BAB II

LANDASAN TEORI

2.1 Edukasi

Edukasi, proses yang dilakukan oleh seseorang untuk menemukan jati dirinya, yang dilakukan dengan mengamati dan belajar yang kemudian melahirkan tindakan dan prilaku. Edukasi sebenarnya tidak jauh berbeda dari belajar yang dikembangkan oleh aliran *behaviorisme* dalam psikologi. Hanya istilah ini sering dimaknai dan diinterpretasikan berbeda dari *learning* yang bermakna belajar. Dan istilah ini seringkali digunakan dalam pendekatan pendidikan yang tentu maknanya lebih dari sekedar belajar.

Dalam teori belajar dan prilaku sebuah tindakan akan didapatkan dari sebuah aktifitas pengamatan yang dikondisikan dan direkayasa dengan tujuan hasil yang sudah ditentukan dan bisa diukur secara logis dan ilmiah. Dalam belajar ada hubungan erat antara penguat, baik yang positif atau negative yang terjadi setelah tindakan dilakukan, atau setelah menjadi sebuah prilaku kongkrit yang bisa dilihat dengan kasat mata. Stimulan yang terjadi sebelum prilaku muncul sangat mempengaruhi tujuan yang diharapakan. Ini artinya prilaku seseorang harus selalu bisa diukur dan selalu pula diberikan penguat dan stimulasi yang diciptakan sedemikian rupa sesuai dengan tujuan yang hendak dicapai.

Dengan demikian kita harus menciptakan dunia sendiri yang sesuai dengan kehendak kita dan mengkondisikannya sesuai dengan tujuan kita.

Sama halnya pula bahwa kehidupan ini tempat berexperimen dalam sebuah laboratorium dunia yang sangat komplek, betapa rumitnya bila itu dibayangkan. Itu semua ranah teoritis ilmiah yang tidak secara bisa diterapkan, dan memang tidak ada keharusan untuk menerapkan dalam kehidupan kita. Tetapi tidak ada yang memungkiri produk ilmiah bisa lebih dipertanggungjawabkan keefektifannya dari banyak sisi.

Edukasi dalam pendekatan pendidikan lebih kepada pemberian kondisi dan pengaruh kepada seseorang agar bisa menciptakan prilaku sepontan dalam jangka waktu yang lama sesuai dengan harapan diri sendiri dan masyarakat. Harapan itu diukur dengan kesepakatan-kesepakatan formal yang disebut hukum, dan kesepakatn-kesepakatan yang tidak formal tetapi menjadi kebiasaan yang dianggap baik berlaku di masyarakat yang disebut adat. inilah prilaku yang diharapkan dari edukasi yang sepadan dengan kata *learning* dalam pandangan psikologi *behaviorisme.*

2.2 Kurikulum Pendidikan

Kurikulum, seperangkat rencana dan pengaturan mengenai tujuan, isi dan bahan pelajaran serta cara yang digunakan sebagai pedoman penyelenggaraan kegiatan pembelajaran untuk mencapai tujuan pendidikan tertentu. Tujuan tertentu ini meliputi tujuan pendidikan nasional serta kesesuaian dengan kekhasan, kondisi dan potensi daerah, satuan pendidikan dan peserta didik. Oleh sebab itu kurikulum disusun oleh satuan pendidikan

untuk memungkinkan penyesuaian program pendidikan dengan kebutuhan dan potensi yang ada di daerah.

Pendidikan bertujuan mencetak anak didik yang beriman. Menurut Tatang (2012:61) wujud dari tujuan itu adalah akhlak anak didik yang mengacu pada kurikulum yang diterapkan dalam pendidikan yang dilaksanakan di berbagai lembaga, baik lembaga formal maupun informal. Tujuan itu sasaran yang hendak dicapai sekaligus pedoman yang memberi arah aktivitas yang dilakukan.

Pengembangan Kurikulum Tingkat Satuan Pendidikan (KTSP) yang beragam mengacu pada standar nasional pendidikan untuk menjamin pencapaian tujuan pendidikan nasional. Standar nasional pendidikan terdiri atas standar isi, proses, kompetensi lulusan, tenaga kependidikan, sarana dan prasarana, pengelolaan, pembiayaan dan penilaian pendidikan. Dua dari kedelapan standar nasional pendidikan tersebut, yaitu Standar Isi (SI) dan Standar Kompetensi Lulusan (SKL) merupakan acuan utama bagi satuan pendidikan dalam mengembangkan kurikulum.

Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 (UU 20/2003) tentang Sistem Pendidikan Nasional dan Peraturan Pemerintah Republik Indonesia Nomor 19 tahun 2005 (PP 19/2005) tentang Standar Nasional Pendidikan mengamanatkan kurikulum pada KTSP jenjang pendidikan dasar dan menengah disusun oleh satuan pendidikan dengan mengacu kepada SI dan SKL serta berpedoman pada panduan yang disusun oleh Badan Standar Nasional Pendidikan (BSNP).

Selain dari itu, penyusunan KTSP juga harus mengikuti ketentuan lain yang menyangkut kurikulum dalam UU 20/2003 dan PP 19/2005. Panduan yang disusun BSNP terdiri atas dua bagian. Pertama, Panduan Umum yang memuat ketentuan umum pengembangan kurikulum yang dapat diterapkan pada satuan pendidikan dengan mengacu pada Standar Kompetensi dan Kompetensi Dasar yang terdapat dalam SI dan SKL. Termasuk dalam ketentuan umum penjabaran amanat dalam UU 20/2003 dan ketentuan PP 19/2005 serta prinsip dan langkah yang harus diacu dalam pengembangan KTSP.

Kedua, model KTSP sebagai salah satu contoh hasil akhir pengembangan KTSP dengan mengacu pada SI dan SKL dengan berpedoman pada Panduan Umum yang dikembangkan BSNP. Sebagai model KTSP, tentu tidak dapat mengakomodasi kebutuhan seluruh daerah di wilayah Negara Kesatuan Republik Indonesia (NKRI) dan hendaknya digunakan sebagai referensi.

2.3 Biologi

Biologi berasal dari bahasa Inggris: *biology* dan gabungan kata dari bahasa Yunani yaitu *βίος, bios*, yang berarti "kehidupan" atau "peradaban"; λόγος, dan logos, yang berarti "kata" atau "ilmu" atau "alasan"). Biologi merupakan studi ilmiah tentang kehidupan makhluk hidup. Biologi mempelajari struktur, fungsi, pertumbuhan, asal, evolusi, distribusi benda hidup, morfogenesis, reproduksi, patogenesis, dan sebagainya.

Biologi berkaitan erat dengan deskripsi karakteristik dan perilaku organisme individu dan spesies secara keseluruhan, dan reproduksi organisme hidup dan interaksi antara mereka dan lingkungan. Bidang ini menjelaskan bagaimana organisme, fungsi mereka, cara dan sebab spesies, serta interaksi sesama mereka dan alam sekeliling.

Biologi banyak melakukan pengamatan benda hidup maupun benda mati mulai dari struktur biologi molekul organel, sel, agregat sel, jaringan dan organ pada organisme yang kompleks. Dalam konteks yang lebih luas, perilaku organisme individu dan interaksi mereka dengan orang lain dan lingkungan juga diteliti dengan menggunakan dan menghasilkan metode, teori dan model.

Kebanyakan *sains* biologi merupakan disiplin khusus. Botani merupakan penelitian tumbuh-tumbuhan, zoologi merupakan penelitian hewan, dan mikrobiologi merupakan penelitian mikroorganisme. Bidang lain dalam biologi berdasarkan skala organisme dikaji dan metode mempelajari mereka. Biokimia mempelajari kimia dasar hidup, biologi molekul mengkaji interaksi sistem molekul biologis yang kompleks, biologi sel mempelajari benda binaan kehidupan dasar, yaitu sel, sedangkan fisiologi merupakan kajian fungsi jaringan dan organ organisme fisik dan kimia, dan ekologi merupakan studi hubungan sesama berbagai organsima dan alam sekeliling mereka.

Peredaran darah pada manusia tersusun tiga komponen utama, yakni jantung, pembuluh darah, dan darah. Menurut Oman (2007:125)

mendefinisikan sebagai berikut Jantung manusia dan Mamalia lainnya memiliki empat ruangan, yakni serambi kiri (*atrium sinister*), serambi kanan (*atrium dekster*), bilik kiri (*ventrikel sinister*), dan bilik kanan (*ventrikel dekster*).

