BAB II
LANDASAN TEORI

II.1 Belajar

Prestasi belajar tidak dapt dipisahkan dari perbuatan belajar, karena belajar merupakan suatu proses, sedangkan prestasi belajar adalah hasil dari proses pembeljaran tersebut. Bagi seorang anak belajar merupakan suatu kewajiban. Berhasil atau tidaknya seorang anak dalam pendidikan tergantung pada proses yang dialami oleh anak tersebut.

Winkel berpendapat bahwa belajar pada manusia dapat dirumuskan sebagai suatu aktivitas mental atau psikis yang berlangsung dalam interaksi aktif dengan lingkungan, yang menghasilkan perubahan – perubahan dalam pengetahuan dan nilai sikap. Perubahan itu bersifat relatif konstan dan berbekas [10].

Belajar tidak hanya dapat dilakukan di sekolah saja, namun dapat dilakukan dimana-mana, seperti di rumah ataupun di lingkungan masyarakat. Belajar adalah suatu usaha atau kegiatan yang bertujuan mengadakan perubahan di dalam diri seseorang, mencakup perubahan tingkah laku, sikap, kebiasaan, ilmu pengetahuan, keterampilan dan sebagainya [5].

Aktifitas belajar bagi setiap individu tidak selamanya dapat berlangsung secara wajar. Perbedaan individual inilah yang menyebabkan perbedaan tingkah laku belajar dikalangan anak didik. Dalam keadaan dimana anak didik / siswa tidak dapat belajar sebagaimana mestinya.

Disetiap sekolah dalam berbagai jenis dan tingkatan pasti memiliki anak didik yang berkesulitan belajar. Setiap kali kesulitan belajar anak didik yang satu. dapat diatasi, tetapi pada waktu yang lain muncul kesulitan belajar anak didik yang lain.

II.2 Hakikat Pembelajaran Berbasis Komputer

Komputer memberikan kontribusi besar dalam pengembangan dalam banyak hal termasuk pendidikan. Komputer pada pembelajaran berfungsi media pembelajaran dengan program – program aplikatif yang sesuai dengan tujuan dan metode yang digunakan oleh sekolah atau oleh pembelajaran yang menggunakann media komputer. Penggunaan *game* dalam pembelajaran anak usia dini merupakan suatu cara yang tepat, sebab anak pada usia dini memang masih merupakan masa bermain, sehingga melalui metode bermain anak dapat melakukan pembelajaran dengan tanpa sengaja dan tidak merasa terpaksa.

Inti dari pendidikan *Information Technology* ini ada dua yaitu mengenalkan IT sejak dini dan membantu proses belajar anak dengan cara yang lebih meyenangkan. Dengan pengertian lain dalam hal ini komputer dikenalkan sebagai obyek yang dipelajari dan sekaligus juga komputer dipakai sebagai media pembelajaran ilmu – ilmu lainnya.

Bentuk – bentuk pengenalan IT ini berupa pengenalan perangkat keras komputer yang bisa dilihat dan dipegang langsung oleh anak misal CPU. Monitor, *mouse, keyboard* dan *printer*. Pengertian perangkat keras ini dilengkapi dengan penjelasan masing – masing fungsi alat dengan cara langsung mempraktekan penggunaanya. Kemahiran mengunakan *keyboard* dan *mouse* lebih mudah diajarkan pada anak – anak dibandingkan dengan orang yang sudah berumur.

Software – software edutaiment lain yang mengasah intelektualitas dan psikomotorik anak dikenalkan pada usia pra sekolah (4 – 8 tahun). Penyusunan *puzzle*, *game* strategi, latih memori, penyusunan tangram, mencari benda tersembunyi dan mencari perbedaan adalah contoh *software* yang cocok digunakan di usia dini. *Software* pengenalan huruf dan angka dalam dua atau tiga bahasa dan permainan logika dapat membantu perkembangan otak anak. Teknologi khusunya IT diharapkan dapat menjembatani kesimbangan otak kiri dan kanan anak – anak.

