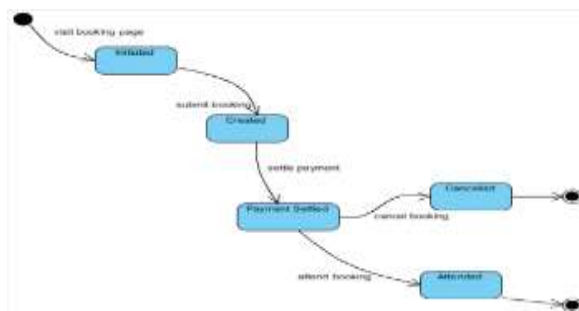
Gambar II.15. Contoh Notasi Usecase Diagram
(Sumber : Haviluddin , 2011 ; 4)

2. Activity diagram

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

3. State Machine diagram

Menggambarkan *state*, *transisi state* dan *event*.



Gambar II.16. Contoh Notasi State Machine Diagram
(Sumber : Haviluddin , 2011 ; 4)

B. Interaction diagram

1. *Communication diagram*

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

2. *Interaction Overview diagram*

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

3. *Sequence diagram*

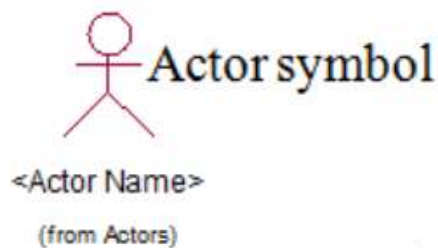
Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*

4. *Timing diagram*

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

Untuk menggambarkan analisa dan desain diagram, UML memiliki seperangkat notasi yang akan digunakan ke dalam tiga kategori diatas yaitu struktur diagram, behaviour diagram dan interaction diagram. Berikut beberapa notasi dalam UML diantaranya :

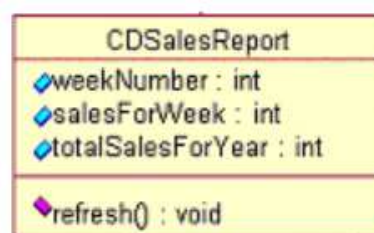
1. *Actor*, menentukan peran yang dimainkan oleh user atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas actor adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas.



Gambar II.17. Notasi Actor

(Sumber : Haviluddin , 2011 ; 6)

2. *Class diagram* notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek.



Gambar II.18. Notasi Class

(Sumber : Haviluddin , 2011 ; 6)

3. *Use Case* dan *use case specification*, *Use case* adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan 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 disebut skenario.

4. Realization, menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.
5. Interaction, digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.
6. Dependency, merupakan relasi yang menunjukan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Terdapat 2 *stereotype* dari *dependency*, yaitu *include* dan *extend*. *Include* menunjukkan bahwa suatu bagian dari elemen (yang ada digaris tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada di garis dengan panah). *Extend* menunjukkan bahwa suatu bagian dari elemen di garis tanpa panah bisa disisipkan ke dalam elemen yang ada di garis dengan panah. (Haviluddin, 2011; 6-7).