Dalam keadaaan normal darah ada di dalam pembuluh darah, ujung arteri bersambung dengan kapiler darah dan kapiler darah bertemu dengan vena terkecil (*venula*) sehingga darah tetap mengalir dalam pembuluh darah walaupun terjadi pertukaran zat, hal ini disebut sistem peredaran darah tetutup.

Darah merupakan cairan yang terdapat pada semua makhlik hidup (kecuali tumbuhan) tingkat tinggi yang berfungsi mengirimkan zat-zat dan oksigen yang dibutuhkan oleh jaringan tubuh, mengangkut bahan-bahan kimia hasil metabolisme, dan juga sebagai pertahanan tubuh terhadap virus atau bakteri. Menurut Istamar (2003:138) alat peredaran darah terdiri dari jantung dan pembuluh-pembuluh darah, yakni arteri, vena, dan kapiler.

Menurut Neil dan Jane (2008:63) kapiler adalah pembuluh darah terkecil, yang memiliki diameter hanya sedikit lebih besar dari sel darah merah. Kapiler juga memiliki dinding yang sangat tipis, terdiri dari endotilium saja dan lamina basalnya. Dinding–dinding arteri dan vena memiliki organisasi yang lebih kompleks daripada dinding-dinding kapiler.

Pencernaan pada manusia, proses pencernaan merupakan suatu proses yang melibatkan organ-organ pencernaan dan kelenjar-kelenjar pencernaan. Antara proses dan organ-organ serta kelenjarnya merupakan kesatuan sistem

pencernaan. Sistem pencernaan berfungsi memecah bahan-bahan makanan menjadi sari-sari makanan yang siap diserap dalam tubuh.

Sistem pencernaaan terdiri atas organ-organ yang membentuk saluran pencernaan. Sistem ini berfungsi memecah zat-zat kompleks menjadi zat-zat yang sederhana sehingga menjadi mudah diserap oleh tubuh. Karbohidrat, protein, dan lemak harus melalui proses pencernaan terlebih dahulu sebelum diserap oleh tubuh. Sebaliknya, vitamin, mineral, dan air dapat langsung diserap oleh tubuh. Saluran pencernaan pada manusia adalah mulut – kerongkongan - lambung – usus halus – usus besar – rektum – anus. Selain melibatkan berbagai organ, proses pencernaan melibatkan kelenjar – kelenjar pencernaan.

2.4 Peranan Multimedia Dalam Pendidikan

Kelebihan Multimedia dalam dunia Pendidikan :

1. Sistem pembelajaran lebih inovatif dan interaktif. Mampu mengabungkan antara teks, gambar, audio ,musik, animasi gambar atau video dalam satu kesatuan yang saling mendukung guna tercapainya tujuan pembelajaran.
2. Mampu menimbulkan rasa senang selama Proses Belajar Mengajar (PBM) berlangsung. Hal ini akan menambah motivasi siswa selama PBM hingga didapatkan tujuan pembelajaran yang maksimal.

3. Mampu memvisualisasikan materi yang selama ini sulit untuk diterangkan hanya sekedar dengan penjelasan atau alat peraga yang konvensional.
4. Media penyimpanan yang relatif gampang dan fleksibel.

2.5 *Education games*

Education games (permainan edukatif) yaitu suatu kegiatan yang sangat menyenangkan dan dapat merupakan cara atau alat pendidikan yang bersifat mendidik. *Education games* (permainan edukatif) sebuah permainan yang digunakan dalam proses pembelajaran dan dalam permainan tersebut mengandung unsur mendidik atau nilai-nilai pendidikan. Selain itu, untuk pemilihan permainan, diusahakan agar seluruh aspek yang dimiliki anak dapat berkembang dengan baik, baik dari segi kognitif, afektif dan juga psikomotorik.

Oleh karena itu perlu ditunjang alat bantu yang tepat saat bermain. Adapun kriteria-kriteria pemilihan alat bantu tersebut agar permainan dapat membantu belajar secara optimal dan tidak terjadi kekeliruan dalam menyelesaikan dan menentukan alat dan bahan yang diperlukan secara tepat guna.