II.3 Hakikat Anak Usia Dini

Pendidikan bagi anak usia dini atau anak usia 0 – 8 tahun sudah lama menjadi perhatian para orang tua, para ahli pendidikan, masyarakat dan pemerintah. Plato mengatakan bahwa waktu paling tepat untuk mendidik anak adlah sebelum usia 6 tahun. Hal ini menggambarkan bahwa anak seperti kain putih yang siap untuk dipola atau dibentuk baik sosial, psikologis, nalar maupun fisiknya. Adapun rentang usia 4 sampai 6 tahun merupakan tahapan yang disebut sebagai usia pra.

Pendidikan prasekolah untuk rentang usia tersebut pada dasarnya adalah untuk membantu pertumbuhan dan perkembangan fisik dan psikologis anak di luar lingkungan keluarga sebelum memasuki pendidikan dasar formal. Sedangkan ttujuan dari pendidikan prasekolah adalah untuk membantu keterampilan fisik dan motorik, sosial, moral, dan daya cipta yang diperlukan oleh anak – anak untuk menyesuaikan diri dengan lingkungannya, serta untuk pertumbuhan dan perkembangan tahap selanjutnya.

Berdasarkan penjelasan diatas anak usia tersebut mempunyai karakteristik dua ranah sebagai berikut :

a. Fisik / Motorik

[Hurtlock] berpendapat pertumbuhan fisik anak – anak pada usia masa ini berlangsung lambat bila dibandingkan dengan pertumbuhan masa bayi. Pada masa ini pertumbuhan relatif seimbang antara berat dan tinggi badan. Otot – otot badan cenderung lebih baik pada usia ini. Bahkan nafsu makan anak – anak ini sudah tidak sebesar pada masa bayi karena tingkat pertumbuhan tidak sebesar masa bayi lagi. Sesuai dengan bertambah matangnya pertumbuhan otak yang mengatur sistem syaraf yang memungkinkan anak menjadi lincah dan aktif bergerak. Dengan meningkatnya usia nampak adanya perkembangan dari gerakan motorik kasar (menggunakan otot semata seperti berdiri tegak, berlari, loncat dan sebagainya) ke motorik halus (menggunakan otak, mata, dan menghasilkannya ke dalam kegiatan) yang memerlukan kecermatan dan kontrol yang lebih baik. Kegiatan – kegiatan ini juga selain memberikan kesempatan mengembangkan keterampilan motorik kasar dan halus, juga mengembangkan keterampilan dan koordinasi mata dan tangan.

b. Intelektualitas

Menurut Izztaydunia kognitif anak usia ini adalah kreatif, bebas, dan penuh imajinasi. Perkembangan kognitif anak pada usia ini berfokus pada tahap pemikiran praoperasional. Hal ini ditunjukan oleh pemikiran bahwa pemikiran anak masih didominasi oleh hal – hal yang berkaitan dengan aktivitas fisik dan persepsinya sendiri. Dikatakan masa pemikiran

praoperasional karena menunjukan kemampuan untuk merekonstruksi pada tingkat apa yang telah dilakukan di dalam perilaku.

II.4 Jenis Pembelajaran Berbasis Komputer

Menurut *strickland Patterson,* ada lima tipe pembelajaran berbasis komputer (*Computer Aided Instruction*) yaitu :

1. *Drill & Practice,* menyajikan materi pembelajaran untuk dipelajari secara berulang.
2. *Tutorial,* menyajikan materi yang telah diajarkan atau materi yang baru di pelajari.
3. *Simulation,* memberi kesempatan untuk menguji kemampuan pada aplikasi nyata dengan menciptakan ituasi yang mengikutsertakan siswa – siswa untuk bertindak pada situasi tersebut.
4. *Problem Solving,*menyajikan masalah – masalah untuk siswa menyelesaikan berdasarkan kemampuan yang mereka peroleh.
5. *Educational Games,* merupakan paket *software* yang menciptakan kemampuan pada lingkunga permainan yang diberikan sebagai alat bantu untuk memotivasi atau membantu siswa untuk melalui prosedur permainan secara teliti untuk mengembangkan kemampuannya. Berdasarkan kriteria – kriteria ini, maka aplikasi yang dirancang penulis dapat ditentukan sebagai aplikasi educational *game* [4].

II.4.1 Kelebihan Pembelajaran Berbasisi Komputer

Menurut Wiharjo, kelebihan komputer sebagai media pembelajaran adalah:

1. Meningkatkan perhatian dan konsentrasi siswa.
2. Meningkatkan motivasi siswa.
3. Menyesuaikan materi dengan kemampuan siswa.
4. Mereduksi penggunaan waktu penyampaian materi.
5. Dapat mengakomodasi banyak siswa dan menjalankan fungsinya dengan sedikit kesalahan.
6. Dapat menggunakan fasilitas penyimpanan untuk mengetahui kemajuan belajar siswa.
7. Bersifat tanggap dan bersahabat sehingga siswa belajar tanpa tekanan psikologis.
8. Materi dapat di desain lebih menarik.
9. Tingakat kemampuan dan kecepatan belajar dapat dikontrol oleh siswa sehingga siswa dapat belajar dan berprestasi sesuai dengan kemampuannya.
10. Dapat mendorong guru untuk meningkatkan pengetahuan dan kemampuan mengenai komputer.

II.5 *Game* Edukasi

Penerapan *education game*bermula dari perkembangan *video game* yang sangat pesat dan menjadikannya sebagai media efektif yang interaktif dan banyak dikembangkan di perindustrian. Melihat kepopuleran *game* tersebut, para pendidik berfikir bahwa mereka mempunyai kesempatan yang baik untuk menggunakan komponen rancangan *game* dan menerapkannya pada kurikulum dengan penggunaan industri berbasis *game*. *Game* harus memiliki desain antarmuka yang interaktif dan mengandung unsur menyenangkan [2].

II.5.1 Pengertian *Game*

Education game merupakan salah satu bentuk pembelajaran berbasis komputer. Pembelajaran berbasis komputer telah banyak diterapkan di sekolah – sekolah. Hal itu disebabkan karena perkembangan yang cukup pesat di bidang teknologi informasi. Sistem pembelajaran berbasis komputer dirancang berbasis multimedia yang menggambungkan unsur – unsur visual, *audio*, dan *video* sehingga menjadikannya sangat interaktif.

Game merupakan kata dalam bahasa inggris yang berarti permainan. Permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan *refreshing*. Suatu cara belajar yang digunakan dalam menganalisa interaksi antara sejumlah pemain maupun perorangan yang menunjukkan strategi-strategi yang rasional. Teori permainan pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944. Teori itu dikemukakan oleh John von Neumann and Oskar Morgenstern yang berisi:

"*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.* (*Theory of Games and Economic Behavior* (3d ed. 1953)) [6].

II.5.2 Pengertian *Game* Menurut Beberapa Ahli

Ada beberapa ahli yang mengemukakan pengertian nya tentang *game*. Berikut ini beberapa ahli tersebut :

1. Menurut Agustinus Nilwan dalam bukunya Pemrograman Animasi dan *Game* Profesional, *game* merupakan permainan komputer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami pengunaan animasi haruslah memahami pembuatan *game*. Atau jika ingin membuat *game*, maka haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan [7].
2. Menurut Clark C. Abt, *game* adalah kegiatan yang melibatkan keputusan pemain, berupaya mencapai tujuan dengan "dibatasi oleh konteks tertentu" (misal, dibatasi oleh peraturan) [1].
3. Menurut Bernard Suits, *game* adalah "upaya sukarela untuk mengatasi rintangan yang tidak perlu" [8].
4. Menurut Greg Costikyan, *game* adalah "sebentuk karya seni di mana peserta, yang disebut pemain, membuat keputusan untuk mengelola sumberdaya yang dimilikinya melalui benda di dalam *game* demi mencapai tujuan" [3].

II.6 Sejarah *Game*

Pada tahun 1952, seorang mahasiswa Universitas Cambridge bernama A.S. Gouglas membuat permainan OXO (tic-tac-toe) dalam versi grafik. Permainan ini ia kembangkan ketika hendak mendemonstrasikan tesisnya tentang interaksi antara manusia dan komputer. Pada tahun 1958 William Higin Botham mendesain sebuah *game* dengan judul *Tennis For Two* yang dimainkan dalam

oscilloscope, dan kemudian ada pula Steve Russel pada tahun 1961 dengan *game* berjudul *Spacewar* yang dibuat dalam komputer mainframe DEC PDP-1 saat mereka menjalani studi di MIT.