2.6 *Neatbeans*

Netbeans merupakan sebuah aplikasi *Integrated Development Environment (IDE)* yang berbasiskan Java dari *Sun Microsystems* yang

berjalan di atas *swing. Swing* merupakan sebuah teknologi Java untuk pengembangan aplikasi desktop yang dapat berjalan pada berbagai macam platform seperti windows, linux, Mac OS X dan Solaris. Sebuah *IDE* merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan *Graphic User Interface (GUI),* suatu kode editor atau text, suatu *compiler* dan suatu *debugger.*

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah). *Sun Microsystems* mendirikan proyek kode terbuka *NetBeans* pada bulan Juni 2000 dan terus menjadi sponsor utama. Dan saat ini pun *netbeans* memiliki 2 produk yaitu *Platform Netbeans* dan *Netbeans IDE*.

Platform Netbeans merupakan *framework* yang dapat digunakan kembali (*reusable*) untuk menyederhanakan pengembangan aplikasi desktop dan *Platform NetBeans* juga menawarkan layanan-layanan yang umum bagi aplikasi desktop, mengijinkan pengembang untuk fokus ke logika yang spesifik terhadap aplikasi. Fitur-fitur dari Platform *Netbeans* antara lain:

1. Manajemen antarmuka (misal: menu & toolbar)
2. Manajemen pengaturan pengguna
3. Manajemen penyimpanan (menyimpan dan membuka berbagai macam data)
4. Manajemen jendela
5. *Wizard framework* (mendukung dialog langkah demi langkah)

Netbeans IDE merupakan sebuah *IDE open source* yang ditulis sepenuhnya dengan bahasa pemrograman java menggunakan platform netbeans. *NetBeans IDE* mendukung pengembangan semua tipe aplikasi Java *(J2SE, web,* dan aplikasi *mobile)*. Fitur lainnya adalah sistem proyek berbasis *Ant*, kontrol versi, dan *refactoring.*

2.7 Java 2 Micro Edition (*J2ME*)

2.7.1 Pengertian *J2ME*

Java 2 Micro Edition (*J2ME*) lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. *J2ME* membawa Java ke dunia informasi, komunikasi dan perangkat komputasi selain perangkat komputer desktop, yang biasanya lebih kecil. *J2ME* mempunya beberapa *library* khusus yang tidak dimiliki *J2SE*.

Teknologi *J2ME* juga memiliki beberapa keterbatasan, terutama jika diaplikasikan pada ponsel. *J2ME* sangat tergantung pada perangkat *(device)* yang digunakan, bisa dari segi merk ponsel, maupun kemampuan ponsel dan dukungannya terhadap teknologi *J2ME*. Misalnya, jika sebuah ponsel tidak memiliki kamera, maka jelas *J2ME* pada ponsel tersebut tidak dapat mengakses kamera. Keterbatasan lainnya pada ukuran aplikasi, karena memori pada ponsel sangat terbatas. Sebagian besar ponsel tidak mengijinkan aplikasi *J2ME* menulis pada file karena alasan keamanan.

2.7.2 Konfigurasi *J2ME*

Konfigurasi hanyalah mengatur hal–hal yang berkaitan dengan "kesamaan", bukan mengatur hal–hal yang "membedakan", sehingga konfigurasi memastikan portabilitas antar *device.* Perkembangan konfigurasi ditentukan oleh *Java Community Process (JCP),* inilah badan non-profit yang berkutat dengan perkembangan teknologi java. Saat ini telah didefinisikan 2 konfigurasi yaitu *connected device configuration (CDC)* dan *connected limited device configuration (CLDC).*

2.7.2.1 *Connected Device Configuration (CDC)*

Connected Device Configuration (CDC) bertujuan untuk mangaplikasikan kemampuan-kemampuan utama dari tiap-tiap jenis device, dimana target CDC tersebut adalah memiliki minimal memori 2MB termasuk RAM dan ROM. CDC menspesifikasikan penggunaan Java 2 Platform Virtual Machine secara penuh dan disebut juga Compact Virtual Machine (CVM).