Sejarah video *game* tidak hanya mengenai orang-orang yang berperan di dalamnya, tetapi juga mengenai berbagai perusahaan *game* yang mempunyai kasus ironis. Atari adalah perusahaan Amerika dengan nama Jepang, dan perusahaan Jepang bernama SEGA didirikan oleh orang Amerika. Magnavox yang memulai peran perusahaan ini telah berumur 1 abad, dan Nintendo sebagai perusahaan yang mempopulerkan kembali *video game* juga sama tuanya, serta tidak ada yang pernah berpikir bahwa Sony yang merupakan perusahaan penemu banyak barang-barang elektronik mulai dari transistor radio hingga alat perekam video, akan membuat sebuah *console* yang menjadi produk dengan tingkat penjualan yang tinggi hingga saat ini.

II.7 Klasifikasi *Game*

II.7.1 Berdasarkan Jenis "Platform" Atau Alat Yang Di Gunakan

Ada beberapa jenis platform di dunia *game* yang selalu dipilih oleh pengguna *game*s , yaitu :

1. *Arcade games,* yaitu yang sering disebut ding-dong di Indonesia, biasanya berada di daerah / tempat khusus dan memiliki box atau mesin yang memang khusus di design untuk jenis video *game*s tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa masuk dan menikmati , seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya tentunya).

2. *PC Games* , yaitu video *game* yang dimainkan menggunakan *Personal Computers.*
3. *Console games*, yaitu video *game*s yang dimainkan menggunakan console tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii.
4. *Handheld games*, yaitu yang dimainkan di console khusus video *game* yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP.
5. *Mobile games,* yaitu yang dapat dimainkan atau khusus untuk mobile phone atau PDA.

II.7.2 Berdasarkan "Genre" Permainanya

1. Aksi *Shooting*, (tembak-tembakan, atau hajar-hajaran bisa juga tusuktusukan, tergantung cerita dan tokoh di dalamnya), video *game* jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga timing, inti dari *game* jenis ini adalah tembak, tembak dan tembak. Termasuk didalam-nya :
    a. *First Person Shooting* (FPS) seperti Counter Strike dan Call Of Duty.
    b. *Drive n' shoot,* menggunakan unsur simulasi kendaraan tetapi tetap dengan tujuan utama menembak dan menghancurkan lawan, contoh : Spy Hunter, Rock and Roll Racing, Road Rash.
    c. *Shoot* em' up, seperti Raiden, 1942, dan Gradius.
    d. *Beat em up* (tonjok hajar) seperti Double Dragon dan Final Fight, lalu *hack and slash* (tusuk tebas) seperti Shinobi dan Legend of Kage.

e. *Light gun shooting*, yang menggunakan alat yang umumnya berbentuk seperti senjata, seperti Virtua Cop dan Time Crisis.

2. *Fighting* (pertarungan) Ada yang mengelompokan video *game* fighting dibagian Aksi, namun penulis berpendapat berbeda, jenis ini memangmemerlukan kecepatan refleks dan koordinasi mata-tangan, tetapi inti dari *game* ini adalah penguasaan jurus (hafal caranya dan lancar mengeksekusinya), pengenalan karakter dan timing sangatlah penting, *combo*-pun menjadi esensial untuk mengalahkan lawan secepat mungkin. Dan berbeda seperti *game*. Aksi pada umumnya yang umumnya hanya melawan Artificial Intellegence atau istilah umumnya melawankomputersaja, pemain jenis *fighting game*ini baru teruji kemampuan sesungguhnya dengan melawan pemain lainnya. Seri Street Fighter, Tekken, Mortal Kombat, Soul Calibur dan King of Fighter adalah contohnya.
3. Aksi Petualangan. Memasuki gua bawah tanah, melompati bebatuan di antara lahar, bergelayutan dari pohon satu ke pohon lain, bergulat dengan ular sambil mencari kunci untuk membuka pintu kuil legendaris, atau sekedar mencari telepon umum untuk mendapatkan misi berikutnya, itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam *video game* jenis ini. Menurut penulis, *game* jenis ini sudah berkembang jauh hingga menjadi genre campuran *action beat-em up* juga, dan sekarang, di tahun 2000 an, jenis ini cenderung untuk memiliki visual 3D dan sudut pandang orang ke-tiga. Tomb Rider, Grand Theft Auto dan Prince of Persia termasuk didalamnya.

4. Petualangan. Bedanya dengan jenis *video game* aksi-petualangan, refleks dan kelihaian pemain dalam bergerak, berlari, melompat hingga memecut atau menembak tidak diperlukan di sini. *Video Game* murni petualangan lebih menekankan pada jalan cerita dan kemampuan berpikir pemain dalam menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan percakapan karakter hingga penggunaan benda-benda tepat pada tempat yang tepat. Termasuk didalamnya:
    a. Petualangan dengan teks atau sistem tunjuk dan klik, contoh : Kings Quest, Space Quest, Heroes Quest, Monkey Island, Sam and Max.
    b. Novel atau *film* interaktif, seperti *game "dating"* yang banyak beredar di Jepang, Dragons Lair dan Night Trap.
5. Simulasi, Konstruksi dan manajemen. Video *Game* jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detil berbagai faktor. Dari mencari jodoh dan pekerjaan, membangun rumah, gedung hingga kota, mengatur pajak dan dana kota hingga keputusan memecat atau menambah karyawan. Dunia kehidupan rumah tangga sampai bisnis membangun konglomerasi, dari jualan limun pinggir jalan hingga membangun laboratorium cloning. *Video Game* jenis ini membuat pemain harus berpikir untuk mendirikan, membangun dan mengatasi masalah dengan menggunakan dana yang terbatas. Contoh: Sim City, The Sims, Tamagotchi.

6. *Role Playing. Video game* jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain (biasanya menjadi semakin hebat, semakin kuat, semakin berpengaruh, dan lain - lain) dalam berbagai parameter yang biasanya ditentukan dengan naiknya level, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun mahluk peliharaan.Secarakebudayaan, pengembang *game* Jepang biasanya membuat *Role Playing Game* (RPG) ke arah cerita linear yang diarahkan seolah karakter kita adalah tokoh dalam cerita itu, seperti Final Fantasy, Dragon Quest dan Xenogears. Sedangkan pengembang *game*RPG Eropa, cenderung membuat karakter kita bebas memilih jalan cerita sendiri secara non-linear, seperti Ultima, Never Winter Nights, Baldurs gate, ElderScroll, dan Fallout.
7. Strategi. Kebalikan dari video *game* jenis action yang berjalan cepat dan perlu refleks secepat kilat, video *game* jenis strategi, layaknya bermain catur, justru lebih memerlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. Video *game* strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai kendaraan, bahkan hingga pembangunan berbagai bangunan, pabrik dan pusat pelatihan tempur, tergantung dari tema ceritanya. Pemain *game* strategi melihat dari sudut pandang lebih meluas dan lebih kedepan dengan

waktu permainan yang biasanya lebih lama dan santai dibandingkan *game* action. Unsur-unsur permainannya biasanya berkisar sekitar, prioritas pembangunan, peletakan pasukan, mencari dan memanfaatkan sumberdaya (uang, besi, kayu,minyak,dan lain - lain), hingga ke pembelian dan peng-upgrade-an pasukan atau teknologi.

8. *Puzzle. Video game* jenis ini sesuai namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, sampai mendorong-dorong kota masuk ke tempat yang seharusnya, itu semua termasuk dalam jenis ini. Sering pula permainan jenis ini adalah juga unsur permainan dalam *video game* petualangan maupun *game* edukasi. Tetris, Minesweeper, Bejeweled, Sokoban dan Bomberman.
9. Simulasi kendaraan. *Video Game* jenis ini memberikan pengalaman atau interaktifitas sedekat mungkin dengan kendaraan yang aslinya, muskipun terkadang kendaraan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus pada detil dan pengalaman realistik menggunakan kendaraan tersebut.
10. *Olahraga.* Singkat padat jelas, bermain sport di PC atau konsol anda. Biasanya permainannya diusahakan serealistik mungkin walau kadang ada yang menambah unsur fiksi seperti NBA JAM. Contohnya pun jelas, Seri winning Eleven, seri NBA, seri FIFA, John Madden NFL, Lakers vs Celtics, Tony hawk pro skater, dan lain - lain.