2.7.2.2 *Connected Limited Device Configuration (CLDC)*

Tujuan dari *Connected Limited Device Configuration (CLDC)* adalah untuk menggambarkan suatu platform Java standar untuk *device* dengan spesifikasi memori 160KB. Karena variasi sistem perangkat lunak yang luas pada berbagai personal device,

CLDC membuat asumsi minimum tentang lingkungan dimana *CLDC* berada.

2.7.3 *Profile J2ME*

Lapisan kedua adalah lapisan *profile.* Terdiri dari sekumpulan kecil *Application Programming Interfaces (API). Profile* merupakan kebalikan dari konfigurasi, yaitu mengatur hal–hal yang spesifik untuk sebuah *device.* Kembali pada analogi mobil, sebuah mobil BMW tentu memiliki ciri spesifik yang tidak dimiliki oleh mobil Mercedes. Berikut ini adalah profil *J2ME* yang tersedia saat ini yaitu :

1. *Foundation Profile*

   Foundation Profile digunakan dengan konfigurasi *CDC* dan merupakan inti untuk hampir semua *profile* yang menggunakan konfigurasi *CDC* karena *Foundation Profile* mengandung *class–class* dasar Java.

2. *Game Profile*

   Game Profile digunakan dengan konfigurasi *CDC* dan mengandung *class–class* penting untuk pengembangan aplikasi *game* untuk semua *small computing device* yang menggunakan konfigurasi *CDC.*

3. *Mobile Information Device Profile (MIDP)*

   Mobile Information Device Profile (MIDP) digunakan dengan konfigurasi *CLDC* dan mengandung *class–class* yang

menyediakan tempat penyimpanan lokal, *user interface* dan kapabilitas jaringan untuk aplikasi yang berjalan pada *mobile computing device* seperti *Palm OS device. MIDP* digunakan dengan *wireless* aplikasi Java.

4. *Personal Digital Assistants (PDA) Profile*

   PDA Profile digunakan dengan konfigurasi *CLDC* dan mengandung *class–class* yang menggunakan sumber daya *sophisticated* yang terdapat pada *PDA.* Fitur ini mengandung *display* yang lebih baik dan memori yang lebih besar dibandingkan sumber daya yang sama yang terdapat pada *MIDP mobile device.*

5. *Personal Profile*

   Personal Profile digunakan dengan konfigurasi *CDC* dan *foundation profile* dan mengandung *class–class* untuk implementasi *user interface* yang kompleks. *Foundatio profile* menyediakan *class–class* dasar dan *personal profile* menyediakan *class–class* untuk implementasi *user interface* yang teruji.

6. *RMI Profile*

   RMI profile digunakan dengan konfigurasi *CDC dan foundation profile* untuk menyediakan *class remote method invocation* untuk *class–class* dasar yang dimiliki *foundation profile*

2.8 *Game*

2.8.1. Pengertian *Game*

Game sebuah hiburan berbentuk multimedia yang di buat semenarik mungkin agar pemain bisa mendapatkan sesuatu sehingga adanya kepuasaan batin. Membuat *game* membutuhkan keahlian dan ketekunan, bahkan juga seni. Seringkali, dibutuhkan berbulan-bulan untuk menyelesaikan satu *game* yang siap dimainkan oleh pengguna.

Menurut Erdhi dan Ridwan (2012:2) jika lingkungan sosial mereka adalah sesama pengguna *gadget*, berbagi informasi *game* baru merupakan bagian yang tak terpisahkan pada saat bertemu. Sekali ada yang tertarik, maka potensi sejumlah penguna juga segera muncul di dalam kelompoknya. Dalam waktu singkat, *game* menjadi populer dan diunduh oleh kerabat-kerabatnya.

Game edukasi dibuat dengan tujuan spesifik sebagai alat pendidikan, untuk belajar mengenal warna, mengenal huruf dan angka, matematika, sampai belajar bahasa asing. Desainer yang membuat *game* harus memperhitungkan berbagai hal agar *game* benar-benar dapat mendidik, menambah pengetahuan dan meningkatkan keterampilan yang memainkannya.

2.8.2 Sejarah Perkembangan *Game*

Manusia telah mengenal dan memainkan *game* sejak zaman dahulu. Di Sahara ditemukan sebuah papan permainan terbuat dari batu

yang berusia ± 5000 tahun. Permainan *Go*, yang populer di negara–negara oriental, telah ada sejak 2000 B.C.. Bahkan permainan mirip Backgamon (Tabula, Nard) dicatat pada *script* romawi kuno. Memasuki zaman modern, jenis permainan (*game*) pun memasuki era baru. Permainan–permainan seperti *Pong*, *Zork*, dan lain–lain muncul di komputer dan digemari oleh orang banyak.

Game Arcade, yang dipelopori hits seperti Pac-man, berhasil menjaring jutaan pemain. Tidak mau kalah dengan *Arcade*, *Console systems* seperti *Magnavox Odyssey*, *Atari* 2006, *Mattel Intellvision* dan *Calleco Vision* menciptakan *game* yang bisa dimainkan di rumah. Seiring dengan itu, pada tahun 1985 *Nintendo Entertainment System (NES)* menghebohkan orang–orang dengan grafik dan *gameworld* yang luar biasa lewat *Super Mario Brothers*.

Pada perkembangannya, *computer gaming* menjadi semakin diterima dan populer melalui beberapa bestseller seperti *: Doom, Quake* dan *Tomb Rider*. *Clearly ultra realistic 3D* merupakan kunci suksesnya game tersebut. Pada akhir 90-an dan awal 2000-an, dengan berkembangnya game seperti : *Ultima Online*, *Ragnarok Online* dan *Everquest* menyadarkan kita bahwa *game* tidak hanya menarik dengan grafik yang bagus, namun juga dengan kehadiran pemain lain *(multiplayer).*

Sementara *multiplayer game* terus berkembang, muncul paradigma baru dalam dunia informasi. Dengan banyaknya penggunan *handphone, game* turut berkembang dan mendukung pesatnya pemasaran *micro device* tersebut. Walaupun grafik yang dapat didukung tidak sebanding dengan *console, handphone* menawarkan kepraktisan yang tidak bisa diberikan oleh *console.*

2.8.3 *Game Mobile*

Pengembangan global dari *game mobile* dipelopori oleh *game i-Mode. I-Mode* merupakan layanan Internet *wireless* yang sangat popular di Jepang juga di negara lain. *Do Communications Over the Mobile Network* (*DoCoMo*) adalah yang pertama kali memperkenalkan Java dan *game* pada *handphone* (*Anonymous A*). Jepang merupakan negara pertama yang memperkenalkan warna layar dan grafik 3-Dimensi (3D) untuk *hanphone*, yang mana sangat penting dalam perluasan penjualan *game mobile* diluar *game–game mobile* yang sangat sederhana.

Game mobile adalah suatu *game* komputer yang dimainkan di *handphone, smartphone, PDA,* atau *handheld. Game mobile* bahkan dapat dimainkan dengan mengunakan teknologi komunikasi yang ada pada alatnya sendiri, contohnya pesan teks (*SMS*), pesan multimedia (*MMS*) atau pengidentifikasian *GPRS* (*Anonymous B*).

2.9 Grafik dan *Sprite*

Salah satu elemen grafik yang penting dalam *game* yaitu *sprite*. *Sprite* merupakan kumpulan *image* yang mewakili berbagai *states* dalam kesatuan *graphikal*. Pengertian secara teratur beberapa *sprite* akan menghasilkan suatu *animasi*. Objek bergerak yang sering kita lihat dalam *game,* misalnya pemain bola pada *game Winning Eleven*, atau karakter *game* lainnya merupakan contoh dari *sprite*.

2.10 *Ponsel*

Telepon selular (*ponsel*) atau telepon genggam (telgam) atau *handphone* (HP) atau disebut pula adalah perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon konvensional saluran tetap, namun dapat dibawa ke mana-mana (portabel, *mobile*) dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel (nirkabel; *wireless)*.

Saat ini Indonesia mempunyai dua jaringan telepon nirkabel yaitu sistem *GSM (Global System for Mobile Telecommunications*) dan *sistem CDMA (Code Division Multiple Access).* Badan yang mengatur telekomunikasi seluler Indonesia adalah Asosiasi Telekomunikasi Seluler Indonesia (ATSI).