II.8 Adobe Flash

Perangkat lunak *Adobe Flash* yang selanjutnya disebut *flash*, dulunya bernama “macromedia flash” merupakan software multimedia unggulan yang dulunya dikembangkan oleh macromedia, tetapi sekarang dikembangkan dan di distribusikan oleh *Adobe Sistem*. Sejak tahun 1996, *flash* menjadi populer untuk menambahkan animasi dan interaktif *website. Flash* biasanya digunakan untuk membuat animasi, hiburan dan berbagai komponen web, diintegrasikan dengan video dalam halaman web sehingga dapat menjadi aplikasi munlimedia yang kaya.

Flash tidak hanya digunakan untuk aplikasi web, tetapi dapat dikembangkan untuk membuat aplikasi desktop karena aplikasi *flash* selain dikompilasi menjadi format .swf, flash juga dapat dikompilasi menjadi format.exe.

Flash dapat untuk memanipulasi *vektor* dan citra rasler, dan dapat mendukung *bidirectional streaming video. Flash* juga berisi bahasa skrip yang diberi nama “*Actionscript*”. Beberapa produk *software, sistem* dan *device* dapat membuat dan menampilkan *flash. Flash* dijalankan dengan *adobe flash player* yang dapat ditanam pada *browser*, telepon seluler, *software* lain.

Format *file flash* adalah swf, biasanya disebut *ShockWafe Flashmovie* biasanya file berekstensi .swf dapat dijalankan melalui web,secara stand alone pada flash player atau dijalankan di windows secara langsung dengan membuatnya dalam format ekstensi.exe.

II.9 Action Script

Action scriptmerupakan suatu programming language (bahasa pemograman) yang dapat ditambahkan pada flash document (baik itu pada frame, movie clip, atau button) untuk dapat membuat suatu animasi yang lebih interaktif. Action script seperti halnya pada pemograman *C, C++* dan *Java* memiliki sifat yang sangat sensitive (*case-sensitive*) artinya penulisan huruf sangat berpengaruh [9].

Ada beberapa aturan yang harus di perhatikan ketika Anda hendak menggunakan action script:

1. **Penggunaan tanda titik koma**

   Di dalam *action script* penggunaan titik koma (;) di gunakan pada saat akhir syntac yang telah di buat. Penggunaan titik koma pada *action script* memiliki fungsi yang sama pada penggunaan tanda titik pada suatu kalimat. Berikut beberapa contoh : *Play*();, *Stop*();

2. **Penggunaan tanda kurung kurawal (*{}*)**

   Penggunaan tanda kurung kurawal pada *action script* sama seperti pada pemograman seperti C,C++ atau Java dalam membuat sebuah fungsi atau *procedure*.Tanda kurung kurawal buka (*{*) mengidentifikasikan blok awal dimana didalam blok itu kita dapat memasukan *action script* sesuai yang kita inginkan, sedangkan untuk mengakhiri blok tersebut diidentifikasikan dengan tanda kurung kurawal tutup(*}*) [9].

II.10 UML

Pemodelan (*modeling*) adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Model piranti lunak dapat dianalogikan seperti pembuatan blueprint pada pembangunan gedung. Membuat model dari sebuah

sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin komplek sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik.

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk *modeling* aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai *diagram* piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

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

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.1 Konsep dasar UML [11]

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

1. *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 cara menarik 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.

2. *Class diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Untuk Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja

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.

4. *Activity diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

5. *Sequence diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence *diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence *diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

- ✓ Masing-masing objek, termasuk aktor, memiliki lifeline vertikal.
- ✓ Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari *class*.
- ✓ *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.
- ✓ Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek *boundary, controller* dan *persistent entity*.

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.

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.

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 [11].