2.11 *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. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. *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-*extends use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2.11.1 Komponen Pembentuk *Use-case* Diagram

1. *Actor*

Pada dasarnya *actor* bukanlah bagian dari *use-case* diagram, namun untuk dapat terciptanya suatu *use-case diagram* diperlukan beberapa *actor*. *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem.

Actor hanya berinteraksi dengan *use-case*, tetapi tidak memiliki kontrol atas *use-case*. *Actor* digambarkan dengan *stick man*. *Actor* dapat digambarkan secara secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*

Ada beberapa kemungkinan yang menyebabkan *actor* tersebut terkait dengan sistem antara lain :

a. Yang berkepentingan terhadap sistem dimana adanya arus informasi, baik yang diterimanya maupun yang dia inputkan ke sistem
b. Orang ataupun pihak yang akan mengelola sistem tersebut
c. *External resource* yang digunakan oleh sistem

d. Sistem lain yang berinteraksi dengan sistem yang akan dibuat

2. *Use-case*

Use-case adalah gambaran fungsionalitas dari suatu sistem, pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. *Use-case diagram* adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use-case* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian.

Cara menentukan *Use-case* dalam suatu sistem :

a. Pola perilaku perangkat lunak aplikasi
b. Gambaran tugas dari sebuah *actor*
c. Sistem atau "benda" yang memberikan sesuatu yang bernilai kepada *actor*
d. Apa yang dikerjakan oleh suatu perangkat lunak.

Tipe relasi atau *stereotype* yang mungkin terjadi pada *use-case diagram* :

a. <<*include*>>, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use-case* adalah bagian dari *use-case* lainnya.

b. <<*extends*>>, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan *alarm*.
c. <<*communicates*>>, mungkin ditambahkan untuk asosiasi yang menunjukkan asosiasinya adalah *communicates association* . Ini merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara *actor* dan *use-case*.

2.11.2 *Activity* Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang 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. Aktivitas digambarkan dengan menggunakan segiempat dengan sudut membulat. *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 merupakan *state* diagram khusus, dimana 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. 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.

Diagram aktivitas ini adalah tipe khusus dari digram *state* yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu *system.* Diagram ini terutama penting dalam pemodelan fungsi-fungsi. Notasi yang digunakan dalam *activity* diagram adalah sebagai berikut:

1. *Activity*

   Notasi yang menggambarkan pelaksanaan dari beberapa proses dalam aliran pekerjaan.

2. *Transition*

   Notasi yang digunakan untuk memperlihatkan jalan aliran *control* dari *activity* ke *activity.*

3. *Decision*

   Notasi yang menandakan kontrol cabang aliran berdasarkan *decision point*.

4. *Synchronization bars*

   Aliran kerja notasi ini menandakan bahwa beberapa aktivitas dapat diselesaikan secara bersamaan (pararel).

2.11.3 *Sequence* Diagram

Sequence diagram suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

Pada diagram ini, dimensi *vertikal* merepresentasikan waktu. Bagian paling atas dari diagram menjadi titik awal dan waktu berjalan ke bawah sampai dengan bagian dasar dari diagram. Garis *Vertikal*, disebut *lifeline*, dilekatkan pada setiap obyek atau aktor. Kemudian, *lifeline* tersebut digambarkan menjadi kotak ketika obyek melakukan suatu operasi, kotak tersebut disebut *activation box.*

Obyek dikatakan mempunyai *live activation* pada saat tersebut. Pesan yang dipertukarkan antar obyek digambarkan sebagai sebuah

anak panah antara *activation box* pengirim dan penerima. Kemudian diatasnya diberikan label pesan. Tujuan penggunaan *sequence* diagram

a. Mengkomunikasikan *requirement* kepada tim teknis karena diagram ini dapat lebih mudah untuk dielaborasi menjadi model *design.*
b. Merupakan diagram yang paling cocok untuk mengembangkan model deskripsi *use-case* menjadi spesifikasi *design*

2.11.4 *Class Diagram*

Class merupakan 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

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

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package* . Kita juga dapat membuat diagram yang terdiri atas *package*.